

Borland In-Depth Series\Borland 大系

JBuilder X

高级技术手册

Borland 中国公司 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

《JBuilder X 高级技术手册》是由 Borland 中国公司的技术团队精心编撰的一本 JBuilder X 的深入应用书籍，它涵盖了最完整与实用的 JBuilder X 开发技术。通过本书详实的内容，读者将学会如何使用 JBuilder 开发高效率的 Java 应用程序。本书内容包括：建立与编辑 XML 文件，利用 BorlandXML 与 Castor 进行 XML 数据绑定，JBuilder X 与 Struts 集成应用以及 JBuilder X 的团队开发功能。对于广大 JBuilder 开发人员来说，本书是一本必备的实务应用书籍。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目（CIP）数据

责任编辑：周 筠 陈元玉

责任校对：张兴田

印 刷：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

经 销：各地新华书店

开 本：787×980 1/16 印张：23.75 字数：560 千字

印 次：2004 年 6 月第 1 次印刷

印 数：6 000 册 定价：39.00 元（含光盘 1 张）

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。质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

序

Borland JBuilder 是全球第一的跨平台 Java 开发环境，可建构符合业界标准的 Java 应用系统，开发 EJB、Web、XML 以及数据库等各类应用程序。经由 JBuilder 双向的、可视化的设计工具，您可以快速开发各种 J2EE 应用程序，并将其部署到多种应用程序服务器上，包括 BEA WebLogic、IBM WebSphere Sun One、Oracle9i Application Server 以及集成于 JBuilder 的 Borland Enterprise Server。

长久以来，Borland 中国公司为持续推广 Java 应用开发技术不遗余力。《JBuilder X 高级技术手册》是由 Borland 中国公司的技术团队精心编撰的一本 JBuilder X 的深入应用书籍，它涵盖了最完整与实用的 JBuilder X 开发技术。通过本书详实的内容，读者将学会如何使用 JBuilder 开发高效率的 Java 应用程序。本书内容包括：建立与编辑 XML 文件，利用 BorlandXML 与 Castor 进行 XML 数据绑定，JBuilder X 与 Struts 集成应用以及 JBuilder X 的团队开发功能。对于广大 JBuilder 开发人员来说，本书是一本必备的实务应用书籍。

JBuilder X 高级技术手册

目 录

第 1 章 建立与检核 XML 文件	1
1.1 建立 XML 文件	2
1.2 使用 JBuilder 编辑器来撰写 XML 文件	5
1.3 检核 XML 文件	8
1.4 实践练习：建立与检核 XML 文件	11
1.5 结束语	18
第 2 章 查看与转换 XML 文件	19
2.1 查看 XML 文件	20
2.2 利用“样式表”转换 XML 文件	22
2.3 实践练习：查看与转换 XML 文件	25
2.4 结束语	32
第 3 章 XML 数据绑定	33
3.1 何谓 XML 数据绑定	34
3.2 JBuilder X 提供的 XML 数据绑定技术	34
3.3 实践练习：JBuilder X 的 XML 数据绑定技术	39
3.4 结束语	59
第 4 章 JBuilder X 的 XML 数据库组件	61
4.1 JBuilder X 的 XML 模型组件	62
4.2 如何利用 XML 模型组件进行数据转换	66
4.3 实践练习一：使用 XML 模型组件	70

4.4 JBuilder X 的 XML 模板组件	89
4.5 实践练习二：使用 XML 模板组件.....	95
4.6 结束语	103
第 5 章 JBuilder X 与 Struts 开发框架.....	105
5.1 Struts 简介	106
5.2 JBuilder X 对 Struts 提供哪些支持.....	106
5.3 利用 JBuilder X 建立 Struts 应用程序.....	122
5.4 结束语	123
第 6 章 使用 Struts Config Editor	125
6.1 Struts Config Editor 的组成.....	126
6.2 Data Sources 设定页	127
6.3 Form Beans 设定页.....	130
6.4 Global Exceptions 设定页.....	134
6.5 Global Forwards 设定页	137
6.6 Action Mappings 设定页	140
6.7 Controller 设定页.....	145
6.8 Message Resource 设定页	148
6.9 Plug Ins 设定页	151
6.10 结束语	154
第 7 章 使用 Struts 可视化设计工具	155
7.1 Action Designer.....	156
7.2 Form Bean Designer.....	161
7.3 实践练习：开发 Struts 应用程序	164
7.4 结束语	178
第 8 章 JBuilder X 内建的版本控管机制.....	179
8.1 比对两份不同的程序代码	180
8.2 在本地端进行项目版本控管	183

