

Practical Ajax Projects with Java™ Technology

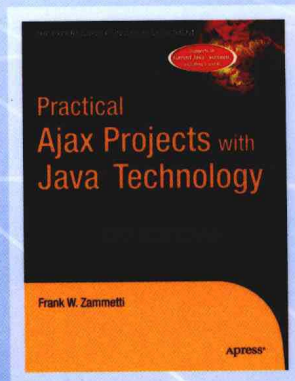
Java Web 2.0

项目开发实战

[美] Frank W. Zammetti 著

许劲松 章曦 周斌 译

- 跟世界级专家现场学习Web 2.0/Ajax开发
- 自己开发Gmail、Flickr、网摘和纯Ajax游戏
- 综合运用Struts、Prototype、DWR、Dojo等众多最炫的技术



TURING

图灵程序设计丛书

Web开发系列

Practical Ajax Projects with Java™ Technology

Java Web 2.0

项目开发实战

[美] Frank W. Zammetti 著

许劲松 章曦 周斌 译

人民邮电出版社

北京

图书在版编目 (CIP) 数据

Java Web 2.0 项目开发实战 / (美) 扎梅蒂 (Zammetti, F.W.) 著; 许劲松, 章曦, 周斌译. —北京: 人民邮电出版社, 2008.12

(图灵程序设计丛书)

书名原文: Practical Ajax Projects with Java™ Technology

ISBN 978-7-115-18865-6

I. J… II. ①扎…②许…③章…④周 III. ①JAVA 语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (2008) 第142668号

内 容 提 要

本书通过对一些有趣实用的 Java Web 开发项目进行由浅入深的分析, 引导开发人员了解基于 Java 的 Ajax 在实际应用中的各种方法和技术, 并帮助开发人员从传统的开发方式逐步平稳地转换到 Web 2.0 开发方式中。

本书不仅可作为 Java Web 开发新手的实验教程, 对于中级 Java Web 开发人员来说, 也是一本实用的参考手册和进阶指导材料。

图灵程序设计丛书

Java Web 2.0项目开发实战

- ◆ 著 [美] Frank W. Zammetti
译 许劲松 章曦 周斌
责任编辑 傅志红
执行编辑 谢灵芝
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京顺义振华印刷厂印刷
- ◆ 开本: 800×1000 1/16
印张: 27.75
字数: 656千字 2008年12月第1版
印数: 1-4000册 2008年12月北京第1次印刷

著作权合同登记号 图字: 01-2006-7031号

ISBN 978-7-115-18865-6/TP

定价: 59.00元

读者服务热线: (010)88593802 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *Practical Ajax Projects with Java™ Technology* by Frank W. Zammetti, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705.

Copyright © 2006 by Frank W. Zammetti. Simplified Chinese-language edition copyright © 2008 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

谨以此书献给我的妻子Traci，也许她对书中的内容一点也不懂，但她给了我极大的鼓励，这让我领悟到自己为何如此爱她。

献给我的两个孩子Andrew和Ashley，我乐意为了他们而长期伏案工作。感谢他们“撒下”爸爸这么久（经常如此），让我可以完成这项工作，我亏欠他们太多了。（希望他们读到这里的时候，能够明白其中的愧疚之意！）

献给我的父母，所有这一切都源于他们1982年圣诞节（也许是1983年，20多年后我的记忆有些偏差！）一次看似不经意的购物行为，他们购买了一台Timex Sinclair 1000计算机。

献给John J. Sheridan，感谢他对我自始至终的支持。

前言

谁能够想象到某些人把一个著名足球俱乐部的名字运用到了一项即将改变世界的编程技术之上？然而这的确发生了！

Ajax（异步JavaScript和XML）声势浩大地席卷了Web开发领域，成为Web 2.0大潮的催化剂。它不仅仅开启了Web应用的一个崭新时代（因为它提供了前所未有的人性化的用户界面），而且彻底改变了Web开发的方式。我们不再局限于文档/页面模型，用户驱动模型现在终于成了现实。也许是第一次，我们终于可以利用自己的既有优势（因为，正如你将要了解到的，Ajax实际上并不是什么新东西），开发出真正的Web应用了。这真是太酷了！

