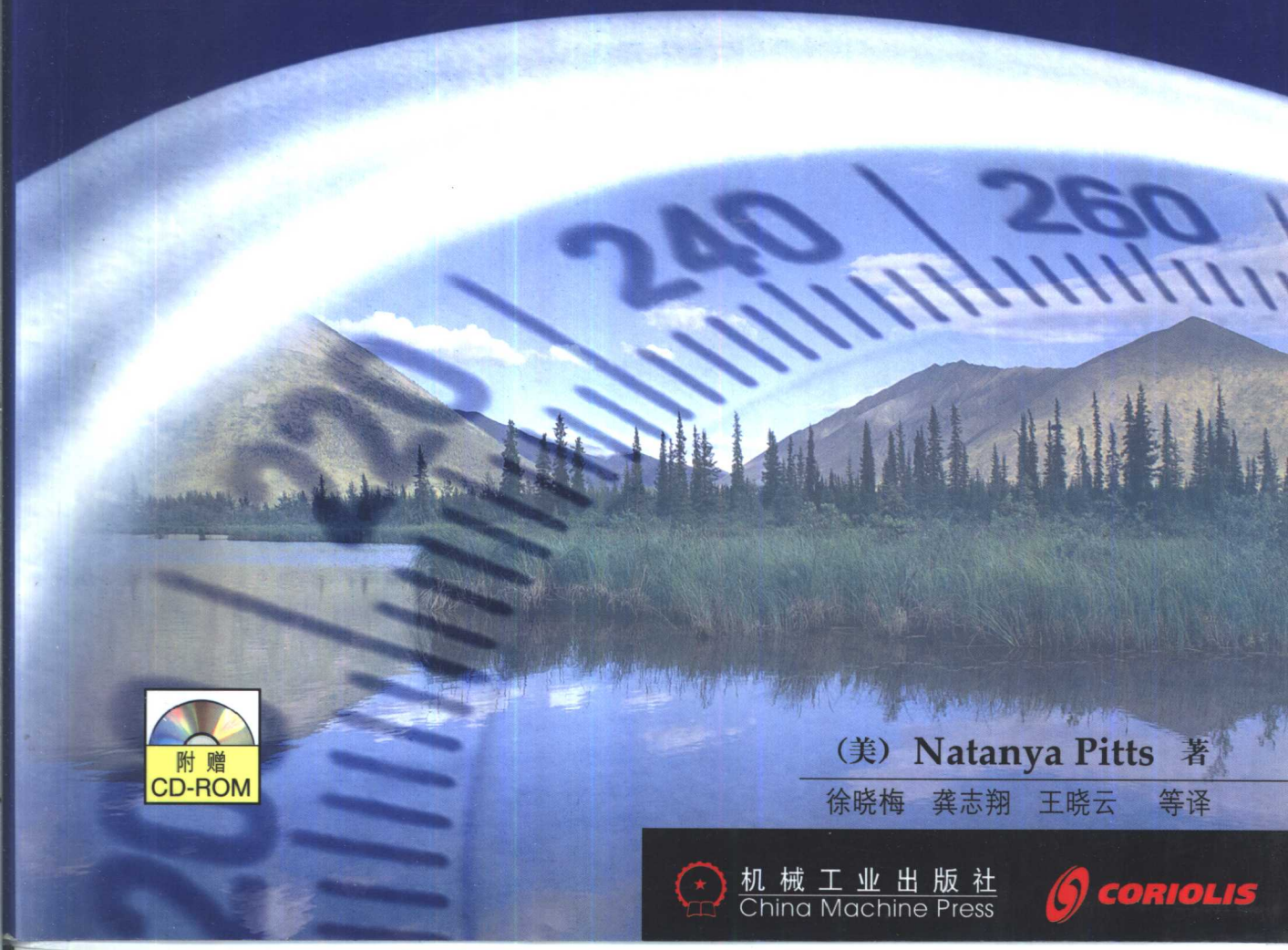
HZ BOOKS

XML Black Book
(2nd Edition)

