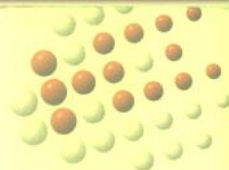
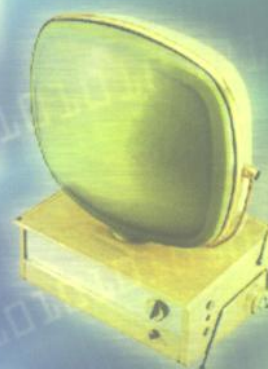


最新 Internet 技术基础与应用系列丛书



实用数据广播技术



(美) Jessica Keyes 著
译友翻译组 译

• *Datacasting:
How to Stream Database over the Internet*



Mc
Graw
Hill

CMP



机械工业出版社

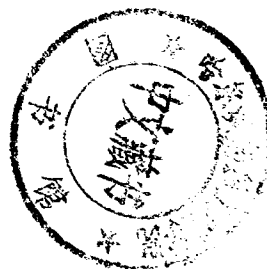
415486

最新 Internet 技术基础与应用系列丛书

实用数据广播技术

(美) Jessica Keyes 著

译友翻译组 译



00415486

机械工业出版社

本书全面介绍了在 Web 上制作页面、站点以及数据库集成的各种最令人感兴趣的产品和技术,其内容主要包括网页制作,一些著名专家有关在网上应用人工智能、数据库等领域的成果介绍,一些著名公司有关 Web 开发的产品和技术,如何在 Web 上重用 COBOL 编写的多种应用,以及将软件工程的成果成功地应用于 Web 开发的技术和工具等等。

全书内容详尽,覆盖面广又独立成章。适合于对 Web/Internet 开发具有浓厚兴趣的软件开发人员、计算机爱好者,同时也是一本综合性极强的参考大全。

Jessica Keyes: Datacasting: How to Stream Databases over the Internet.

Authorized translation from the English language edition published by The McGraw-Hill Companies, Inc.

Copyright 1998 by The McGraw-Hill Companies, Inc.

All rights reserved.

本书中文简体字版由机械工业出版社出版,未经出版者书面许可,本书的任何部分不得以任何方式复制或抄袭。

版权所有,翻印必究。

本书版权登记号:图字:01—98—0731

图书在版编目(CIP)数据

实用数据广播技术/(美)凯丝(Keyes, J.)著;译友翻译组译.—北京:机械工业出版社,1998.8

(最新 Internet 技术基础与应用系列丛书)

书名原文:Datacasting: How to Stream Databases over the Internet

ISBN 7-111-06632-4

I. 实… II. ①凯… ②译… III. 万维网-网页 IV. TP3 93.4

中国版本图书馆 CIP 数据核字(98)第 19627 号

出版人:马九荣(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:廉一兵

北京市密云县印刷厂印刷·新华书店北京发行所发行

1998 年 9 月第 1 版第 1 次印刷

787mm×1092mm1/16·25.75 印张

印数:0 001—5 000 册

定价:46.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

前言

这本书是可以从 Web 上得到的。请访问 www.business-america.com/datacasting。

如果您要将 Internet 或 Intranet 和关系型数据库管理系统（或者甚至是面向对象 DBMS）组合起来，那么会得到些什么呢？根据一些数据库和 Web 软件公司的经验，这将会是一个质的飞跃。这正是为什么领先的数据库开发商、Netscape 和其他一些开发商正全力以赴发布产品、开发新技术的原因。

1997 年，所有的主要关系型 DBMS 公司（包括 Informix、Oracle、Sybase、IBM 和 Microsoft）都宣称他们可以让其客户将 Web 技术的优势和数据库组合起来。最终，大家都想要支持繁重的事务处理。直接的目标是将数据库通过新产品更紧密地连到 Web 上，这些新产品可以从 Web 浏览器接受查询，从数据库中抽取数据，以及将结果格式化为 HTML 以便返回给 Web。长期的目标是得到健壮安全的事务处理。

数据库开发商已经确保了三层数据库处理的安全性，他们将 Web 看成是中间件——广泛分布，和平台无关并且很容易使用。

Oracle 和 Sybase 重新销售 Netscape 浏览器和服务器，Informix 将其 Online Workgroup Server 和 Online Workstation 数据库产品和 Netscape 的 FastTrack Web Server 捆绑在一起。

Netscape 的 Enterprise Server 通过 JavaScript API 提供了数据库连接。您可以使用 HTML 文件，在 server 标签中使用 JavaScript API，在该 HTML 文件中使用嵌入式的 SQL 语句，以便在数据库和浏览器窗体之间移动数据或者显示数据。

IBM 提供了许多工具可以利用现有数据库，包括 Net.Data for DB/2 和一系列的“IBM Connectors” for IMS 和 CICS。

Oracle 的 Web Request Broker 允许在 Web 和数据库之间进行实时交互，而 Oracle 的 InterCartridge Exchange 支持服务器上的插件。

