
Pentaho BI 套件的架构与使用权威指南

罗时飞 著

<http://www.open-v.com>

2011 年 8 月 27 日

【版权所有、侵权必究】

目 录

序	VIII
前言	X
1 商业智能概述	1
1.1 BI 发展动向及趋势	1
1.1.1 从察觉已实施 BI 项目的问题启程	1
1.1.2 开源 BI 在导演 BI 行业的未来	2
1.1.3 一些客户对开源 BI 软件的担忧	4
1.2 主流开源 BI 套件	5
1.2.1 Pentaho BI 套件	6
1.3 小结	7
2 迈入 Pentaho BI 3.5 开源套件	8
2.1 下载及安装 Pentaho BI 平台	8
2.1.1 初识 Pentaho BI 服务器	9
2.1.2 启用 Pentaho 管理控制台	10
2.2 配置 Pentaho BI 平台	11
2.2.1 调整宿主 BI 服务器的 JVM 参数	11
2.2.2 调整 BI 服务器的日志输出策略	12
2.2.3 调整宿主 BI 服务器的 Apache Tomcat 参数	12
2.2.4 将 Pentaho BI 服务器的资料库迁移到 Oracle 数据库	13
2.2.5 将 Pentaho BI 服务器的资料库迁移到 MySQL 数据库	17
2.2.6 保护 Pentaho 管理控制台	19
2.3 小结	20
3 数据加工王者—Kettle	21
3.1 ETL 及 Kettle 概述	21

3.1.1	基于“流”架构的 Kettle.....	21
3.1.2	下载及安装 Kettle.....	22
3.2	Spoon—设计转换及作业的集成开发环境	23
3.2.1	启动 Spoon.....	23
3.2.2	从 Kettle 内置的 ETL 转换和作业示例谈起	24
3.2.3	监控 ETL 转换的执行性能	29
3.2.4	调整宿主 Spoon IDE 的 JVM 内存	30
3.3	将转换和作业进行外在化管理	30
3.3.1	存储到数据库中—以 Oracle 为例	30
3.4	Kettle 内置的 ETL 相关辅助工具	32
3.4.1	Pan—执行转换	32
3.4.2	Kitchen—执行作业	32
3.4.3	Carte—添加新的 ETL 执行引擎	33
3.4.4	Encr 加密工具.....	35
3.5	基于集群并发加工大批量数据	35
3.5.1	静态集群模式	36
3.5.2	动态集群模式	36
3.6	与 Pentaho BI 服务器的集成.....	36
3.7	自定义及扩展 Kettle.....	36
3.8	Kettle 最佳实践.....	36
3.8.1	善待 Kettle 内置的变量集合	36
3.9	其他 ETL 解决方案.....	36
3.9.1	同 IBM DataStage 的对比	36
3.9.2	Spring Batch—另一种风格的 ETL 解决方案	37
3.10	小结.....	37
4	Action Sequence—集大成者	39
4.1	Action Sequence 概述.....	39
4.1.1	Pentaho Design Studio 开发工具.....	40
4.2	深入到 Action Sequence 中	42

4.2.1	Action Sequence 定义.....	42
4.2.2	测试 Action Sequence.....	43
4.2.3	组件集合	44
4.3	于复杂 BI 场景中进行 Action Sequence 实战	48
4.3.1	银行 ETL 调度场景概述	48
4.3.2	Action Sequence 的创建过程.....	48
4.3.3	运行并验证 Action Sequence 的执行	49
4.4	小结	49
5	Pentaho 报表工具—数据展现解决方案	50
5.1	Pentaho 数据展现解决方案概述.....	50
5.1.1	Pentaho 元数据编辑器概述.....	50
5.2	Pentaho Report Designer.....	51
5.2.1	PRD 的下载及安装.....	51
5.2.2	借助 PRD 完成报表的制作	52
5.3	借助 PME 梳理报表模型.....	52
5.3.1	PME 的下载及安装.....	52
5.3.2	使用 PME.....	52
5.3.3	PRD 中报表模型的使用.....	52
5.4	Pentaho 即席报表.....	52
5.4.1	揭秘 metadata.xmi.....	53
5.4.2	即席报表的制作	53
5.5	嵌入式 Pentaho 报表引擎.....	53
5.5.1	操作型 BI 报表	53
5.5.2	嵌入式报表的研发过程	53
5.6	Pentaho 数据展现最佳实践.....	53
5.6.1	中文问题	53
5.7	小结	54
6	Mondrian OLAP 引擎—多维数据分析利器.....	55