8.3 针对单一文件进行版本控管	185
8.4 实践练习: JBuilder X 的文件版本控管	190
8.5 结束语	211
第 9 章 CVS 版本控管技术	213
9.1 使用 CVS 前的环境设定	214
9.2 将 JBuilder 项目上传至 CVS 储存库	215
9.3 从 CVS 储存库取出 JBuilder 项目	219
9.4 更新 CVS 储存库的项目 (或文件) 状态	223
9.5 结束语	233
第 10 章 JBuilder X 与 CVS 实例探讨	235
10.1 实践练习 10-1: 将 JBuilder 项目上传至 CVS 储存库	236
10.2 实践练习 10-2: 从 CVS 储存库下载 JBuilder 项目	245
10.3 实践练习 10-3: 查看项目开发进度	251
10.4 实践练习 10-4: 排除“合并冲突”问题	254
10.5 结束语	268
第 11 章 Borland StarTeam 概览	269
11.1 Borland StarTeam 概览	270
11.2 安装 StarTeam Server	272
11.3 重要的 StarTeam Server 管理技巧	279
11.4 实践练习: StarTeam Server 基本管理技巧	293
11.5 结束语	300
第 12 章 利用 StarTeam 进行团队开发	301
12.1 利用 StarTeam 管理 JBuilder 项目	302
12.2 利用 StarTeam 管理特定文件	313
12.3 管理 StarTeam 的“处理项目” (process items)	323
12.4 结束语	327

第 13 章 JBuilder X 与 StarTeam 实例探讨	329
13.1 实践练习 13-1: 指定 JBuilder 项目的版本控管系统	330
13.2 实践练习 13-2: 将 JBuilder 项目上传至 StarTeam Server	331
13.3 实践练习 13-3: 从 StarTeam Server 下载 TeamDev 项目	337
13.4 实践练习 13-4: 加入“变更请求”	350
13.5 实践练习 13-5: 依照“变更请求”修改 TeamDev 项目	358
13.6 结束语	365

第 1 章

建立与检核 XML 文件

近年来网际网络蓬勃发展，电子商务日渐盛行，许多企业对于应用程序网络化的需求大增。然而，如果要在不同系统之间互相传递数据，必须有一套共通的数据描述标准才行。

XML (eXtensible Markup Language —— 可扩展标记语言) 自从 1998 年被 W3C 正式采用后，就快速融入各种应用软件与程序。现在，XML 已经是电脑世界里信息交换的标准语言，其应用范围相当广泛。在本章中，将介绍如何利用 JBuilder X 建立与检核 XML 文件。

本章内容着重在 JBuilder X 的 XML 集成功能，假设您对 XML 语言已有一定程度的了解。如果您想进一步了解 XML 技术，请自行参阅 XML 相关书籍。

1.1 建立 XML 文件

针对 XML 文件的撰写，JBuilder X 提供了许多实用的功能，可以帮助您建立、检核、编辑与查看 XML 文件内容。本节先介绍两项建立 XML 文件的利器：

- n DTD to XML Wizard。
- n XML to DTD Wizard。

1.1.1 DTD to XML Wizard 的用途

JBuilder X 的 **DTD to XML Wizard** 可以通过某个 DTD (Document Type Definition) 文件产生一份 XML 文件。使用这个向导之前，请注意一点：DTD to XML Wizard 只能产生 XML 文件的“范本” (Template)。

为什么只能产生 XML 范本呢？由于 DTD 规定了 XML 文件应使用的元素与属性，所以 JBuilder 只能根据 DTD 产生 XML 文件的基本结构，无法决定每个元素的真实内容。下列范例即为 DTD to XML Wizard 所产生的 XML 文件内容：

```
1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE Orders SYSTEM "Orders.dtd">
3. <Orders>
4.     <Order>
5.         <OrderID>pcdata</OrderID>
6.         <ProductID>pcdata</ProductID>
7.         <ProductType>pcdata</ProductType>
8.         <Price>pcdata</Price>
9.         <Quantity>pcdata</Quantity>
10.    </Order>
11.</Orders>
```