是的，对Ajax的宣传一度充满了泡沫。在网络时代，许多类似的东西都如昙花一现，来得快，去得也快。但是，从载入史册的那一天，一个聪明人为大家长期以来一直使用的技术想出一个别致的名字到现在，已经几年过去，我们依然前行。有人会说，Ajax没有什么真正的思想，只不过是有了个好名字而已。真是那样吗？

直至今日，Ajax的宣传声势仍然不减，这一事实说明，Ajax可能的确有过人之处。实际上，当人们从最初的惊喜和营销的迷雾中走出来，真正开始理解并应用一种技术，认识到其中实实在在的好处时，炒作的成分也就所剩无几了。Ajax不仅仅只有一个好名字，它是有实货的技术。这一点是毋庸置疑的！

我编写本书的一个重要原因，就是让编程人员了解，为什么Ajax不只是一时的流行货，而是一种大变革，我们必须在实战中去了解它。我们阅读了所有想要的白皮书，琢磨了所有能找到的介绍性文章，然而我们仍然感到困惑：“OK，我理解了它的原理，但如何实际运用它呢？”我写这本书的目的就是向编程人员展示如何将Ajax巧妙地运用到真实的应用程序中。

本书针对的是编程人员，他们更喜欢通过真实代码进行学习，更喜欢对代码进行分析和运用。本书并不是为理论家或是那些不再经常编程的家伙们写的。这并不是说架构师（我本身就是一个架构师，比较幸运的是我还能编写代码！）不能从这本书中学到任何东西。他们当然同样能学到东西！但是，我相信那些身处编程第一线的开发人员才是本书最大的受益者！

那么，你为什么阅读这本书呢？首先，本书中有一款很不错的游戏！我并不是在谈论一个隐藏的复活节彩蛋游戏……马上跳到第10章吧，我会等着你。

看到了吗？我并没有开玩笑！

此外，书中还有6个各不相同的项目，每一个项目都展示出Ajax是如何让Web开发更完善、更丰富、最终更动态的。书中将介绍许多方法和大量不同的库，这样当你开发自己的Ajax作品

时，就能够判断哪一种方式适合你以及哪种情况对你最有利，此外，书中有现成的示例帮助你起步。

从个人角度出发，我不得不说，编写本书让我体会到了极大的乐趣。因为这是我的第一本书，所以写作中确实有痛苦的经历，但这是相对的。的确，我倾注了大量心血，在编写本书的过程中，我的睡眠习惯被彻底打乱了！但这是非常值得的。我坚信任何阅读本书并对书中描述的应用进行研究的开发人员都会收获丰富的知识和经验，而且我认为这样做也是非常愉快的。如果你阅读本书非常愉快并且从中学到了一些东西，那么我的使命就完成了。

本书概览

第1章对Web开发的发展历程进行总体介绍，并会特别介绍Ajax。

第2章介绍了掌握Ajax必须了解的要素：JavaScript、CSS、DOM脚本及XML。

第3章讨论服务器端的一些东西，简要介绍Ant、Apache Tomcat、Web应用（Webapp）、servlet和JSP，以及关于XML的一些知识。

第4章从Karnak开始介绍实际项目，这个项目和Google Suggest有些类似。这一章会介绍JWP（Java Web Parts）中的AjaxTags。（如果你对它或是这里提到的其他概念不甚了解，请不用担心，我们会讲的！）

第5章介绍的是一个webmail客户端，与Google的Gmail类似。在这个项目中，会介绍一个叫DWR的出色的Ajax库。

第6章介绍一个相当典型的Ajax应用程序：RSS提要阅读器。这个项目又一次示范了JWP中AjaxTags的使用方法。

第7章介绍的是一个名为PhotoShare的应用程序，利用它可以创建相册并且与亲朋好友分享。这个应用程序演示了一些非常好的动画技术，并且介绍了另一个库：Dojo。

第8章介绍的是一个名为Organizer的项目，它是一个基本的PIM应用程序（即备忘录、任务、约会和联系人）。这个项目中，你会了解到一个叫WebWork的流行应用程序框架以及Spring框架的一些知识，还有最流行的Ajax库之一Prototype。