软件开发技术丛书

XML 技术内幕



附赠
CD-ROM

(美) Natanya Pitts 著

徐晓梅 龚志翔 王晓云 等译

机械工业出版社
China Machine Press

CORIOLIS

软件开发技术丛书

XML 技术内幕

（美） Natanya Pitts 著

徐晓梅 龚志翔 王晓云 等译

机械工业出版社
China Machine Press

本书介绍了如何使用 XML 来解决现实生活中的信息传递问题。主要内容包括:标记语言介绍,XML 与 HTML 比较,XML 设计和解决方案实例等等。本书每章都分为深入讲解和快速解决方案两部分,使本书既具有一定的理论深度,又具有极大的实用价值。

本书附赠光盘包括:多个最流行的 XML 解析器、浏览器和开发系统,相关信息以及书中所有的代码文件。

Natanya Pitts:XML Black Book,2nd Edition.

Original English language edition published by The Coriolis Group LLC, 14455N. Hayden Drive, Suite 220, Scottsdale, Arizona 85260 USA, telephone(602)483-0192, fax(602)483-0193.

本书版权登记号:图字:01-2001-1110

图书在版编目(CIP)数据

XML 技术内幕/(美)皮特斯(Pitts, N.)著;徐晓梅等译 .-2 版 .-北京:机械工业出版社,2002.1

(软件开发技术丛书)

书名原文:XML Black Book,2nd Edition

ISBN 7-111-09311-9

M535/02

Ⅰ.X... Ⅱ.①皮 ...②徐 ... Ⅲ. 可扩充语言,XML-程序设计 Ⅳ.TP312

中国版本图书馆 CIP 数据核字(2001)第 067260 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:张鸿斌 徐江红

北京昌平奔腾印刷厂印刷 ·新华书店北京发行所发行

2002 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·33 印张

印数:0 001-4000 册

定价:65.00 元(附光盘)

译 者 序

XML 和 HTML 都是 SGML 的子集，SGML 是 60 年代出现的标记语言，最初起源于 IBM 所制定的通用标记语言(GML)。SGML 提供了创建、展示文档，以及与其他用户交换文档的一种标准方式，而无须关心所使用的系统或平台。然而对于许多用户来说，该语言太高级，也太复杂，实现起来比较麻烦。当 20 世纪 90 年代早期 Web 成为全球焦点时，欧洲粒子物理研究所设计了 HTML，它是简化的 SGML 应用。HTML 很快风靡全球，原因是它摒弃了 SGML 的许多复杂性，但仍提供大部分 SGML 功能。然而 HTML 缺乏扩展性，不能满足多媒体、数据库访问等新兴需求，而借助第三方工具或者插件技术来实现又往往带来兼容性的问题。因而，人们迫切希望能够得到兼具 HTML 和 SGML 优点的标记语言。这样 XML 就诞生了。

XML 是可扩展(或可扩充)的，并且能利用结构化标记语言的所有高级功能，同时又摒弃了 SGML 的复杂性。最关键的是 XML 可以描述数据，而 HTML 不具备这种能力。XML 集中处理怎样形成将被展示的数据，而至于数据展示则由显示设备负责(借助于样式表)。开发者可以根据特定的行业需求，使用 XML 创建自定义的数据结构。这些数据结构和数据库可以在许多设备上查看，而不需要使用自定义接口在不同显示设备上查看相同的数据。借助于 XML 设计的应用，你在浏览站点时将很少见到“最好使用 Netscape 浏览器”，或者“本网站需要使用 Internet Explorer”之类的告诫。

本书不但具有一定的理论深度，而且还非常实用。通过对本书的学习，读者不但能够掌握标记语言(尤其是 XML)的基本理论知识，而且能够立即使用 XML 进行自己的设计和开发。本书每章都分为两部分：“深入讲解”和“快速解决方案”。其中，“深入讲解”主要在理论上介绍基本术语、方法和过程；“快速解决方案”部分则就某一实例，详细分析实现和操作步骤，让你完全理解理论部分所讲解的理论知识，最终实现理论和实践的紧密结合。

现在，我们很荣幸能够承担本书的翻译工作。在翻译过程中，我们经常为一句话、一个术语进行反复的讨论，到处查找资料，力图使本书的翻译能正确、体贴地反映原文的意思，同时注意使句子、段落符合中国人的语言习惯。我们真挚地希望广大读者能够从本书中有所收获，这是作者的初衷，也是我们良好的愿望！