您是否注意到，第 5 ~ 9 行的 XML 元素值均为 **pcdata** 字符串。您应该在事后将 **pcdata** 置换为真实数据。

1.1.2 使用 DTD to XML Wizard 建立 XML 文件范本

在 AppBrowser 内启动 DTD to XML Wizard 的方式有两种：

- n 在“项目窗格”内选取欲产生 XML 文件的 DTD 文件，按下鼠标右键，选取 (Generate

XML)。

- n 打开 (Object Gallery) 对话框。在 XML 窗格内，以鼠标左键双击 (DTD to XML) 图标。

如图 1-1 所示：DTD to XML Wizard 只有一个设置窗口，包含以下设置：

- n Input DTD file: 指定 DTD 源文件的名称与路径。如果您在“项目窗格”内事先选取某个 DTD，再执行 (Generate XML) 指令启动 DTD to XML Wizard，那么 JBuilder 会自动将 DTD 文件名称填入该字段。
- n Root element: 指定 XML 文件的“根元素” (root element)。
- n Output XML file: DTD to XML Wizard 所产生的 XML 文件名称与存放路径。

下方的 Identifiers 区域与 DOCTYPE 的定义有关：

- n Public: 指定 public DTD 的位置与名称。
- n System: 指定 local DTD 的位置与名称。

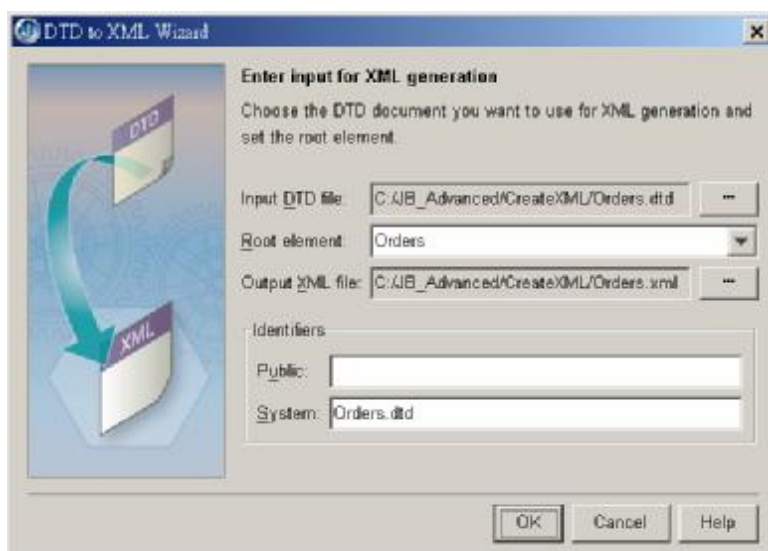


图 1-1 DTD to XML Wizard

注意：

如果 DTD 内包含 ATTLIST 定义，那么 DTD to XML Wizard 会自动在 XML 内产生相对应的属性 (attributes)。

1.1.3 使用 XML to DTD Wizard 产生 DTD 文件

JBuilder X 的 **XML to DTD Wizard** 可针对某份 XML 文件产生相对应的 DTD 文件。XML 解析器 (parser) 将可以利用此 DTD 来检核 XML 文件内容。

启动 XML to DTD Wizard 的方式有两种：

- n 在“项目窗格”中选取欲产生 DTD 文件的 XML 文件，按下鼠标右键，选取 (Generate DTD)。
- n 打开 (Object Gallery) 对话框。在 XML 窗格内，以鼠标左键双击 (XML to DTD) 图标。

XML to DTD Wizard 只有一个设置窗口 (见图 1-2)，包含下列两个设置：

- n **Input XML file** 字段：指定 XML 源文件之名称与路径。如果您在“项目窗格”内事先选取某个 XML，再执行 (Generate DTD) 指令打开 XML to DTD Wizard，此时 JBuilder 会自动将 XML 文件名称填入该字段。
- n **Output DTD file** 字段：XML to DTD Wizard 产生的 DTD 文件名称与存放路径。