Oracle 称其服务器插件为“cartridges”，而 Sybase 使用 Sun 给出的术语“servlets”描述 Web 服务器端的插件，它可以执行高级的数据库处理逻辑，使得终端用户不需要在自己的浏览器中添加“插件”了。以 Web.sql 为基础，Sybase 构建了庞大的 Web 到数据库的中间件。

Web.sql 允许 Web 用户作为客户端，根据 URL 触发数据库。Web 服务器将准备好一个针对 Web.sql 的调用或 URL，然后将用 HTML 编码的数据传送回来。

最后一个大的数据库开发商 Informix 也使用其 Universal Server 开辟了一片天地。Universal Server 被设计用来集成对象技术，使得数据库可以处理任何数据类型。

为什么选择本书

可以将本书看作是一个导游手册，它会带领您参观各种数据库产品和决策。这里会给出许多的选择，您需要选择适合自己的。我在本书所介绍的都是我认为在市场上当前最有趣的 Web 数据库集成产品。这里没有列出所有的产品，但是列出的都是最好的。除了给出了如何使用这些产品的介绍以外，本书还列出了大量的代码和 176 幅图示。

在第1章,我们介绍了一些向Web发布非动态数据的方式。很有趣的是,今天的大多数Web站点都是使用的静态数据。在第2章,我们讨论了两种最流行的数据库发布产品——Information Builders WebFocus 和 Microsoft 的 Visual InterDev。

在第3章中,我们讨论了如何充分发挥Web数据库的效率。根据对许多数据库驱动的Web站点的研究,我认为许多Web数据库是由Webmaster生成的而不是由数据库管理员给出的。在本章中,我给出了一些有关的建议供他们参考。

我邀请了一些专家和我们交流一下他们的想法。Object Design 的 Mark Palmer 在第4章中介绍了如何在Web上发布对象数据库。Larry Harris 博士(人工智能运动的奠基者)在第7章中介绍了我们如何使用自然语言查询Web数据库。如果您想要了解有关SQL数据库连接的内容,请阅读 Dennis Minium (XDB Systems 的产品策略主任)的第8章和 Kingsley Idehen (CEO of OpenLink 的主席)的第9章。如果您想要了解在向Web发布数据的市场上最好的工具之一的話,请阅读 David Isaacson 的第19章,该章介绍了如何使用 Sapphire/Web 将数据粘贴到Web上。

已经有650亿行COBOL代码存在,其中许多代码都成功地访问数据库。如果能够将COBOL连接到Web上,难道这不是个绝妙的主意? Fujitsu 的 Ron Langer 在第20章介绍了相关的内容,并给出了代码。如果您是一位公司的开发者,您会想要阅读这一章,因为 Ron 会告诉您到哪里可以得到一个免费的COBOL开发环境,而节省上千美元。

第22章和第25章介绍了创建健壮的Internet应用的有关内容。这里给出了一些实例分析和对于基本不同技术以及自生成的数据库Web应用的工具集的描述。Hamilton Technologies, Inc. 的 Margaret Hamilton 介绍了称为“先于事实开发(DBTF)”方法。然后, Margaret 和 Ron Hacker 介绍了如何使用DBTF及其实现工具集001,用它们为州政府构建一个事故记录系统。然后,加拿大 Cadeon Strategic Technologies 的 Stephen Dolha 和 Dave Chiste 介绍了他们是如何使用Web和数据库构建一个资源图书馆的远程查询系统。最后 NetBenefit 的 Marc 和 Norman Beaulieu 介绍了他们的依赖于数据库技术的Web站点是如何构建的。

最大的三个数据库开发商都在这里进行了介绍。Sybase 的 Web.sql 在第11章介绍, Oracle Designer/2000 在第17章介绍,而 Informix 的 Universal Web Connect 在第13章进行介绍。

这里也介绍了 IBM 和 Microsoft。Microsoft 在第15章介绍,我们在那里讨论了 Advanced Data Connector,而 IBM 在第12章中介绍,我们还讨论了如何将IMS和CICS连到Web上。

除了最大的三个数据库开发商的工具以外,还有许多流行的数据库连接工具可以使用。在第5章介绍了 Cold Fusion,在第14章介绍了 Progress Software 的 WebSpeed,在第15章介绍了 NetObject 的 Fusion。

没有数据库,但是想要使用数据库,该如何办?第6章中介绍了共享件产品 MiniSql。有了数据库,但是没有钱买中间件,该怎么办?那么请阅读第21章,那里介绍了 WDB,这是一种Web和SQL数据库之间的共享件接口。

现在,我们来开始我们的旅途吧!