6.1	OLAP 概述	55
6.1.1	多维建模及数据仓库设计	55
6.1.2	Mondrian OLAP 引擎.....	55
6.2	使用 Mondrian	55
6.2.1	下载 Mondrian OLAP 引擎	55
6.2.2	初探 Mondrian OLAP.....	56
6.2.3	Mondrian OLAP 使用案例研究.....	56
6.3	借助 PSW 设计 OLAP Cube.....	56
6.3.1	下载 Pentaho Schema Workbench.....	56
6.3.2	初探 PSW.....	57
6.3.3	PSW 使用案例研究.....	57
6.4	Mondrian 技术架构探讨	57
6.5	与 Pentaho BI 服务器的集成.....	58
6.6	借助 Pentaho Aggregation Designer 提升数据分析性能	58
6.6.1	数据聚合概述	58
6.6.2	PAD 的下载和安装.....	58
6.6.3	PAD 使用案例研究.....	59
6.7	小结	59
7	基于 Weka 的数据挖掘解决方案.....	60
7.1	数据挖掘概述	60
7.1.1	Weka 介绍.....	60
7.2	采纳 Weka 进行数据挖掘	60
7.2.1	下载 Weka.....	61
7.2.2	Weka 使用案例研究.....	61
7.3	小结	61
8	Pentaho 仪表盘工具.....	63
8.1	Pentaho Dashboard 工具概述	63
8.1.1	Community Dashboard Framework 介绍.....	63

8.1.2	借助 Flash 展现	64
8.2	小结	64
9	Pentaho BI 套件高级特性讨论	65
9.1	配置新的解决方案库	65
9.1.1	Solution 概述	65
9.1.2	实践 Solution	65
9.2	基于元数据的架构思路	65
9.3	基于领域模型的安全性管理	65
9.4	小结	65
10	附录 A: Kettle 组件权威指南	66
10.1	专注转换的组件集合	66
10.1.1	输入组件	66
10.1.2	输出组件	66
10.1.3	转换组件	66
10.1.4	实用 (Utility) 组件	66
10.1.5	流程控制 (Flow) 组件	66
10.1.6	脚本组件	67
10.1.7	查询组件	67
10.1.8	连接组件	67
10.1.9	数据仓库组件	67
10.1.10	校验 (Validation) 组件	67
10.1.11	统计 (Statistics) 组件	67
10.1.12	作业组件	67
10.1.13	映射组件	68
10.1.14	内联组件	68
10.1.15	批量装载 (Bulk Loading) 组件	68
10.2	专注作业的组件集合	68
10.2.1	通用组件	68

10.2.2	邮件组件	68
10.2.3	文件管理组件	68
10.2.4	条件组件	68
10.2.5	脚本组件	69
10.2.6	批量加载组件	69
10.2.7	XML 组件.....	69
10.2.8	文件传输组件	69
10.2.9	资源库组件	69
11	附录 B: Spring Batch	70
11.1	为 ETL 而战	70
11.2	Spring Batch 概述.....	70
11.3	实践 Spring Batch	70
12	附录 C: 相关资料.....	71
12.1	图书.....	71
12.2	网站.....	71

序

Anyplace, Anywhere, Anytime。

虽然它只是一首著名歌曲的歌名，但却能够代表商业智能（Business Intelligence, BI）的未来。

透过 http://en.wikipedia.org/wiki/Business_intelligence，我们能够了解到，它是这样定义 BI 的：

“Business intelligence (BI) refers to skills, technologies, applications and practices used to help a business acquire a better understanding of its commercial context. Business intelligence may also refer to the collected information itself.

BI technologies provide historical, current, and predictive views of business operations. Common functions of business intelligence technologies are reporting, OLAP, analytics, data mining, business performance management, benchmarking, text mining, and predictive analytics.”