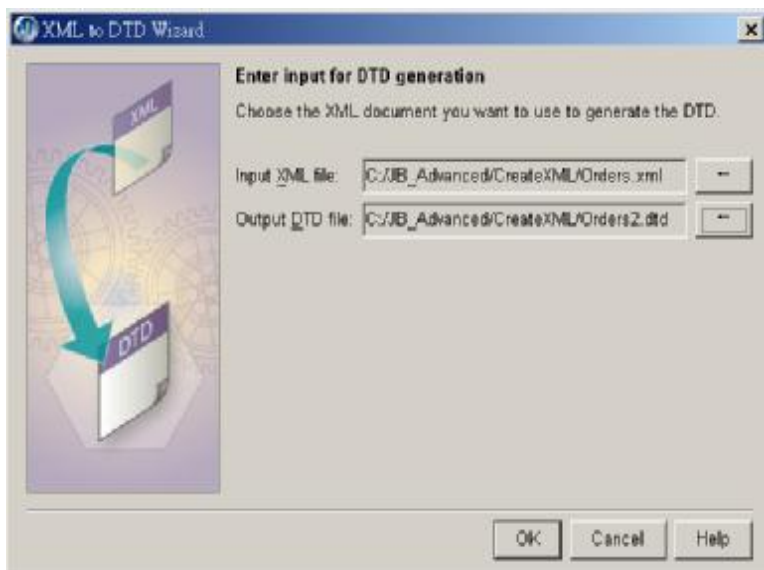


图 1-2 XML to DTD Wizard 设置窗口

1.1.4 手动建立 XML 文件

前面两节介绍过如何以 JBuilder 自动产生 XML 或 DTD 文件,但是前提是您必须拥有一份 XML 或 DTD 文件。如果您希望在目前项目内建立一份 XML 文件(或是其他与 XML 有关的文件,例如 XSL、DTD、XSD),可采用下列方式:

1. 选取 File | New File, 打开〔Create New File〕对话框。
2. 在 Name 字段输入文件名称。
3. 在 Type 下拉式菜单内选取 **xml** (或其他文件类型)。
4. 在 Directory 字段内指定新文件的储存路径。
5. 如果勾选 (Add saved file to project) 选项, 新文件建立之后将会自动加入您的项目。

回到 AppBrowser, 您将可以在 JBuilder 编辑器内撰写 XML 文件内容。

1.2 使用 JBuilder 编辑器来撰写 XML 文件

对于 XML 文件的撰写, JBuilder X 也提供相当完整的支持。在 JBuilder 编辑器内撰写 .xml、.xsl、.xsd 等 XML 文件时, “语法标示 (Syntax Highlighting)” 与 XML TagInsight 这两项实用功能可提高 XML 文件的撰写效率。

1.2.1 XML 文件之“语法标示”功能

JBuilder X 的“语法标示 (Syntax Highlighting)”功能会自动以不同颜色来标注不同元素。举例来说, XML 元素名称默认是以蓝色显示, 属性名称则是以红色显示。当然了, 您也可以针对“语法标示”功能进行自定义, 方式如下:

1. 选取 Tools | Preferences, 打开〔Preferences〕对话框。
2. 切换至 Editor | Color 窗格 (如图 1-3 所示), 然后选取 “HTML/XML” 页签。
3. 在 “Screen element” 区域内选取欲更改的项目, 然后设置前景颜色 (位于 Foreground 调色盘) 与背景颜色 (位于 Background 调色盘)。