本书由徐晓梅、龚志翔、王晓云组织翻译，万方工作室的全体同仁都参加了本书的翻译、校对和输入等工作。具体参加本书翻译、录排、校对工作的其他人员有：韩存兵、陶华敏、强秀丽、任宇飞、尹建军、刘今朝、李红玲、白红利、田敏、龚露娜、马军、马丽、田军、田野、田蕴哲、金荣学、薛彪、叶哲、邓海燕、邢倩、王育红、李军、刘彬、钱斌、赵锁、姜南、李智、田韫、李林、张巧莉、陈曙晖、邓波、邓涛、李卓林、聂宛析、王小将、李素丽、天海鹏等。本书的出版是集体劳动的结晶，在此特别感谢万方工作室的全体工作人员。

由于时间仓促，且译者经验和水平有限，译文难免有不妥之处，恳请读者批评指正！

万方工作室

2001 年 5 月

前　言

本书专门介绍目前非常流行的元语言，即可扩展标记语言，简称为 XML(Extensible Markup Language)。现在，网站管理员、Web 内容开发者、数据库专家，以及所有 IT 专业人士都对 XML 趋之若鹜，希望 XML 能帮助他们更有效地表示结构化数据，尤其当他们必须通过 Web 方式把数据移交给顾客时更是如此。

为什么 XML 具有如此大的魅力，以至于那么多人都相信 XML 能有助于完成各种信息管理任务，不仅有利于传递 Web 内容，而且有助于组织其他方面的内容和代码呢？原因主要有两个：

其一，XML 中的 X 代表可扩展性(Extensible)，这意味着你可以使用 XML 定义自己特定的标记，以便更方便地获取、组织和表示数据。在本书中，你将学习如何定义各种自定义 XML 标记，以及如何使用这些特定的标记来表达数据。

其二，现在已经出现 100 多种定义好的 XML 标记语言——也称做 XML 应用(applications)。你可以使用这些现成的工具在许多领域(如电子商务事务、化学或者数学公式、系统图等等)来做许多事情，帮助自己获取或者表达信息；你也可以随时学习和使用许多 XML 应用来完成自己的任务。在本书中，你会学习如何选定和使用许多有趣的 XML 应用，以及学习如何确定和分析其他 XML 应用，以满足自己的数据处理需求。

在本书中，我们首先介绍一些关于 XML 起源和发展的历史背景知识。但主要介绍如何使用 XML 来解决现实生活中的信息传递问题。具体内容包括标记语言的概括性介绍、XML 与 HTML 的比较，以及如何创建、验证和转换 XML 标记以满足工作需求。通过本书的学习，你还能够了解如何使用多种样式表来处理 XML 文档的表示问题，如何管理链接和引用，以及如何使用 XML 名字空间。在本书中，我们将讨论如下 XML 应用：信道定义格式(CDF，Channel Definition Format)、可扩展超文本标记语言(XHTML，Extensible Hypertext Markup language)、XML 路径语言(XPath，XML Path Language)、XML 指针语言(XPointer，XML Pointer Language)，以及 XML 链接语言(XLink，XML Linking Language)。除此之外，你还将学习如何操纵数据库，如何用 XML 编程等等。

本书适合你吗？

本书的读者对象是 XML 中级或者高级用户。在学习本书所讨论的主题时，你会发现作者首先从理论上进行讲解，然后用大量的实例帮助说明。下面这些主题就是如此：

- 诸如 XML、XHTML 和 HTML 等标记语言的使用原理和方法，以及如何设计自定义 XML 标记，以满足特定的数据处理和数据表示需求。除此之外，你还要学习 XML 标记语言的所有基本组成部分(如元素、属性、实体等等)。同时，我们还会向你解释如何使用自定义标记来创建正式的文档类型定义(Document Type Definition，DTD)，以用于有效性验证和错误检测。

• 当通过 Web 进行访问时 XML 文档的处理方式，包括 XML 文档怎样转换成常见的 HTML 格式，以便于尽可能为 Web 用户所接受。

• 对于使用级联样式表语言(Cascading Style Sheet，CSS)和可扩展样式表语言(Extensible StyleSheet Language，XSL)的 XML 文档，怎样管理文档的表示方式？以及在使用用于链接、路径和指针的 XML 应用的 XML 文档(外部文档或者外部引用)中，怎样创建和扩展超链接？