目 录

前言

第 1 章 开始	1
1.1 本章简介	1
1.2 使用无格式文件发布数据	1
1.3 第 2 级数据广播——输出到 HTML	11
1.4 一个工具, 许多种数据库方法	12
1.4.1 Web.DATA 示例	13
1.4.2 使用 Web.DATA 和 CGI	16
1.5 结论	17
第 2 章 Microsoft Visual Inter Dev 和 Information Builders 的 WebFOCUS	18
2.1 本章简介	18
2.2 Microsoft Visual InterDev	18
2.2.1 创建简单的 Active Server 页面	19
2.2.2 向 Web 项目中添加外部文件	20
2.2.3 向 Web 项目中添加数据连接	20
2.2.4 Data Form Wizard	21
2.2.5 使用 Database Designer	22
2.3 Information Builders 的 WebFOCUS	27
2.3.1 在 Web 上发布报表	28
2.3.2 从超文本链调用 WebFOCUS	29
2.3.3 从窗体中调用 WebFOCUS	29
2.3.4 运行特别报表	32
2.3.5 特殊的 WebFOCUS 语法	35
第 3 章 提高 Web 数据库的效率	37
3.1 本章简介	37
3.2 E-R 模式	37
3.3 简单正规化的要领	38
3.4 使用反规范化提高性能的技巧	39
3.5 分裂和合并表的要点	40
3.6 索引的作用和影响	41
3.7 如何管理参照完整性	42
3.8 解决控制数据库瓶颈的问题	43
3.9 理解数据结构和数据关系	43

3.10 有效使用联结和子查询	44
3.11 减小 CPU 开销	46
3.12 考虑视图性能问题	47
3.13 最小化在 SQL 环境中使用的资源	48
3.14 确定进行性能调整的需求	49
第 4 章 在 Web 上成功使用对象数据 库	52
4.1 本章简介	52
4.2 商业问题, 技术解决方案	52
4.3 一个动态的、活动 Web 应用的解析	53
4.4 数据库层的可扩充性	56
4.5 在 Web 服务器体系中进行高速缓存的 性能和影响	57
4.6 可缩放性问题	58
4.7 权衡现有系统	60
4.8 小结	60
第 5 章 使用 Cold Fusion	61
5.1 本章简介	61
5.2 入门	62
5.3 Cold Fusion 技术	63
5.4 Cold Fusion 制标语言	64
5.5 Cold Fusion 管理员	65
5.6 选择和输出数据	65
5.7 在 SQL 语句中使用动态参数	68
5.8 创建 HTML 窗体查询前端	68
5.9 在 CFOUTPUT 节中使用参数	69
5.10 插入、更新和删除数据	71
5.11 更新数据	73
5.12 删除数据	75
5.13 Cold Fusion 应用框架	75
5.14 示范应用	77
第 6 章 MiniSQL 和 W3-mSQL 接口	83
6.1 本章简介	83
6.2 MiniSQL 规范	83
6.2.1 Create 子句	83

6.2.2 Drop 子句	84	7.2.3 在 Web 应用中嵌入 English Wizard Web/Server	105
6.2.3 Insert 子句	84	7.2.4 English Wizard Web/Server 体系	106
6.2.4 Delete 子句	84	7.2.5 结论	106
6.2.5 Select 子句	84	7.3 普及数据仓库	106
6.2.6 Update 子句	86	7.3.1 概述	106
6.3 数据库引擎	86	7.3.2 为什么数据仓库很难发布	107
6.4 C 程序设计 API	87	7.3.3 数据仓库查询概貌	107
6.4.1 msqConnect ()	87	7.3.4 指点单击方案的局限性	108
6.4.2 msqSelectDB ()	87	7.3.5 English Wizard 和数据仓库	109
6.4.3 msqQuery ()	88	7.3.6 English Wizard 语义层	110
6.4.4 msqStoreResult ()	88	7.3.7 English Wizard 体系	110
6.4.5 msqFreeResult ()	88	7.3.8 结论	111
6.4.6 msqFetchRow ()	88	7.4 使用 English Wizard	111
6.4.7 msqDataSeek ()	89	第 8 章 使用 Java 访问数据库	116
6.4.8 msqNumRows ()	89	8.1 本章简介	116
6.4.9 msqFetchField ()	89	8.2 Java 为什么如此特殊	116
6.4.10 msqFieldSeek ()	89	8.2.1 Java 是面向对象的	116
6.4.11 msqNumFields ()	90	8.2.2 Java 是和平台无关的	116
6.4.12 msqListDBs ()	90	8.2.3 Java 是动态配置的	116
6.4.13 msqListTables ()	90	8.2.4 Java 是安全的	117
6.4.14 msqListFields ()	90	8.3 有关基于 Java 数据访问的特殊问题	117
6.4.15 msqClose ()	90	8.4 JDBC——明显的选择	118
6.5 mSQL 终端监控器	91	8.5 JDBC 和 ODBC 的差异	118
6.6 mSQL 数据库管理	91	8.6 JDBC 的两层和三层实现	119
6.7 mSQL 数据模式查看器	92	8.7 JDBC 驱动程序的类型	120
6.8 访问控制	92	8.8 JDBC 的替代者	123
6.9 W3-mSQL	93	8.8.1 Java 的 ODBC 实现	123
6.10 Lite 程序设计语言	95	8.8.2 本地 API	123
6.10.1 窗体数据	96	8.9 对象方案	123
6.10.2 变量、类型和表达式	97	8.10 高层接口	124
6.10.3 条件和循环	98	8.11 企业计算的优化	125
6.10.4 用户定义函数	99	8.12 应用: XDB 系统的 JetExpress	126
6.10.5 用户定义库	100	8.12.1 动态 SQL 的问题	126
6.11 其他信息和插件	101	8.12.2 联编 SQL 的 JetExpress 方法	127
第 7 章 在 Internet 上使用自然语言处 理访问关系型数据	103	第 9 章 使用 OpenLink 进行 Java 数据 库互连	129
7.1 本章简介	103	9.1 本章简介	129
7.2 在 Internet 上有效地使用关系型数据 库	103	9.2 Java 语言环境	129
7.2.1 查找正确的数据库	103	9.3 Java 数据库互连标准	129
7.2.2 浏览关系型数据库	105	9.4 JDBC 和 ODBC	130