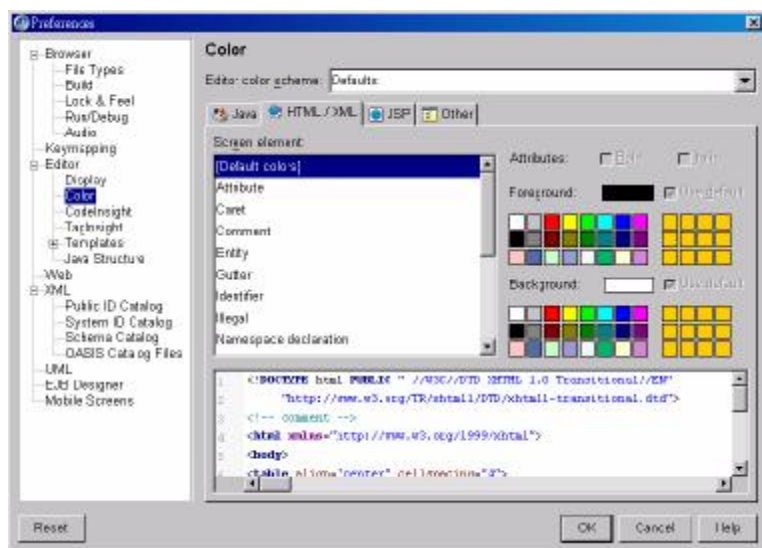


图 1-3 (Preferences) 对话框的 Editor | Color 窗格

1.2.2 XML TagInsight

在编辑 XML 文件内容时，**XML TagInsight** 功能可以加入 XML 元素名称、属性名称或是属性值。对于撰写 XML 文件而言，这是相当实用的技巧。

使用 XML TagInsight 功能之前，请注意：您的 XML 文件必须先定义 DTD 或是 XML schema 所在位置，例如：

```
<!DOCTYPE Orders SYSTEM "Orders.dtd">
```

否则 XML TagInsight 功能将无法使用！

在图 1-4 中，在 `<Order>` 元素下方键入 “`<`” 符号，XML TagInsight 功能会自动显示 (Order Tag Subelements) 弹出式菜单。

您或许已注意到，图 1-4 的 (Order Tag Subelements) 弹出式菜单内只有 OrderID 可以选取，其余元素却被加上“删除线”？别误会，这也是 XML TagInsight 的一项贴心设计。根据 *Orders.dtd* 的定义，`<Order>` 元素的第一个子元素为 `<OrderID>`，因此 XML TagInsight 功能希望您先选取 `<OrderID>` 元素，以符合 DTD 的定义。当您在 (Order Tag Subelements) 菜单内选取 OrderID 之后，按下【Enter】键，JBuilder 将自动补上 `</OrderID>` 结束标签。

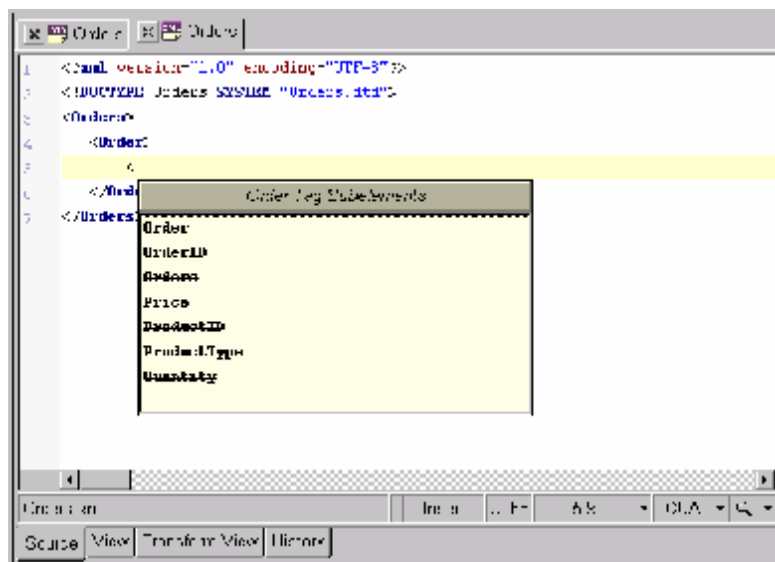


图 1-4 XML TagInsight 功能

如图 1-5 所示，当在 `<OrderID>` 元素下方键入 “`<`” 符号时，XML TagInsight 菜单内只有 `ProductID` 未被加上“删除线”。这意味着 `ProductID` 为 `<Order>` 的第二个子元素。