• 如何操作已有的大量 XML 应用，包括 CDF 和 XHTML？如何创建和处理自己的基于 XML 的应用，以使自己的结构化数据方便顾客、用户和合伙人的处理和使用。本书还讨论如何链接保存于数据库管理系统(DBMS)中的数据和 XML 应用。

• 本书向你推荐一个丰富的 XML 软件工具集，其中包括多种编辑器、支持 XML 的 Web 浏览器、分析器和处理器、转换工具、数据库、内容管理和发布系统，以及可以实现把 XML 转化成 HTML 以便于 Web 发布的 XSL 转换(XSL Transformation，XSLT)工具。

总之，通过本书的学习，你能掌握 XML 概念、工具和相关技术。我们同时相信：你的 IT 生活中使用这种功能强大的数据处理工具时，尤其当使用 Web 来表达自己的想法时，你会深深地认识到本书的重要性。

如何使用本书？

尽管熟悉 XML 的读者可以选择本书中自己需要的章节和主题进行学习，但我建议那些尚未使用过 XML 的读者最好依次阅读本书内容。这样的话，你可以对 XML 术语、主题和工具有一个全面的认识，有助于你理解 XML 是什么，以及怎样使用 XML 满足自己的数据处理需求。

本书分成 23 章，书后还附带有一组附录。

首先，我们主要介绍 XML 的基本概念，以及 XML 所能够提供的功能。第 1 章从总体上介绍标记语言，给出 XML 的基本原理，并讨论了 XML 历史、XML 产生动机、规范说明，以及典型应用。在第 2 章中，我们分析了 XML 文档的结构，以及可以执行的各种功能。在第 3 章中，我们对 XML 和 HTML 进行了比较，并提供一组准则，以帮助你决定何时使用标记语言，以及使用标记语言的目的是什么。在第 4 章中，我们再一次综述前 3 章所讨论的主题，并解释怎样使用它们来理解和解决现实工作中遇到的问题。

从第 5 章开始讨论描述 XML 标记的 DTD，到定义和创建 XML 文档以获取特定内容的各种活动。在第 6 章中，我们讨论了 XML 标记元素，这些元素是 XML 文档的组成模块，并可以定义 XML 文档的结构。第 7 章讲述了如何使用 XML 来创建内容。第 8 章讨论了 XML 属性，以及怎样使用属性来帮助调节和控制 XML 文档内容。第 9 章讲解了 XML 实体，XML 实体可以定义标准的特征集和文本，也可以用于获取和简化常见的文档元素和内容。

接下来主要讨论 XML 扩展、样式表和链接机制，链接机制可用于管理 XML 文档被浏览时的外观，以及 XML 文档相互之间怎样进行交互。在第 10 章中，我们主要讲解 CSS，CSS 是一种外观管理标记语言，在 HTML 和 XML 中都可以使用。第 11 章介绍 XSL，XSL 是一种特定于 XML 的样式语言，尤其需要注意的是，XSL 具有转换能力(XSLT)，可以把用 XML 标记表示的内容转换成 HTML 页面。第 12 章到第 14 章主要讨论 XML 的各种链接处理、路径处理和指针处理应用，这样可以为 XML 文档提供功能更强的超链接机制。最后，在第 15 章中，

我们介绍和解释了 XML 名字空间。借助于名字空间,你可以在文档中集成预定义的标准 XML 标记,而无须集成那些控制文档的 DTD。

然后主要讨论两个重要的 XML 应用,现在许多站点都想利用这两个应用来开发。首先,在第 16 章中,我们介绍了定义和使用 XML 应用的一般机制。然后,在第 17 章中,我们讨论了 XML 应用 CDF。借助于 CDF,Web 站点可以向那些预定用户发送新闻、文档、代码或者其他信息的实时数据流。第 18 章主要讨论基于 XML 的 HTML 实现,称做 XHTML。该章解释了为什么 XHTML 比 HTML 更优越(从最简单的角度考虑,XHTML 遵循与控制 XML 相同的严格的、正式的规则),以及如何实现 HTML 到 XHTML 的转换。