9.5 OpenLink 多层 ODBC 体系	132	11.2.9 API——在 HTS 文件中使用 Perl 变量和窗体数据	161
9.6 OpenLink JDBC 体系	132	11.2.10 使用 Web.sql 返回非 HTML 数 据	162
9.7 OpenLink JDBC 技术指南	134	11.2.11 使用 HTS 文件和 Perl 脚本返回 数据的例子	162
9.7.1 OpenLink JDBC URL 详解	134	11.2.12 返回 HTTP 头信息	163
9.7.2 OpenLink URL 连接属性	135	11.2.13 设置 cookie 值的例子	163
9.8 OpenLink JDBC 连接 URL 示例	136	11.2.14 URL 重定向的例子	164
9.9 配置 OpenLink JDBC 连接属性	136	11.3 Sybase web.sql 例子	164
9.10 剖析 OpenLink JDBC 连接 URL 翻译	137	11.3.1 HTS 文件中的 Transact SQL 语 句	164
9.11 Java 到 C——桥	138	11.3.2 HTS 文件中的 Perl 脚本	165
第 10 章 Java 的 JDBC 数据库访问协议		11.3.3 执行 SQL 事务语句	165
概述	139	11.3.4 分析窗体数据	166
10.1 本章简介	139	11.3.5 在窗体数据中使用 Perl 变量	167
10.2 SQL 级 API	139	11.3.6 处理错误消息	167
10.3 主要接口概述	140	11.3.7 处理 Perl 脚本	170
10.4 JDBC 驱动程序接口	140	11.3.8 处理服务器结果	171
10.5 数据库连接	141	第 12 章 IBM Web Connectors for IMS 和 CICS	177
10.6 传递参数和接收结果	143	12.1 本章简介	177
10.7 将 SQL 数据类型映射为 Java	144	12.2 IMS Web	177
10.8 异步、线程化和事务	145	12.2.1 IMS Web Studio	178
10.9 游标	146	12.2.2 IMS Web 开发	178
10.10 JDBC 接口	146	12.2.3 在 IMS Web Studio 中创建项目	178
10.11 动态数据库访问	147	12.2.4 自定义生成的代码	180
10.12 动态类型化数据访问	147	12.3 IMS Web 遗留系统重用示例	181
10.13 示范 JDBC 程序	149	12.3.1 修改输入 HTML	184
10.14 JDBC 可下载资源	151	12.3.2 修改输出 HTML	184
10.15 JDBC-ODBC 桥驱动程序	151	12.4 IBM CICS Internet 插座	186
第 11 章 Sybase 的 web.sql	152	12.4.1 CICS Gateway 的作用	189
11.1 本章简介	152	12.4.2 配置 Web 服务器	189
11.2 Sybase web.sql 概述	154	12.4.3 启动 CICS Gateway	190
11.2.1 在 HTS 文件中使用 SQL	155	第 13 章 INFORMIX-Universal Web Connect	219
11.2.2 使用 SQL 查询数据库	155	13.1 本章简介	219
11.2.3 使用 SQL 执行其他的数据库操 作	156	13.2 Web 上的数据库应用体系	219
11.2.4 在 HTS 文件中使用 Perl 变量和 窗体数据	157	13.3 解决方案概观	221
11.2.5 访问 HTS 文件中的 HTML 窗体 数据	157	13.4 技术概观	222
11.2.6 HyperText Sybase (HTS) 文件 格式	159	13.5 INFORMIX-Universal Web Connect	
11.2.7 访问 HTS 文件	160		
11.2.8 在 HTS 文件中使用 Perl	161		