第9章是一个基于Ajax的聊天应用程序，直接命名为AjaxChat。它是建立在最著名的Struts框架之上的，采用“纯粹的”Ajax技术，不需要使用任何库。

最后，第10章是一款名为AJAX Warrior（AJAX勇士）的游戏。它也是利用“纯粹的”Ajax建立起来的，这一章还会介绍一种经常出现在Ajax应用程序中的名为JSON的流行数据格式。

获取本书的源代码

本书中所有的示例都可以从Apress网站的Source Code（源代码）区域免费获取（也可到图灵网站www.turingbook.com本书网页免费注册下载）。事实上，针对本书自身的特点，在学习第4章之前，你绝对有必要下载源代码。访问www.apress.com，点击Source Code链接，在列表中找到*Practical Ajax Projects with Java™ Technology*。从本书的主页面就可以下载源代码的压缩文件，

这些源代码是按章编排的。

获取本书的最新资料

写书是一项非常浩大的工程，比我当初想象的还要浩大！我并不是完人，也和其他人一样会犯错误。让我提前对可能出现在本书的任何错误向你表示歉意。我坚信参与本书创作的每一个人都尽了最大的努力保证里面没有错误——但是，我们必须面对现实，我们以前都曾经读过技术书，明白现实总是残酷的。抱歉，抱歉，抱歉！

从Apress网站（www.apress.com）的本书主页上可以获取最新的勘误表，并且还有关于你发现错误后如何通知我们的信息。

与作者联系

我非常期待听到你关于本书内容和源代码示例的问题和评论。请直接通过 fzammetti@omnytex.com（垃圾邮件将会被Sentinel捕获并处理）与我取得联系。我会尽快对你的询问给予回复。

致谢

许多人以各种不同的方式帮助我完成了这本书，而他们中的某些人甚至没有意识到这点！我会尽量记住他们，但这并不可能，所以我要提前向他们表示道歉！

首先，我要感谢Apress出版社的每位员工，他们让本书变成了现实，感谢他们承担风险出版了一本发行量无法预知的图书。没有什么可表达的，只有将感激的话语送给曾与我直接交流的每一个人，以及整个出版流程涉及的每一个人。感谢他们让本书顺利出版！

我要特别感谢本书的编辑Chris Mills和项目经理Richard Dal Porto。他们二位指引我跨越了一个完全陌生的领域，并让这个过程变得非常愉快，我不知该如何感谢他们！我知道，他们都只是在做自己分内的工作，正是这样才使所有的一切更加令人感动。

非常感谢Herman van Rosmalen，他是我在JWP（<http://javawebparts.sourceforge.net>）项目中最忠心的合作伙伴之一，是本书的技术审稿人。我知道他为了让我保持实事求是的作风投入了大量的时间和精力，对此我十分感激！即使当我们在某些事情上无法达成一致的时候，他也总是让我冷静思考，并换一个角度考虑问题。现在，我们还要一起合作，一起整理JWP列表中的那些繁杂条目。

还有一份特别的感谢要送给Anthony Volpe，这位优秀的艺术家为本书制作了插图。他和我已经是十几年的朋友了，我们合作过许多项目，包括三款掌上电脑游戏（可以访问www.omnytex.com查询它们）、许多Flash游戏以及一些Web动画片。他是一位具有惊人创造力的了不起的艺术家，而且是一个益友，相信他自己也了解这一点。

我要感谢Wendy Smoak，在我写书的过程中，他忍受了我许多激烈的言辞。他还曾帮忙解决了几个技术难题，这些当然是要十分感谢的，不过更要感谢的是，当工作陷入困境时，他能够

听进去那些恶言恶语！

我还要感谢本书中用到的一些库的创建者，包括Joe Walker (DWR)、所有完善Dojo的人们以及Sam Stephenson (Prototype)。

最后，同样重要的是，我要感谢所有购买本书的人们！我真诚地希望你们阅读本书的乐趣和我写书时一样大，希望你们感到物有所值，从而证明这是一次具有教育意义并且开阔眼界的体验。