最后介绍 XML 的工作机理,以及怎样才能充分利用 XML 提供的功能。所以从这个角度来说,可以把本部分称做"XML 用户指南"。在第 19 章中,讲解了当支持 XML 的浏览器、编辑器或者分析器读取标记时,怎样分析和处理 XML 文档。在第 20 章中,我们解释了完善的 XML 数据移交——不管通过 Web,还是通过其他途径进行移交——解决方案的组成元素。在第 21 章中,讨论了怎样实现 XML 与数据库管理系统的结合,以使 XML 文档可以访问或者更新数据库内容,提供更强的应用能力。最后,在第 22 章中,给出一个理想的 XML 软件工具箱,其中包括各种编辑器、浏览器、分析器和其他 XML 相关的软件,这些工具基本上可以满足你的开发和使用需求。当然,你也可以根据本身特定的应用需求,添加自己选定的工具。

本书后面还包括几个附录,附录 A 是 XML 资源一览表,内容详尽,而且非常有用。借助于该附录中的资源,你能够及时了解 XML 最新的发展动态,最新的规范和应用,以及搜索出最好的支持 XML 的工具和技术。附录 B 是 XML 规范说明。通过附录 B,你能够进一步了解 XML 的内部工作机理。术语表给出许多属于 XML,或者与 XML 相关的技术术语的详尽解释和说明;借助于术语表,你能够对关键术语有更深刻和正确的理解。

在阅读完本书前 3 部分之后(或者你已经具有必要的背景知识),你就有能力选择自己最感兴趣的章节直接进行学习。再说一遍,本书的内容是前后贯通,浑然一体的,每个章节的内容都是基于前面的章节内容的。因此,如果你对 XML 并不是非常熟悉的话,我建议你能够按照顺序阅读本书所有章节,这样有利于阅读和掌握本书的内容,同时有利于建立自己的知识框架。

我们相信,在阅读或者浏览本书提供的材料之后,你肯定会有所收获。本书是解决问题的参考书,能够在极大程度上提高你对数据和文档的管理和组织能力。在深刻理解所学内容之后,你应该能够在本书的基础上学习更高深的主题、标记语言和工具,并且以多种方式变通应用。

我建议你首先阅读本书,对 XML 术语、概念、工具和技术有一个较深刻的认识。然后,尽量找机会使用本书提供的技术来解决疑难问题,或者提供信息移交解决方案。只有在边学边练的基础上,你才能精益求精,获取对 XML 更深入的了解。

本书由经验丰富的专业人士编写,相信会帮助你更优秀地完成自己的任务。本书非常有特色,每个专题都分成两部分:第一部分是纯粹的技术讲解,第二部分是实践性很强的解决方案,这种组织结构有助于你更快更好地掌握高深的技术细节,同时使你能够应用自己的知识来解决问题,提高实践动手能力,从而进一步提高自己对技术的理解,最终成为一位高手。通过

把高深的主题组织成易于操作和学习的结构，以及丰富的图表和代码，你能够更快地认识到自己的缺陷，并有的放矢地进行学习。

当你学习本书时，请注意：本书中的所有已经编号的代码范例都有电子版本，并都放在本书所附带的光盘中。当想重复书中实例时，你不必重新输入代码，省却许多无谓的工作。同时，光盘还包含本书第 23 章所提到的某些工具，你可以随时使用，只不过有的工具只是评估版（开发者的声明条款中有版权和使用说明）。借助于这些工具，你可以对本书所讨论的内容进行进一步的研究，也可以随意地创建、调试、验证、测试和配置自己的 XML 标记。

总之，要花很多时间研究和考虑 XML 到底能为你做什么工作。如果你对 XML 的熟悉程度与我们一样（我们制作考试和课程的 XML 标记，以便于 Web 移交；我们能够获取关于书籍、课程和工具的信息；能够把 XML 标记转化为 HTML，以便于在我们的 Web 站点上发布），你会发觉 XML 的用处多少是与你在 XML 上花费的时间和精力成正比的，而 XML 这个灵活的功能强大的技术本身的局限性则没有那么重要了。充分开发和利用该技术吧！