组件	223	15.5.1 构建站点的步骤	256
13.6 使用 Universal Web Connect	224	15.5.2 设计页面	256
13.7 INFORMIX-Universal Web Connect API	226	15.5.3 修改站点样式	258
13.8 组件详细信息	227	15.5.4 数据库发布	259
13.8.1 Web 浏览器	228	15.5.5 预演和发布站点	261
13.8.2 Web 服务器	228	第 16 章 Netscape 的 LDAP 协议和目录 服务	263
13.8.3 Web 驱动程序模块	228	16.1 本章简介	263
13.8.4 终端用户应用	228	16.2 概述	263
13.8.5 共享内存	229	16.3 Internet 和 Intranet 目录	264
13.8.6 配置文件	229	16.4 LDAP 协议和目录服务	265
13.8.7 日志文件	229	16.5 目录操作	267
13.8.8 Universal Web Connect 服务	229	16.6 目录拓扑、Referral 和复制	267
13.8.9 wcstat 程序	229	16.7 拓扑限制和挑战	268
13.8.10 浏览器连接的类型	230	16.8 目录 API	268
第 14 章 Progress Software 的 WebSpeed: 确保 Internet 上事务处理的安 全性	233	16.9 外部表示和交换格式	270
14.1 本章简介	233	16.10 登录、口令和访问控制	270
14.2 挑战	234	16.11 目录和 Netscape Navigator	271
14.3 安全性和完整性解决方案	235	16.12 目录和分类服务器互操作性	272
14.4 CGI 脚本	236	16.13 目录和其他服务	272
14.5 高级安全性特性	236	16.14 两个附加的例子	272
14.6 Internet 事务处理系统	238	第 17 章 Oracle Designer/2000: 在 Web 上发布应用	276
14.7 WebSpeed 方法	239	17.1 本章简介	276
14.7.1 WebSpeed 和安全性	240	17.2 Oracle WebServer 概述	276
14.7.2 WebSpeed 安全性小结	241	17.3 Designer/2000 WebServer Generator	276
14.7.3 未来的 WebSpeed	241	17.4 逆向工程和生成用于 Web 的应用	277
14.8 Internet 安全性术语词汇表	242	17.5 生成的 Web 应用	281
14.9 其他信息索引	243	17.6 WebServer 生成器	284
第 15 章 使用 NetObjects Fusion	245	17.7 Oracle WebServer 生成器的工作原 理	286
15.1 本章简介	245	17.7.1 模型设计	286
15.2 NetObjects Fusion 主要视图	245	17.7.2 生成处理	286
15.3 Assets 视图	246	17.8 模型定义和数据使用	286
15.4 Publish 视图	247	17.9 生成的应用	287
15.4.1 发布数据	248	17.10 生成的页面	288
15.4.2 创建数据对象	249	17.11 启动页面	288
15.4.3 为外部数据创建数据对象	250	17.12 查询窗体	289
15.4.4 创建数据列表	252	17.13 记录列表页面	289
15.4.5 创建堆叠页面	255	17.14 视图窗体	290
15.5 使用 Fusion	255	17.15 HTML 格式化	290

第 18 章 Microsoft 的 Advanced Data Connector	292	21.1 本章简介	338
18.1 本章简介	292	21.2 概述	338
18.2 详细构件视图	295	21.3 如何使用 WDB	339
18.3 详细控制流	295	21.4 Form 模式	339
18.4 构建 ADC 应用	298	21.5 Query 模式	341
18.4.1 连接到数据库	299	21.6 Default 模式	342
18.4.2 运行 SQL 脚本	300	21.7 将 WDB 移植到其他的数据库系统	342
18.4.3 安装和验证数据库	300	21.8 安装	346
18.5 HTML 框架	302	21.9 写入窗体定义文件	347
第 19 章 使用 Sapphire/Web 向 Web 发布数据	306	21.10 窗体定义文件语法	348
19.1 本章简介	306	21.10.1 内部变量	351
19.2 Web 对于数据库的挑战	306	21.10.2 函数	351
19.3 各种用于数据/Web 集成的方法	307	第 22 章 使用 OOI 开发更好、更快和更便宜的基于 Internet 的应用	353
19.4 Sapphire/Web 和 Bluestone 简介	307	22.1 本章简介	353
19.5 什么是 Sapphire/Web?	308	22.2 为什么软件会失败	354
19.6 Sapphire/Web 界面	308	22.3 “纠正错误”	355
19.7 Sapphire/Web 体系概述	309	22.4 系统和开发难以控制	356
19.8 使用 Sapphire/Web	311	22.5 事情如何发展到如此地步	356
19.8.1 步骤 1: 创建新的项目	311	22.6 一种完全不同的方法	357
19.8.2 步骤 2: 创建 HTML 页面	311	22.7 DBTF 哲学	358
19.8.3 步骤 3: 使用 DSQL 创建数据库查询	314	22.8 对面向系统对象 (SOO) 进行模型化	361
19.8.4 步骤 4: 联结处理	315	22.9 开发过程	363
19.8.5 步骤 5: 发布完成的 Web 应用	320	22.10 不再需要	365
19.9 运行时刻应用	320	22.11 为 Internet 而开发的 OOI 应用: 三个实例研究	366
19.10 Sapphire/Web 特性	322	22.12 结论	366
19.10.1 数据库访问特性	323	第 23 章 州高速公路部门交通管理	
19.10.2 体系结构特性	323	的事故记录系统	368
19.10.3 Sapphire/Web 应用服务器	325	23.1 本章简介	368
第 20 章 为什么使用 COBOL 开发 Web 数据库应用?	326	23.2 道路安全工程处理	369
20.1 本章简介	326	23.3 ARS 系统	371
20.2 访问 SQL 数据库	331	23.4 用于修改/演化/维护的设计	372
20.3 将现有的 CICS 应用转换为基于 Web 的应用	334	23.5 使用 FMap 和 TMap 的螺旋式开发过程	373
第 21 章 WDB: Web 和 SQL 数据库之间的一个共享件接口	338	23.6 系统体系	374
		23.7 用户接口	375
		23.8 查询/报表实例	377
		23.9 OOI 客户/服务器分层	378
		23.10 数据库设计	381