我曾说过，我肯定会忘了感谢许许多多的人，那么就让我感谢整个世界吧。

第一部分 使用 Ajax 和 Java 编程

第 1 章 Ajax: 勇闯新世界	2
1.1 Web开发简史: “经典”模型	2
1.2 经典Web的问题	7
1.3 走进Ajax	9
1.4 为什么说Ajax是一场革命? 通向RIA之路	13
1.5 第一个Ajax示例	15
1.6 Ajax工具包的选择	22
1.7 Ajax的替代方法	22
1.8 小结	25
第 2 章 Ajax 的支柱	27
2.1 JavaScript基础	27
2.1.1 变量、作用域和函数	29
2.1.2 关键字	31
2.1.3 空白	32
2.1.4 分号	32
2.1.5 注释	33
2.1.6 字面量、数组和对象初始化程序	33
2.1.7 数据类型	34
2.1.8 匿名函数	39
2.1.9 内存管理	40
2.2 面向对象的JavaScript	40
2.3 DOM和脚本编程: 在运行中操纵内容	46
2.4 XML: 描述语言的语言	51
2.5 在JavaScript中解析XML	53
2.6 CSS	60
2.7 小结	68
第 3 章 服务器端	69
3.1 项目开发环境	69
3.2 Ant: 以开源的方式构建Java应用程序	70

3.3 Apache Tomcat	77
3.4 Web应用、servlet、JSP和在服务器上解析XML	80
3.4.1 Web应用	80
3.4.2 servlet	85
3.4.3 JSP	87
3.4.4 在服务器上解析XML	89
3.5 安装和使用应用程序	93
3.6 小结	94

第二部分 开发项目

第 4 章 Kamak: 输入前提示	96
4.1 需求和目标	96
4.2 如何实现	97
4.3 最终呈现	102
4.4 研究解决方案	103
4.4.1 客户端代码	104
4.4.2 服务器端代码	113
4.5 推荐练习	120
4.6 小结	121
第 5 章 InstaMail: 基于 Ajax 的 webmail 客户端	122
5.1 需求和目标	122
5.2 如何实现	123
5.3 最终呈现	126
5.4 研究解决方案	129
5.4.1 客户端代码	131
5.4.2 服务器端代码	152
5.5 推荐练习	174
5.6 小结	175
第 6 章 AjaxReader: Ajax 图书必备内容	176
6.1 需求和目标	176
6.2 如何实现	177

6.3 最终呈现	179	8.4.2 服务器端代码	299
6.4 研究解决方案	182	8.5 推荐练习	310
6.4.1 客户端代码	183	8.6 小结	310
6.4.2 服务器端代码	201	第9章 AjaxChat: 以 Ajax 方式聊天	311
6.5 推荐练习	219	9.1 需求和目标	311
6.6 小结	219	9.2 如何实现	312
第7章 PhotoShare: 分享图片	220	9.3 最终呈现	322
7.1 需求和目标	220	9.4 研究解决方案	324
7.2 如何实现	221	9.4.1 客户端代码	326
7.3 最终呈现	225	9.4.2 服务器端代码	339
7.4 研究解决方案	228	9.5 推荐练习	359
7.4.1 客户端代码	229	9.6 小结	360
7.4.2 服务器端代码	258	第10章 AJAX 勇士游戏: 回到未来	361
7.5 推荐练习	272	10.1 需求和目标	361
7.6 小结	273	10.2 如何实现	362
第8章 Organizer: 管理个人信息	274	10.2.1 多用途的函数和集中式Ajax	362
8.1 需求与目标	274	10.2.2 JSON	365
8.2 如何实现	275	10.3 最终呈现	367
8.2.1 Prototype	276	10.4 研究解决方案	370
8.2.2 WebWork	277	10.4.1 客户端代码	373
8.2.3 HSQLDB	280	10.4.2 服务器端代码	396
8.2.4 Spring JDBC	280	10.5 推荐练习	423
8.3 最终呈现	281	10.6 小结	424
8.4 研究解决方案	284	附录A XMLHttpRequest 对象	425
8.4.1 客户端代码	286	附录B 库、网站和图书	429