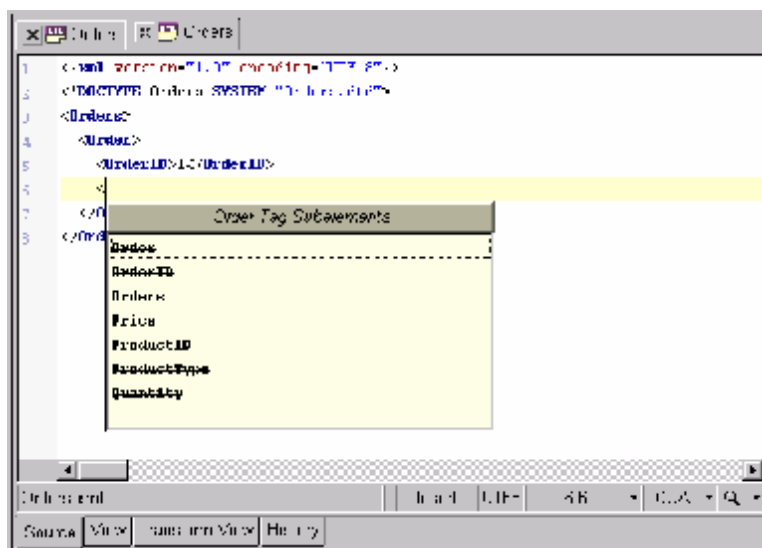


图 1-5 XML TagInsight 功能

注意：

在 XML TagInsight 菜单中，被加上“删除线”的元素仍然可以加入 XML 文件。

在 (Preferences) 对话框的 Editor | Color 窗格中提供 XML TagInsight 的高级选项 (见图 1-6), 这里将它们整理于表 1-1 中。

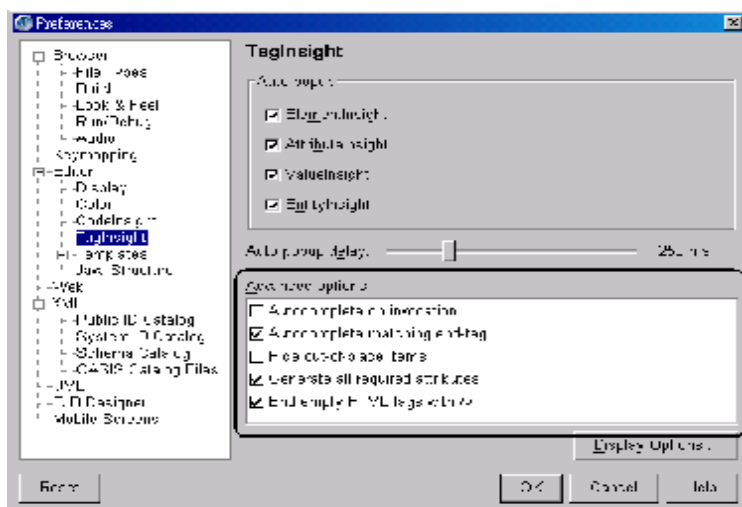


图 1-6 XML TagInsight 功能的高级选项

表 1-1 XML TagInsight 的高级选项

XML TagInsight 高级选项	说 明
Autocomplete on invocation	调用 XML TagInsight 功能时, 如果符合的子元素只有一个。JBuilder 会自动加入该元素, 此时并不会显示 XML TagInsight 菜单
Autocomplete matching end-tag	在 XML TagInsight 菜单内选取某个 XML 元素名称后按下【Enter】键, JBuilder 会自动补上该元素的结束标签
Hide out-of-place items	隐藏不适当的元素名称
Generate all required attributes	加入新元素时, JBuilder 会自动加入所有必要的属性
End empty HTML tags with />	如果 HTML 标签并未包含任何内容, 将以 “/>” 作为结束标签

1.3 检核 XML 文件

在 XML 文件中, 检核内容正确性的方式有两种: “格式正确”(well-formed)与“文法正确”(grammatical validity)。一份“格式正确”的 XML 文件必须遵循 XML 文件结构及语法的规范, 例如: 每份 XML 文件都必须有一个根元素, 所有 XML 元素都位于根元素之下。

注意，“格式正确”的 XML 文件并不代表一定是“文法正确”。一份“文法正确”的 XML 必须符合下列两点：

- n 必须是“格式正确”的 XML。
- n 必须遵循 DTD 或 XML Schema (XSD) 文件所设置的规则。

DTD 文件除了定义 XML 文件结构，也定义 XML 元素的类型，以及各元素的属性。换言之，如果 XML 文件无法通过 DTD 的检核，或是 DTD 文件不存在，就不能算是“文法正确”的 XML。

1.3.1 JBuilder 编辑器采用的 XML 解析方式

JBuilder X 是采用 Xerces parser 来检核与解析 XML 文件内容的。只要您在 XML 文件内声明 DTD 或 XSD 的所在位置或 URL，JBuilder 就会自动进行程序解析。