23.11 结论	382
第 24 章 Resource Librarian Remote Query 系统	384
24.1 本章简介	384
24.2 开发策略	385
24.3 开发方法	386
24.4 远程查询设计目标	388
24.5 系统体系	388
24.6 革新与挑战	394
24.7 小结	394
第 25 章 面向系统解决方案: Enterprise	

Management Architecture 方 法	395
25.1 本章简介	395
25.2 方法	395
25.3 Enterprise Management Architecture™	396
25.4 校友开发 Internet 应用	398
25.5 NetBenefit Web 站点	398
25.6 独特特性和优势	399
25.7 结论	400

第1章 开 始

1.1 本章简介

数据，到处都是数据，简直就是数据的海洋。

对于 Web，的确是这样，直到今天仍然如此。在过去，Web 的世界只是由相当普通的静态数据组成。直到移植工具 (porting tool) 和自动 HTML 生成器出现，基于数据库 (database-derived) 的数据才结束了其小心翼翼而又充满艰辛的手工编码时代。幸运的是，今天的 Web 是一个全新的“体育运动”世界。实际上，这就是本书要介绍的内容。我们会介绍当要将数据连接到 Web 上时，如何开始这个“游戏”。

今天，数据以各种形式出现。一些是以文件的形式，一些是以数据库的形式，而另一些以数据仓库 (data warehouse) 的形式。从本质上说，你的目标是要在 Web 上广播其中一些或者全部数据。这就是本书为什么叫“数据广播 (datacasting)”的原因，即：数据 + 广播 = 数据广播 (data + broadcasting = datacasting)。

数据广播需要你做出一些非常重要的决策。首先是个十分重大的决定。你是否真正需要数据库？将企业数据库都挂到 Web 上显然是不太可能的。Web 数据库连接的主要目标是在一次连接中尽可能做更多的事情。但是，系统在进行这样的处理时效率越高，数据返回给用户的速度也越慢。需要考虑的另一件事情是数据完整性 (data integrity)。尽管提交、备份和恢复的概念是数据库管理员要在企业环境中执行的职责，维护数据完整性的工作也是 Web 专家全新的前沿阵地。还有一个问题，你是允许对企业数据仓库本身进行访问，还是对这些数据仓库的副本进行访问，以及如何使得这些副本达到最优。最近我访问了数个基于数据库的 Web 站点，它们的响应速度慢得惊人，好像好多 Web 专家还没有意识到这个问题的存在 (我在整个第 3 章中讲述了这些故事，在那里我列出了权威数据库管理员的一些经验)。

那么，Web 专家应该怎样做呢？首先，我们头脑中必须清楚，要做成一件事有许多条途径。对于数据广播也是这样。本章作为数据广播的序曲，将讨论一些从 Web 上获取数据的相当直接和简单的方法。

1.2 使用无格式文件发布数据

更新 Web 上的数据的一个最简单的方法是使用无格式文件 (flat file, 即 ASCII/TEXT)。如果数据量很小，并且不是很多人将要同时更新系统时，这种方法非常快。

这种方法的一个很好的例子是现在广为采用的来客签名簿 (guest book)，每个人都有。事实上，他们可能拥有同一个签名簿。Matt Wright 在 Web 上可能是最受欢迎的一个站点的所有人和管理者。他的档案中包含了用于各种目的脚本——从搜索引擎 (engine) 的脚本到提供下面所讨论的来客签名簿的脚本 (www.worldwidemart.com)。