Part 1

第一部分

使用 Ajax 和 Java 编程

我从不预想未来——因为它很快就会到来。

——爱因斯坦

我们都听说过一百万只猴子在一百万个键盘上跳可能碰巧打出莎士比亚全集，感谢因特网，让我们知道那是不可能的。

——Robert Wilensky

计算领域永远都需要老调新弹。

——Alan Perlis

因特网就是浪费时间，这正是对它合适的定义。

——William Gibson

拥有计算机最易被忽略的一个好处就是，如果它将事情搞砸了，没有法律禁止你狠狠地敲打它。

——Joe Martin

大部分专家都认为，世界被摧毁的最可能的方式是发生意外。那正是我们所擅长的，我们是计算机专业人士，经常制造意外。

——Nathaniel Borenstein

对于第一次接触Ajax，尤其是刚刚接触Web开发的人来说，本章可以作为了解后面内容的一个不错的热身。然而，已经具有一些经验，尤其是对Ajax并不陌生的开发人员，可以跳过本章，因为对他们来说本章很可能只是一次复习。我们首先介绍过去大约15年以来应用（尤其是Web应用）的发展过程。你会发现，应用程序的基本结构经历了一个有趣的循环。接下来将介绍一些非常不错的Ajax实战示例，并讨论Ajax为什么如此重要，以及它是怎样从根本上改变应用程序开发的。第一个代码示例会以非常简单而基础的方式介绍Ajax。我们还会简单地谈及Ajax的一些竞争技术，以及一些已有的使Ajax更加易用的库和工具包。

1.1 Web 开发简史：“经典”模型

Web刚出现时的确不错。各种各样易记的新单词、词组和术语进入了词典，而且在我们说起这些新词时，感觉它愈发地酷了。随着我们的工作渐渐为人所知，Web应用（Webapp）这个词出现了。这些应用程序有点像回到了分时访问方式的大型机时代。它们无论从使用方式还是外观上都不如随后出现的Visual Basic、PowerBuilder和C++“胖客户端”（虽然它们在Web应用出现后少了，但仍在使用）那样光彩夺目，而且那些胖客户端完成了并仍然在做很多工作，就像那些旧的分时系统一样。

如果认真考虑一下，就会发现应用程序开发其实走了一个循环。如果按时间从前向后梳理一遍，就会发现这一点。

我们就从稍早的时代开始。在这个时代，应用程序采用了大家基本能够理解的形式。首先用简单的终端仿真设备（我们就不讲实际终端啦）来访问远程运行进程。图1-1所示的屏幕就是这类应用程序的典型例子。

TN3270屏幕还是有些意义，因为它们仍然大量被使用，尤其在商业领域中，任何做过一些大型机工作的人都可以作证。有两件有趣的事情需要注意。第一，注意当时UI（用户界面）的简单特性——它们只是文本，通常限制在80列25行，只有极其有限的输入能力——基本上只是可编辑的映射区域。下拉列表、复选框和网格等在这个领域中是完全未知的。如果碰到一个写得很好的应用程序，那么可能幸运地看到如下的实际菜单：

- C. 创建记录
- D. 删除记录

E. 编辑记录

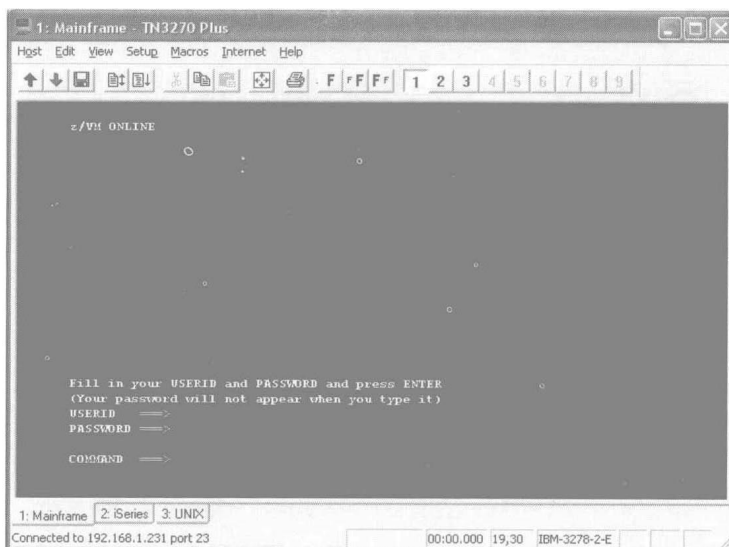


图1-1 TN3270大型机“绿屏”终端显示器

如果不走运，有可能会是如下的信息：

```
..... 01A7C0D9ABABAC00
```

```
..... 89A6B3E34D79E998
```