我殷切地期望听到你对本书的反馈意见，不管批评、赞扬或者其他意见，我都会非常诚挚耐心地听取。你可以给 Coriolis Group 发邮件，地址为 ctp @coriolis.com，或者直接给我发邮件，natanya@io.com 。在 www. lanw. com /books /errata 站点上有本书的勘误表、更新情况，以及其他信息。

目　录

第1章 标记语言

你可能已经开始阅读本书，并希望能够了解可扩展标记语言是什么，以及如何实现它。当然，在读完本书之后，你不仅会得到上述问题的答案，而且会获得更多的知识。但是我们在开始讲解这些问题之前，你有必要首先了解一点标记语言的原理，以及 XML、标准通用标记语言(SGML, Standard Generalized Markup Language)和超文本标记语言(HTML, Hypertext Markup Language)之间的区别。对标记语言的深刻理解是学习如何开发 XML 文档的先决条件。如果没有这些知识，你将不能把 XML 概念和实践应用到自己的文档开发中去。

在本章中，我们会解释标记语言是什么，标记语言的各种组件，以及当前的标记语言提供哪些功能。除此之外，我们还会简略地介绍 XML 的发展历史，以及 XML 存在的原因。

1.1 样式和标记基础

阅读本书时，你会发现本书在设计和布局方面具有高度的一致性。章节标题和文本使用一致的字体显示。除此之外，段落标题的高度也很固定，工具条和提示很容易辨别。为获得一致性的外观，以及为了加速出版进度，我们在本书排版时对本书中不同信息创建了不同的样式——或者说规则。设计者根据不同的信息特点创建样式，设置样式属性，给格式化文本分配特定的名称，这样只需要简单地对选定的信息块实施特定的样式，就可以快速地实现信息呈现的一致性。

下面我们以样式 Code(Code 样式是应用于本书中代码行的样式)为例加以说明。设计者设计该样式的目的是想保证本书中所有 XML 文档代码在字体和字号等方面的一致性。为使本书中用户输入的代码与用户阅读的文本能够很好地区分，设计者使用如下形式来定义代码的外观：程序代码采用一种类似于计算机屏幕所显示的字体。但是设计者并没有局限于此。他们通读全书，根据不同文本在文档中的作用，将不同的文本进行分类(段落、标题，等等)，分别为这些文本定义不同的样式。结果，本书中具体某段文本的显示形式由样式表决定，并且完全依赖于本段文本在文档中所扮演的角色。

在角色被定义，以及特定样式的详细说明被指定之后，每种样式的电子版本就创建起来。对于本书而言，样式是在 Microsoft Word(一种广泛应用于出版行业的应用程序)中创建的。借助于内置于字处理程序的二进制说明，编辑可以指定特定的格式说明，并可以给特定格式命名。具有特定自定义名称的格式化规则集合称做样式表(style sheet)。当作者需要某种样式时，他可以直接从样式列表中选择，然后应用到所选定的文本。不管文本是谁写的，不管内容如何，具有相同样式的文本在外观上都是一致的，这样全书在总体上看起来也比较统一。样式伴随着文档，这样设计者、编辑和作者在文本样式上就达到统一的认识。

当以电子文档形式交换本书的章节时，Microsoft Word 会读取样式列表，然后基于所应用的样式显示相应的文本。作者既看不到把简单文本转换为粗体、24 点、Times New Roman 文

本的特殊指令，也无须自己输入任何特定的命令以启用特定格式。他们只需选定文本，然后从样式菜单中选择特定样式即可，这样就可以保证该段文本在屏幕和打印页上都具有相同的格式。

对于诸如 PageMaker、QuarkXPress、FrameMaker，以及 Microsoft Word 等字处理程序而言，屏幕上所显示文本格式与打印时的文本格式几乎相同。不像老式字处理器（在老式字处理器中，格式化命令由需要进行格式化处理的文本之前和之后的格式字符串组成），现在的字处理器程序以二进制形式存储格式化指令。然而，如果你在不支持 Word 格式化命令的字处理器中打开 Microsoft Word 文件，你会在文件头看到许多难以理解的文本（这是格式化命令的产物），之后才可能是真正的正文。

许多软件包有内置的翻译器，它们能够读取和转换其他格式的文件（包括理解其他的样式），并且那些翻译器也存在于其他软件包。然而，在文件从某格式（比如 Microsoft Word 格式）转换成其他格式（比如 WordPerfect）之后，文件被无情地改变了。文件格式也从第一种格式变成另外一种。样式不同，所以尽管文档中文本的角色也许相同，但是显示效果不同。