针对 XML 的检核，请注意一点：如果您以 URL 来声明 DTD 或 XSD 所在位置，JBuilder 必须从网络上寻找到正确的 DTD 位置，才能进行解析与检核。这种方式将影响 XML 文件的编辑效率。为了避免这种情况，建议您先下载 DTD 或 XSD，然后在本地端进行验证。举例来说，如果您欲使用的 DTD 所在位置为 <http://www.abc.com/DTD/Orders.dtd>，原来在 DOCTYPE 的声明为：

```
<!DOCTYPE Orders SYSTEM "http://www.abc.com/DTD/Orders.dtd">
```

此时您可以先下载这份 DTD，保存为 `D:\OrderEntryApps\DTD\Orders.dtd`，然后将 DOCTYPE 声明更改成：

```
<!DOCTYPE Orders SYSTEM "D:\OrderEntryApps\DTD\Orders.dtd">
```

1.3.2 检核 XML 文件是否为“格式正确”(Well-formed)

如果 XML 文件并非“格式正确”，“结构窗格”的 Errors 文件夹将显示错误信息。如果 XML 文件不符合“文法正确”的要求，错误信息将会显示于“消息窗格”。在图 1-7 中，笔者刻意将 `<Order>` 元素的结束标签 (`</Order>`) 移除，Errors 文件夹会立刻显示下列错误信息：

```
The end-tag for element type "Order" must end with a '>' delimiter.
```

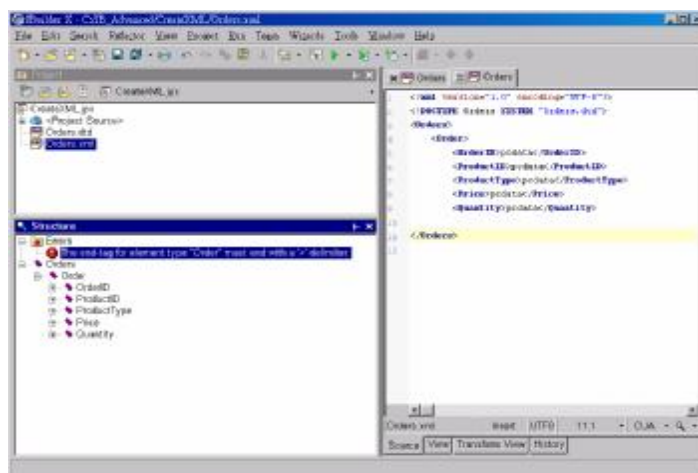


图 1-7 格式不正确的 XML 错误信息

1.3.3 检核 XML 文件是否为“文法正确”

如果要检核 XML 文件是否为“文法正确”，请依照下列步骤进行：

1. 先在 XML 内声明 DTD 或 XSD。
2. 在 XML 文件上点击鼠标右键，执行 (Validate) 指令，JBuilder 将立刻执行检核程序。

在图 1-8 中，笔者刻意将 DOCTYPE 声明移除，执行 Validate 指令后，“消息窗格”会显示错误信息，代表 JBuilder 无法针对这份 XML 文件进行检核。

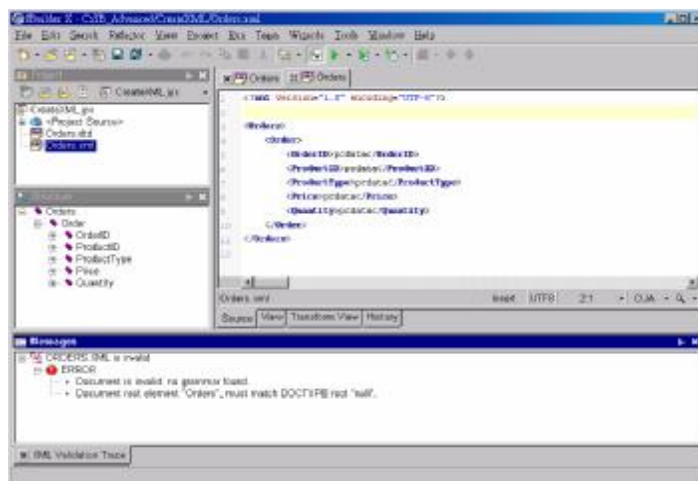


图 1-8 缺少 DOCTYPE 声明，无法进行检核程序