如果你从来没有用过大型机，那么下面简单地介绍一下。在大型机中编辑文件（文件又叫数据集）通常使用叫做TSO/ISPF的工具。这只是一种文本编辑器，能够在文本显示和十六进制显示间进行转换。上文是内容的十六进制显示，每一行前面的点组成命令区。例如，在以89开头的行上面加入一行，就需要把该行的第一个点替换为i，然后按回车键；如果想删除以01开头的行和刚插入的那一行，就需要把以01开头的行的前两个点换成dd，然后在刚刚插入的行的同样位置也键入dd，再按回车键。（dd用来删除行块，可以使用单个d来删除单行。）

第二点更重要，就是当用户执行了一些需要应用程序响应的动作时会发生什么。很多时候，主机会重绘整个屏幕，包括那些不会因为该操作而改变的部分。每个操作都在主机上执行，基本没有本地处理。甚至对于简单的客户端输入验证也没有本地处理，它仅仅是远程应用程序状态的一个展示。

随着PC的出现，本地处理能力提高了很多，出现了新的趋势。这个时候我们看到，应用程序开始存放在本地而不是中央主机上，至少部分应用程序是在本地执行的。很多时候，整个应用程序本身是在用户使用的机器上运行的。随着微软公司的Windows的流行，“胖客户端”开始被人们所熟知，而它实际上很快成了应用程序开发的事实标准。在这种方式中，UI的功能和用户友好性极大地提高了，而集中的中央硬件在很多地方不再那么重要了（数据库服务器之类的不在此列）。图1-2中所示的屏幕开始成为规范。