来客签名簿说明了使用 CGI 和文本文件所要遵循的准则，这些准则使得终端用户可以更新数据。而且再查看这些数据。我采用了 Matt 的脚本，并做了一些改动，使得它为我的

www.business-america.com 站点提供一个小的商业论坛。请看图 1-1 的屏幕。

图 1-1 Business-America 论坛

请注意来访者的留言 (comment)。实际上, 它使用了两个 HTML 页面和一个 CGI 脚本, 以便使用 CGI 在终端用户和 Web 服务器之间移动数据。

第一个 HTML 页面是“添加”页面, 它是指向上面的 HTML 页面的超文本链。图 1-2 给出了一个典型的来客签名簿的添加输入窗体。

这个窗体的 HTML 源码如下所示:

```
<HTML>
<TITLE> Business America Add to Our Guest
Book </TITLE>
```

```
<body bgcolor = " # ffffff" text = " # 000000"
link = " # 342567">
<center>
<body>
<h1> Add to our Business Guestbook </h1>
</center>
```

```
Fill in the blanks below to add to our guestbook.
The only blanks that you
have to fill in are the comments and name section.
Thanks! <hr>
```

```
<form
method = POST action = "/cgi-bin/guest.pl">
```

Business Forum Comments: `<br>`

`<textarea`

`name = comments COLS = 60 ROWS = 8> </textarea>`

`<hr>`

Your Name:

`<input type = text name = realname size = 30> <br>`

E-Mail:

`<input type = text name = username size = 40> <br>`

URL: `<input type = text name = url size = 50> <br>`

city: `<input type = text name = city size = 15>`, State:

`<input type = text name = state size = 2>` Country: `<input`

`type = text name = country size = 15> <p>`

`<p>`

`<input type = submit> * <input type = reset>`

`</form>`

`<hr>`

`<a href = "guest, html">`

Back to the Guestbook Entries `</a> <br>`

`</body>`

`</html>`

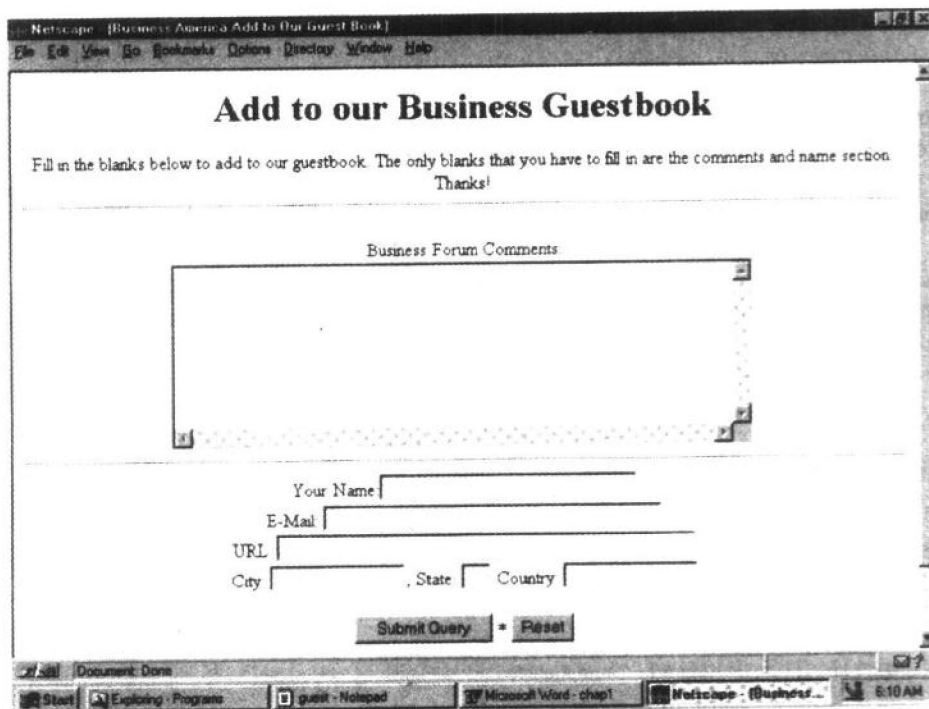


图 1-2 典型的输入窗体

当来访者单击了 Submit (提交) 按钮以后, 控制会转给该站点的 cgi-bin 目录下的一个程序。然后, 一个称为 guest.pl 的 Perl 程序会处理输入信息, 进行一些简单的错误检查确保某些输入域是非空的, 然后将输入信息写入到一个称为 guest.htm 的输出文件, 同时显示 “thank you for adding your comments” 消息。最后这一条消息是从 Perl 程序中作为 HTML 文件动态生成的。