然而，一些标准化的文档格式类型，比如 Rich Text Format（RTF）实际上会显示控制文本样式的指令——与在老式字处理器查看格式化信息的方式一样。RTF 使用一组格式化文本的命令，如果其他程序理解这些命令，并能解释 RTF 文件，则这些文件可以在不同字处理器程序之间，甚至不同应用程序之间实现共享。如果你细细审查某个 RTF 文件，则能够对格式化语言如何解释有自己的认识，也能够理解格式化语言在屏幕或者打印机上的显示格式化文本。对于 RTF 而言，文本集（在技术上称做 ASCII 代码）代表着文本的格式化命令或者标记。下面就是一个 RTF 标记语言如何格式化一段简单文本的例子：

```
\keepn\par\sb240\b0
Now is the time for all good men to come to the aid of their country.
```

正如你看到的那样，每个标记命令都是以一个反斜杠开始的。在这种格式化语言中，这些命令可以在一行中输入完，格式化标记告诉显示软件或者打印机如何对格式化标记下面的文本段落进行格式化处理。对于上例而言，那些格式化标记执行如下功能：

- \keepn——让显示软件或者打印机在显示文本时使本段文本与下段文本紧挨着。
- \par——告诉显示软件或者打印机重开一个新段落来显示如下文本。
- \sb240——在段落前空出 12 点的空格。
- \b0——关闭粗体格式。

由于本段文本使用 RTF 格式进行格式化处理，这种格式包含一组格式化规则和命令，对于任何拥有 RTF 解释器的字处理器或者其他应用程序而言，它们都能正确地显示和处理如上文本。即使如上文本是在 Macintosh、主机或者 Windows PC 等不同平台创建的，也不会影响解释器对它的正确理解。如将上述文本保存到 ASCII 文件，只要用于查看该文件的应用程序（可能是字处理器、浏览器，甚至工作表格处理器）包含有 RTF 解释器，则段落显示的格式就会统一。

标记的概念

RTF 负责屏幕上和打印机产生的文本效果。之所以能够实现这个目标,主要是因为 RTF 文件中存放有相应格式化的命令。如果你想变换文本显示的外观,则必须打开文本文件,对格式化命令进行修改。比如,你不可能在不打开文件的前提下,对文件文本实施不同的规则集,把所有文本的格式变成为 24 点字号,粗体。你不能在不同的字处理器下使用不同的方式处理文件中的数据,因为结果肯定不一样。同时,尽管你可以创建多种样式,但是这些样式并没有绑定到文本中特定部分内容。比如,你可以对所有的段落应用 Code 样式,但是并非所有应用该样式的段落的外观都一样。为什么呢? 原因是这些段落可能同时被应用了其他格式。比如,文档作者也许会给已经应用了 italics 字型的部分段落应用粗体格式。RTF 允许这样实施格式化,原因是它并未要求文本严格遵守某种特定的规则。

因而,尽管 RTF 能够描述文本怎样格式化,但是对于更复杂的格式化问题则比较难以应付。RTF 格式有太多例外,并且它不能提供定义整个文档的真实结构。RTF 和其他此类的格式化语言只能简单地帮助显示信息,而不能提供文档更详细的结构信息。在前面的代码范例中,惟一真正描述文档结构的命令是 \ par(paragraph),该命令说明接下来的文本将单独成为一个段落。

相对于 RTF 之类的格式化语言而言,标记语言在结构上要复杂得多。诸如标准通用标记语言(SGML)和可扩展标记语言(XML)之类的标记语言在功能和复杂性方面要远远超越了简单的格式化语言,能够创建文档的明确、详细的结构。与格式化语言相比,标记语言具有如下鲜明特征:

- 标记语言会描述文档中文本的结构。许多明确的规则可以决定特定文档结构的开始和结束地点。这些明确的规则创建了一个定义良好的、近似树型或者分级型的结构,样式表借助于该结构图就可以显示信息。
- 内容与格式相隔离。格式通常是借助于样式表来处理的,而后者是单独的指令,可以描述文档中不同部分怎样格式化。借助于样式表,你可以在无须改动内容的前提下更改格式。
- 在设定特定元素之后,格式更改会影响该元素的所有发生实例。