从中可以推理出，业务数据是 BI 的基础、灵魂。BI 的一切工作都是围绕业务数据展开的，并从中获得各种有利于商业运作的信息，从而为智能决策提供最强有力的支撑。

借助 BI 产品能够对数据实施全生命周期管理，涉及的环节包括数据加工、数据展现、数据分析等。比如，以 ETL 为主的数据加工，以报表为主的数据展现，以多维分析(OLAP)、数据挖掘为主的数据分析。从形态来看，企业可以部署单独的 BI 产品来管理这些环节，它们也可以以嵌入式方式进行，比如直接在传统业务系统（OLTP）中嵌入实时报表。

现如今，各大商业软件巨头都有自身成熟的 BI 产品栈，比如 IBM、Oracle、SAS。由于开源运动的逐渐发展、成熟，使得开源 BI 开始对这些商业 BI 巨头构成冲击，这其中以 Pentaho BI 和 Jaspersoft 为代表。社区的开放性，敏捷收集各种 BI 需求，灵活的实施模式，较低的实施费用，源码公开，定制化工作能够很容易进行，等等这些都是开源 BI 的优势所在。我们有理由相信，开源 BI 必将得到广泛部署。

过去的几年中，我们所在的银行 BI 团队成功将 Pentaho BI 应用到各种生产场景，这其中以集成和扩展 Kettle、Pentaho Reporting（JFreeReport）、Mondrian、Weka 为主。

开源 BI 的兴起，加上多年 Pentaho BI 经验，使得本书的诞生成为可能。透过本书，读者不仅能够掌握 Pentaho BI 套件的使用，而且对其技术架构有较深入了解。虽然本书是围绕 Pentaho BI 展开的，但我们更希望读者能够将它看成是一本传播 BI（开源 BI）知识和实践

的著作。当开源 BI 在国内被广泛采纳时，你们能够第 1 时间想起此书，则我们的写作目标就达到了。让我们共同为这一目标努力！

如果我们的图书对您有帮助，或者愿意支持图书的持续写作，则可以通过支付宝支持我们，支付宝帐号是：openvcube@gmail.com。让我们一起做得更好！

当然，BI 涉及的知识面非常广，加上我们经验有限，书中难免出现错误，还望同行批评指正，并提出各种宝贵写作建议。

罗时飞

E_mail: openvcube@gmail.com

2011 年于广州

前言

全书将贯穿“数据加工、数据展现、数据分析”这一主线进行 BI 知识及实践的讨论。在这里，Pentaho BI 套件是我们依托的 BI 产品栈。

我们将各章的主体内容安排如下。

- 第 1 章，商业智能概述。针对当前 BI 的发展动态及趋势进行探讨。
- 第 2 章，迈入 Pentaho BI 3.5 开源套件。Pentaho BI 服务器和管理控制台处在 Pentaho BI 套件的舞台中央，它们将这一套件的各组成部分集成成统一的视图，通过这一视图能够高效使用、管理及监控整个 Pentaho BI 套件。
- 第 3 章，数据加工王者—Kettle。我们的实践证明，Kettle 能够胜任各种数据加工场景，比如 ETL、数据质量管理。
- 第 4 章，Action Sequence—集大成者。这是 Pentaho BI 解决方案中最为重要、基础的特性。
- 第 5 章，Pentaho 报表工具—数据展现解决方案。Pentaho 不仅提供了一流的报表工具，而且报表的执行场景可以多样化，比如单独部署、集成到 Pentaho BI 服务器视图中、嵌入到传统 OLTP 业务系统中。
- 第 6 章，Mondrian OLAP 引擎—多维数据分析利器。Mondrian 是业界不错的多维数据分析引擎，许多开源 BI 套件集成了它。
- 第 7 章，基于 Weka 的数据挖掘解决方案。在众多开源 BI 套件中，能够提供数据挖掘能力的不多。Pentaho 的 Weka 值得我们实践、研究。
- 第 8 章，Pentaho 仪表盘工具。
- 第 9 章，Pentaho BI 套件高级特性讨论。
- 第 10 章，附录 A，Kettle 组件权威指南。用户要经常查阅它。
- 第 11 章，附录 B，Spring Batch。在各种数据加工场合，除了 Kettle 外，我们还可以启用 Spring Batch，它是 Spring 社区同 Accenture（埃森哲）合作的结晶。
- 第 12 章，附录 C，相关资料。

值得注意的是，<http://openv-cube.googlecode.com> 提供了全书配套代码、脚本的下载，借助如下 SVN 命令能够将它们下载到 D:\springsource\ebooks 位置。

```
D:\springsource>svn co http://openv-cube.googlecode.com/svn/trunk/ ebooks
```