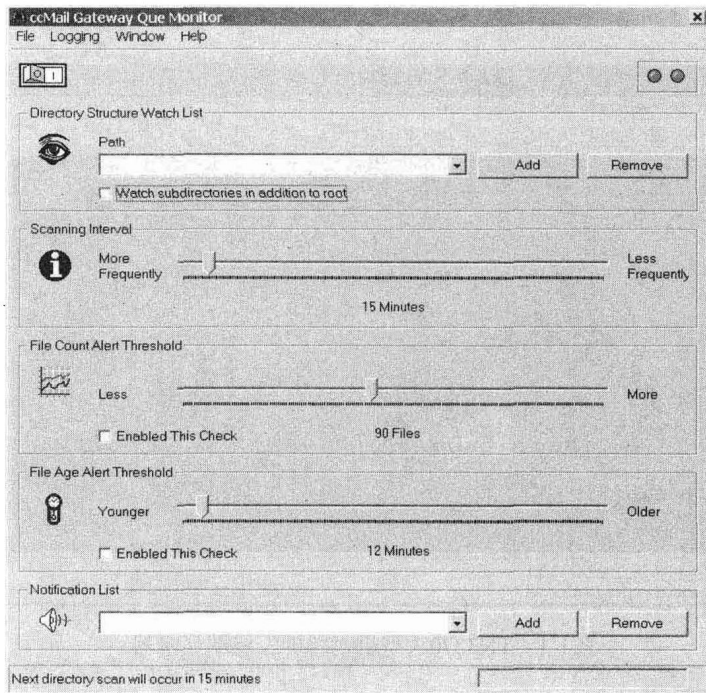


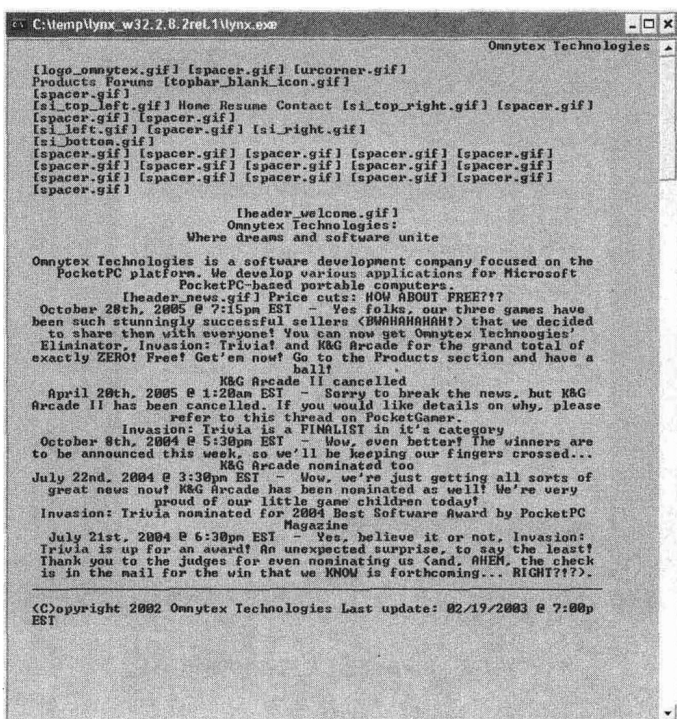
图1-2 典型的“胖客户端”应用程序

注意，图1-2里的UI元素是多么的丰富。显然，这里的UI已有所改进。“改进”当然是相对的，换个环境它就不是更好。可以想象，那些整天低头输入数据的人可能倾向于使用旧的绿屏幕，因为它们对于键盘输入更加高效。不用摆弄鼠标，没有指向和点击，没有必要为了保存记录而把目光从文档上移开。这些固然有理，但无可否认，总的来说，任何时候人们都会选择胖客户端版本而舍弃仅基于文本的版本！你可以试试，找一个现成的主机应用程序，把它和同一应用程序的胖客户端版本并排放在一起，看看有多少用户会选择主机应用程序。给你一个提示：选择主机应用程序版本的人会少于1，但不会是负数。

想一想应用程序是如何工作的，当用户单击按钮，拉动滑块或者单击一个菜单项时会发生什么？多数情况只有部分屏幕需要更新，通常也不需要与外部系统交互。显然胖客户端更加有效，对用户也更加友好。

但是，接下来又发生了什么呢？随着因特网的出现，更明确地说，是由于万维网（World Wide Web，或者简称为Web，记住，Web并不等于因特网）这个组成部分的出现，我们的工作发生了极大变化。随着Web的出现，图1-3所示的屏幕开始变得常见了。

等一下，怎么回事儿？精美的单选按钮、3D按钮、列表框等去哪里了呢？Web的第一代看上去非常像旧的大型机世界。但更重要的是这背后所发生的事情：我们又回到了老的方法，通过在中央机器运行应用程序来实现这些东西，基本上每一个用户的交互都要求重新绘制整个屏幕。



```
C:\temp\lynx_w32.2.6.2ref.1\lynx.exe
Omnytex Technologies

[logo_omnytex.gif] [spacer.gif] [urcorner.gif]
Products Forums [topbar_blank_icon.gif]
[spacer.gif]
[isi_top_left.gif] Home Resume Contact [isi_top_right.gif] [spacer.gif]
[spacer.gif] [spacer.gif]
[isi_left.gif] [spacer.gif] [isi_right.gif]
[isi_bottom.gif]
[spacer.gif] [spacer.gif] [spacer.gif] [spacer.gif] [spacer.gif]
[spacer.gif] [spacer.gif] [spacer.gif] [spacer.gif] [spacer.gif]
[spacer.gif] [spacer.gif] [spacer.gif] [spacer.gif] [spacer.gif]
[spacer.gif]

  [header_welcome.gif]
  Omnytex Technologies:
  Where dreams and software unite

Omnytex Technologies is a software development company focused on the
PocketPC platform. We develop various applications for Microsoft
PocketPC-based portable computers.
  [header_news.gif] Price cuts: HOW ABOUT FREE??
October 28th, 2005 @ 7:15pm EST - Yes folks, our three games have
been such stunningly successful sellers <BWAHAHAHAH!> that we decided
to share them with everyone! You can now get Omnytex Technologies'
Eliminator, Invasion: Trivia! and K&G Arcade for the grand total of
exactly ZERO! Free! Get 'em now! Go to the Products section and have a
ball!
  K&G Arcade II cancelled
April 29th, 2005 @ 1:28am EST - Sorry to break the news, but K&G
Arcade II has been cancelled. If you would like details on why, please
refer to this thread on PocketGamer.
  Invasion: Trivia is a FINALIST in it's category
October 8th, 2004 @ 5:38pm EST - Wow, even better! The winners are
to be announced this week, so we'll be keeping our fingers crossed...
  K&G Arcade nominated too
July 22nd, 2004 @ 3:36pm EST - Now, we're just getting all sorts of
great news now! K&G Arcade has been nominated as well! We're very
proud of our little game children today!
Invasion: Trivia nominated for 2004 Best Software Award by PocketPC
Magazine
July 21st, 2004 @ 6:30pm EST - Yes, believe it or not, Invasion:
Trivia is up for an award! An unexpected surprise, to say the least!
Thank you to the Judges for even nominating us <and, AHEN, the check
is in the mail for the win that we KNOW is forthcoming... RIGHT!?!?>.

<<Copyright 2002 Omnytex Technologies Last update: 02/19/2003 @ 7:00p
EST
```