我不打算更详细地剖析此 Perl 脚本的细节 (Matt 在他的站点给出了很详细的指导)。我只想告诉大家, 这是向 Web 发布准动态 (quasi-dynamic) 数据的简单而有效的方法。

guest.pl Perl 脚本的代码如下所示:

```
#!/usr/local/bin/perl
#####
Guestbook                version 2.3.1
# Copyright 1996 Matt Wright    mattw@worldwidemart.com
# Created 4/21/95              Last Modified 10/29/95
# Scripts Archive at:         http://www.worldwidemart.com/scripts/
#####
# COPYRIGHT NOTICE
# Copyright 1996 Matthew M. Wright All Rights Reserved.
#
# Guestbook may be used and modified free of charge by anyone so long as
# this copyright notice and the comments above remain intact. By using this
# code you agree to indemnify Matthew M. Wright from any liability that
# might arise from it's use.
#
# Selling the code for this program without prior written consent is
# expressly forbidden. In other words, please ask first before you try
# and make money off of my program.
#
# Obtain permission before redistributing this software over the Internet
# in any other medium. In all cases copyright and header must remain intact.
#####
# Set Variables

$guestbookurl = "http://www.business-america.com/guest.htm";
$guestbookreal = "/usr/local/wwwdocs/www.business-america.com/guest.htm";
$guestlog = "/usr/local/wwwdocs/www.business-america.com/guestlog.htm";
$cgiurl = "http://www.business-america.com/cgi-bin/guest.pl";
$date_command = "/usr/bin/date";

# Set Your Options:
$mail = 0;                # 1 = Yes; 0 = No
$uselogs = 1;             # 1 = Yes; 0 = No
$linkmail = 1;            # 1 = Yes; 0 = No
$separator = 1;           # 1 = <hr>; 0 = <p>
$redirection = 0;          # 1 = Yes; 0 = No
$entry_order = 1;         # 1 = Newest entries added first;
                          # 0 = Newest Entries added last.

$remote_mail = 0;         # 1 = Yes; 0 = No
$allow_html = 1;          # 1 = Yes; 0 = No
```

```

$line_breaks = 0;          # 1 = Yes; 0 = No

# If you answered 1 to $mail or $remote_mail you will need to fill out
# these variables below:
$mailprog = '/usr/lib/sendmail';
$recipient = 'you@your.com';

# Done
#####

# Get the Date for Entry
$date = "$date_command + "%A,%B %d,%Y at %T (%Z)"; chop ($date);
$shortdate = "$date_command + "%D %T %Z"; chop ($shortdate);

# Get the input
read (STDIN, $buffer, $ENV {'CONTENT_LENGTH'});

# split the name-value pairs
@pairs = split (/&/, $buffer);

foreach $pair (@pairs) {
    ($name, $value) = split (/=/, $pair);

    # Un-Webify plus signs and %-encoding
    $value = ~tr/+//;
    $value = ~s/% ([a-fA-F0-9]) [a-fA-F0-9]) /pack ("C", hex ($1)) /eg;
    $value = ~s/<(! - (.|\n)*) -> //g;

    if ($allow_html != 1) {
        $value = ~s/< ([^]) | \n)*) -> //g;
    }

    $FORM {$name} = $value;
}

# Print the Blank Response Subroutines
&no_comments unless $FORM {'comments'};
&no_name unless $FORM {'realname'};

# Begin the Editing of the Guestbook File
open (FILE, "$guestbookreal") || die "Can't Open $guestbookreal: $! \n";
@LINES = <FILE>;
close (FILE);
$SIZE = @LINES;

# Open Link File to Output
open (GUEST, "> $guestbookreal") || die "Can't Open $guestbookreal: $! \n";

for ($i=0; $i <= $SIZE; $i++) {
    $_ = $LINES [$i];

```

```

if ( / <! - begin - > / ) {

    if ( $entry_order eq '1' ) {
        print GUEST " <! - begin - > \n";
    }

    if ( $line_breaks == 1 ) {
        $FORM {'comments'} = ~ s/ \ CM \ n/ <br> \ n/g;
    }

    print GUEST " <b> $FORM {'comments'} </b> <br> \n";

    if ( $FORM {'url'} ) {
        print GUEST " <a href = \" $FORM {'url'} \"> $FORM {'realname'} </a>";
    }
    else {
        print GUEST " $FORM {'realname'}";
    }

    if ( $FORM {'username'} ) {
        if ( $linkmail eq '1' ) {
            print GUEST " \ &it; <a href = \"mailto: $FORM {'username'} \">";
            print GUEST " $FORM {'username'} </a> \ &gt;";
        }
        else {
            print GUEST " &lt; $FORM {'username'} &gt;";
        }
    }

    print GUEST " <br> \n";

    if { $FORM {'city'} } {
        print GUEST " $FORM {'city'} ,";
    }

    if ( $FORM {'state'} ) {
        print GUEST " $FORM {'state'} ";
    }

    if ( $FORM {'country'} ) {
        print GUEST " $FORM {'country'} ";
    }

    if ( $separator eq '1' ) {
        print GUEST " - $date <hr> \n \n";
    }
    else {
        print GUEST " - $date <p> \n \n";
    }

    if ( $entry_order eq '0' ) {

```