随后，开发者可以使用它们，或在 STS 中导入各自的代码或脚本，并完成各自运行和调试工作。如果需要不定期更新它们，则可借助如下 SVN 命令。

```
D:\springsource\ebooks>svn update
```

任何问题，可以同我们取得联系，谢谢！我们特提供了 QQ 群（106813165），以探讨同本书相关的技术问题。

1 商业智能概述

本章内容将从分析已实施 BI 项目的问题入手，从而进入到 BI 的发展动向及趋势当中。最终，开源 BI 进入我们的视野，它为 BI 领域的发展指明了道路。

1.1 BI发展动向及趋势

事物都有它的过去、现在及将来。为了更好地掌握现在及将来，人们常说，“以史为鉴”。是的，为了更好地认识 BI 的发展动向及趋势，我们还是先从分析现有已实施 BI 项目所暴露的问题入手。

1.1.1 从察觉已实施BI项目的问题启程

现如今，商业智能领域已经发展了多年，大量的客户采购了各种商业 BI 产品，并实施了许多 BI 项目。然而，期间碰到了不少问题，下面列举了比较突出的常见问题。

已实施 BI 项目暴露的突出问题

- 传统 BI 项目的投入惊人，无论是人力上的，还是费用方面。在短期内，客户很难看到这类项目的业务价值。不仅如此，实施周期长也严重降低了客户对传统 BI 项目的好感
- “徒有虚名”的商业智能。按理说，“灵活应对最终用户的各种未知合理业务需求”，这应该是商业智能项目的最基本诉求，比如调整报表的布局、动态修改 ETL 中待加工的数据库表集合、挖掘行业趋势等。很多时候，即使已实施的 BI 项目支持此类需求，也是在技术人员的参与下才能够完成，业务人员很难“独善其身”
- 不少传统 BI 项目的应用价值仅仅停留在固定报表生成层面
- 没有同 OLTP（On-Line Transaction Processing，联机事务处理）类型的应用融合在一起，比如未将 BI 功能嵌入到实时票务系统中。我们一般将 BI 项目看成 OLAP（On-Line Analysis Processing，联机分析处理）类型的应用，理由主要在 BI 应用处理的数据量大，而且数据一般都不是实时的，比如很多时候都是 T+1、T+2 之类的数据

- 应对新需求的可扩展性差。随着 BI 项目实施进程的推进，新的 BI 需求会逐渐加入进来。然而，一开始并没有考虑到这种需求的不确定性。更何况，很多时候，BI 项目本身的需求就是不确定的，这是 BI 应用同 OLTP 系统的重要差别。这一问题主要体现在商业 BI 产品上，因为上新的“扩展”意味着新的大笔支出

1.1.2 开源BI在导演BI行业的未来

现如今，市场讯息万变，至少 80% 以上的企业客户不能够忍受 BI 项目的实施周期过长，比如有些传统 BI 项目实施周期超过 2 年。这么长的时间中，无论是已采购的商业 BI 产品是否仍然适用，还是现实的市场环境是否发生了翻天覆地的变化，等等都是问题。最终，按照这种周期实施下来的 BI 项目会存在不少变数。

费用过高也是传统 BI 项目实施过程中不可避免的问题，不少商业 BI 产品不仅昂贵，它们对部署环境的要求非常苛刻，它们几乎要独占小型机或其他更好的物理机器。不少企业客户也正是这一原因不敢实施 BI 项目，作者就碰到很多客户向我咨询这类问题。

借助开源 BI 能大大降低 BI 项目的实施风险，比如项目的实施费用减少、周期变短。同商业 BI 产品相比，开源 BI 的某些企业级特性会稍差一些，但我们可以通过其他一些办法进行变通。值得注意的是，不少开源 BI 产品的研发是在商业公司的主导下研发而成的，这其中 Pentaho、Jaspersoft 便是，它们主导的许多开源 BI 项目都由资深 BI 专家领衔，比如 Matt Casters 架构的 Kettle ETL 工具、Julian Hyde 负责的 Mondrian OLAP 引擎。而且，类似 Kettle 和 Mondrian 的开源 BI 软件不比商业 BI 产品差，甚至在某些方面还超越了它们，比如部署和维护方便、数据加工和分析性能优异等。读者不信可以去动手实践一下，并比对各自的结果。