图1-3 在基于文本的浏览器Lynx中显示omnytex.com

显然我们倒退了一大步。对于每个操作，服务器都重新绘制屏幕并返回给用户。每个用户交互（这里忽略了客户端脚本，因为第一代Web开发人员最开始还无法使用脚本）都要求调用服务器来做繁重的处理。这样或多或少地回到了以大型机方式来处理事情。当然，我们并不是一夜之间忽然变傻了，这么做其实是有一定原因的。原因之一是为了避免“DLL Hell”，该现象出现在由于库版本冲突而导致各种问题的Windows世界中。另一个原因是分布应用程序的需要。让应用程序运行在一台中心服务器上，就可以通过任何带有Web浏览器的PC来访问它，而不需要在每个PC上先安装该应用程序。还有一个不错的原因是，这样一来应用程序开发变得相对容易。至少在一开始，只是用HTML和后台CGI程序来实现相当简单的Web应用，几乎任何人都可以快速、容易地掌握。即使对于那些先前没有做过太多应用程序开发的人，学习过程也不是那么难。

当然，Web成长得非常快！几乎是一眨眼的功夫，就从图1-3发展到了图1-4。

现在，它看起来当然更美观了。除了视觉质量更高外，还有更强大的UI工具调色板可以使用，如下拉列表、单选按钮、复选框等。在很多方面比Web出现以前的胖客户端更好，因为现在“多媒体”已经变得很平常。图形学开始成为我们工作中的一大部分内容，因此屏幕更美观了。

那些讨厌的用户交互怎么办？是的，你猜对了：这个时候还是需要重新绘制整个屏幕。客户端脚本的出现使得一些功能不需要服务器就能实现，但是总体上还是两层结构：可视层和其他部分！框架在一定程度上缓解了问题，但是这更多的只是一个小支流，而不能影响底层模式。

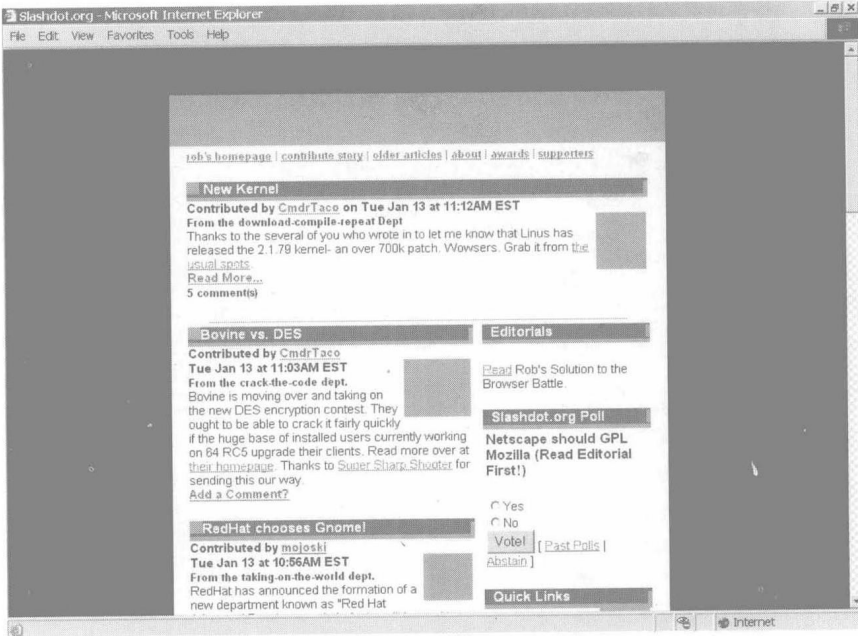


图1-4 Slashdot, 大约在1998年

我们发现自己已经进入处在Web时代了（见图1-5）。



图1-5 “现代” Web: 以omnytex.com为例