像格式化语言一样,标记语言需要能够读取用各种格式——也称做标记元素——描述的文本的应用程序,并且还能够显示或者处理相应文本。

当你使用相同的标记元素集合来描述文档的内容时,该文档可以被许多不同类型的软件处理,其中每个软件可以应用不同的处理指令到标记的文本。比如,基于 Web 的财务分析处理器也许只从报告文档中抽取数字类数据,并将这些数据显示在屏幕上。然而,一个基于页布局的字处理器也许不仅需要处理文本格式信息,而且还要能够处理所有页眉和页脚信息,要收集所有的打印信息,并在文件合适的地点插入页号和脚注。借助于标记语言,比如 SGML,不同的处理指令可以与相同文件建立关联。

诸如 RTF 之类的描述性语言重点在于格式化和显示整个文件。在另外一方面,标记语言会映射文档的结构,根据其功能确定(实质上是映射)每个特定文档结构,并在每个将显示该文

档的设备上的样式表上留下格式化信息。

借助于诸如 XML 之类的结构化标记语言，你也将拥有更多的搜索灵活性。你不仅可以在标准 Web 页面中搜索任何单词，而且可以在某文档的各种元素中搜索。比如，你也许只想搜索页眉信息，或者只想在特定标题中搜索，比如本书的章节标题中搜索某关键字。

1.2 通用标记语言：HTML、SGML 和 XML

在前面我们已经讨论过标记语言，也许你对其中部分标记语言已经比较熟悉，但是我还想更详细地介绍一下各种通用标记语言。

1.2.1 HTML

我们首先介绍 HTML。你也许知道，HTML 是个相对比较容易使用，且实现也相对较快的工具；但是，尽管它比较简单，但功能不弱，足以提供以 Web 页面方式或 HTML 格式的 e-mail 方式在 Internet 上传递文档的能力。它使用元素标记来标定有限的文档结构，比如页面头和页面体。并且在相同文档中，HTML 元素也定义了在 Web 页面上显示文档的方式。尽管从技术上来讲 HTML 是一种标记语言，但是实际上，它被用作格式化语言，以在 Web 浏览器中规范内容的显示效果。

插件程序向 HTML 提供了一种近似于 XML 所提供的可扩展性的扩展功能。借助于插件程序，用户具有在 Web 页面中查看不同类型数据的能力。插件程序是用其他语言编写的外部程序，通过在 HTML 文档中引入 HTML object 对象就可以在 Web 文档中实现。借助于 object 元素，Web 开发者就可以在 Web 页面中嵌入非 HTML 和非文本数据，比如多媒体演示、音频和视频文件。Object 元素的属性可以确定该对象的文件类型（movie，Macromedia Flash 演示，等等）。根据文件类型不同，浏览器会添加相应特定的插件程序，以使浏览器显示组件可以处理该文件。如果用户没有合适的插件程序，浏览器会提示用户来指定他或她本地计算机上其他应用程序，以便于读取和显示该文件。

插件程序的问题是开发这种程序的唯一目的是显示专有数据。这种专有特性意味着可以显示 Adobe Acrobat 文档的插件程序很可能不能显示 Excel 工作表数据。除此之外，插件程序一般只着重数据显示，而对数据的结构并不关心。因而，它们也许不能认识到某种结构化元素。插件技术类似于创建和使用专用浏览器元素。这两种方式（插件技术和专用浏览器元素）都意味着并非每位用户都能够正确获取数据。

1.2.2 SGML：HTML 和 XML 的祖先

SGML 是种广泛应用于高端信息出版领域的标记语言。你也许会看到用于技术写作的 SGML，在技术写作领域中对跨平台处理复杂的、大型文档的需求比较强烈。SGML 也常应用于如下领域：自动化工业，医疗保健，通讯工业，以及任何大量文本需要，以易于访问格式进行结构化处理的领域。

SGML 已经使用多年，甚至在 1986 年成为国际标准化组织（ISO）的标准之前已经在许多方面有所应用。SGML 拥有广泛的支持，原因是其具有许多很适于基于文本应用程序的特性。