有人可能会问，开源 BI 能够解决上述谈到的传统 BI 项目实施期间碰到的各种棘手问题吗？下面列举了开源 BI 的若干优势，透过它们来阐述开源 BI 有助问题的解决。

开源 BI 的优势

- 开源 BI 软件几乎不用付费，即使需要，费用也是很低。而且，它们对物理机器的要求并不苛刻，普通或高档 PC Server 便可胜任。与此同时，客户也能够不断地免费升级到新版开源 BI 软件。这意味着，借助开源 BI

软件，客户能够迅速启动 BI 项目的实施工作，比如硬件采购环节将会很顺利。因此，借助开源 BI 实施 BI 项目的失败代价较低

- 通常，为吸引客户试用和使用它们，开源 BI 软件会以模块化、集成式方式提供，即客户可以根据自身的 BI 项目实施需要灵活组装具有不同 BI 能力的 BI 产品集合。自始至终，客户都是在同一界面中进行操作的，加上开源 BI 软件操作使用简单，进而提升了用户体验和使用效率。不幸的是，很多商业 BI 软件不允许客户试用它们，比如通过它们的官方网站根本找不到下载入口。与此同时，考虑到商业利益的需要，商业 BI 厂商通常不会以模块化、集成式方式提供它们的 BI 软件，而且操作过程较为复杂，没有厂商的支持一般是较难上手的
- 借助开源 BI，客户可以更容易分阶段敏捷实施 BI 项目，实施周期可以灵活控制，这其中“开源 BI 软件对实施环境的要求很低”这一因素起到了很重要的作用，比如客户可以在性能较差的笔记本上实践各种复杂 BI 场景。由于 BI 项目要应对太多的业务不确定性（这一点前面谈到过），因此通过分阶段实施 BI 项目以降低项目实施风险显得非常重要。比如，上一阶段的经验教训能够直接为下一阶段服务，而后续阶段能够不断改进此前的不足，这其中包括“如何有效应对业务需求的不确定性”这一棘手问题。商业 BI 软件则存在很多麻烦，很多时候，往往要求厂商的技术或业务人员在场，因为商业 BI 软件的使用门槛较高、复杂，加上它们一般都是“身材魁梧”，安装下来占据若干 GB 的物理空间很正常
- 开源 BI 软件能够很容易融入到传统 OLTP 应用中。我们知道，开源软件在遵循开放标准、扩展能力等方面往往做得非常不错。在引领操作型 BI 方面，开源 BI 能够做得更好，比如将开源 BI 软件嵌入到 OLTP 应用中
- 如有需要，客户可扩展开源 BI 软件本身，这是商业 BI 产品所不允许的行为。对于开源 BI 软件而言，源码通常都是公开的。这些都说明，客户在借助开源 BI 软件实施 BI 项目时，他们全局掌控能力可以得到淋漓尽致的发挥
- 安装、部署非常方便。比如，Pentaho BI 套件的安装过程非常简单，直接将它解压到某一位置即可；Kettle ETL 工具也是如此。商业 BI 产品的安

装部署过程往往非常复杂，比如 IBM DataStage

- 维护成本极低。开源 BI 软件往往以实用著称，不会有太多的花架子，毕竟它们不用借助这类特性去迎合潜在的客户。最终，这使得基于开源 BI 实施的 BI 项目容易得到维护和升级

在商业 BI 产品和开源 BI 软件逐渐同质化的今天，开源 BI 在实施 BI 项目的费用、周期（包括分阶段、分步骤迭代）、质量保证等方面更具优势。我们相信，随着越来越多的大型项目采纳开源 BI 软件，商业 BI 产品的市场份额将遭到重创，这也是大势所趋。

经过这么多年的发展，BI 项目不应该再留给客户“昂贵、实施成功率低、项目周期长”等印象，它们应该是“亲民”的，更多的行业、企业，尤其是中小企业需要它们。当 BI 项目到处实施时，其价值才能够得到真正体现，而这些必将带领 BI 行业实现跨越式发展。

上述针对开源 BI 列举出的优势正说明，未来 BI 的发展离不开开源 BI。我们的市场、客户需要高质量、快速实施 BI 项目，从而真正为他们的业务决策提供帮助。在业务层面，也不希望 BI 项目独立于 OLTP 应用，客户希望它们走向融合，将 BI 功能嵌入到 OLTP 应用中，从而为现实的业务提供实时、精确的 BI 能力。

基于开源 BI 的新型 BI 项目的实施工作将到处可见，开源 BI 软件为新型 BI 项目插上了飞翔的翅膀。最终，客户使用 BI 软件的深入程度将得到逐渐改善，他们能够尽可能专注于业务价值的挖掘，进而最大化 BI 项目的价值。

1.1.3 一些客户对开源BI软件的担忧

“缺少商业支持，文档匮乏、Bug 太多”，这些是开源软件在不少客户心中的印象。这里，我们需要一一澄清这些问题。

先来谈谈“缺少商业支持”这一问题。通常，开源软件已经形成了一个生态圈，有些开源软件是个人行为发起的，而有些是商业公司在运作它们，并提供商业支持。甚至，一些商业公司会同时提供开源及商业版本。这就是说，不同开源软件的质量会存在一定的差别。在决定使用某开源 BI 软件前，客户可以去进行足够的调研工作，这其中包括软件的试用。比如，选择的开源软件功能如何、是否存在商业支持、市场占有率如何、费用如何等。

人们常说，“日久见人心”，软件的使用又何尝不是。当客户借助开源 BI 软件实施 BI 项目后，如果觉得非常不错，这时可以考虑选购费用并不高、基于开源 BI 软件的商业版本。如果客户觉得选用的开源 BI 套件不怎么样，则即使不再使用它们，代价较低，毕竟花费的

人力和软硬件费用成本很少。但客户如果借助传统商业 BI 软件实施的 BI 项目失败了，则代价非常高。软件是否好用，这是需要客户在真实的生产环境中使用的，而且使用群体要广、时间要足够长，传统商业 BI 软件是几乎不允许客户在未购买它前去用于生产的，但开源 BI 软件可以做到这一点。比如 Pentaho 公司同时提供 Pentaho BI 社区套件和 Pentaho BI 企业套件，Pentaho BI 企业套件构建在 Pentaho BI 社区套件之上，功能更加丰富，但它们的基代码是一致的。Pentaho BI 社区套件可供客户免费使用，而 Pentaho BI 企业套件需要收取一定的费用，这些费用包括产品和服务本身。即使只使用 Pentaho BI 社区套件，客户也可以选择购买 Pentaho 的服务支持，包括第三方提供的 Pentaho 服务支持，比如 Pentaho 合作伙伴。

好了，来探讨“文档匮乏”这一问题。成功的开源项目往往存在丰富的文档，这其中包括操作手册、技术文档、Wiki（知识库）、论坛、博客。同传统商业 BI 软件相比，开源 BI 软件的文档可能会稍差一些，但获得解答问题的渠道非常广，客户能够通过它们迅速找到问题的答案，毕竟成功的开源项目存在广泛的参与者和使用者。

再来研究“Bug 太多”这一问题。还是那句话，对于成功的开源项目而言，这一问题出现的几率很低，否则没有什么客户使用它。再说了，没有 Bug 的软件是不存在的。微软的 Windows 也不是没有 Bug。当一流的商业团队在运作成功的开源项目时，他们能够积极地、第 1 时间修改广大客户提交的 Bug，比如通过邮件、JIRA。客户可以随时拿到最新版本的开源 BI 软件。

最后要强调的是，各行各业已经在深入使用各种开源软件了，比如 Linux、Apache Http Server、Tomcat、MySQL。对于 BI 行业而言，也是积极启用开源 BI 软件的时候了。开源 BI 软件能够改变传统 BI 项目的“昂贵、实施成功率低、项目周期长”诟病，让我们一起拥抱新型 BI 项目吧！

1.2 主流开源BI套件

从目前的市场格局及产品构成看，开源 BI 领域中主要存在 Pentaho BI 套件和 Jaspersoft BI 套件，它们能够提供完整的、成熟 BI 套件，而且是基于 Java EE 平台技术研发而成。这两个套件各有各的特色，而且它们底层的一些技术支撑存在交集，比如基于 Java EE 平台技术、采纳了同样的 Mondrian OLAP 引擎等。同 Jaspersoft BI 套件相比，Pentaho BI 套件借助 Weka 项目进入到数据挖掘领域，从而进一步延伸了 Pentaho 的 BI 能力。

本书主要以 Pentaho BI 套件为例展开论述，作者也一直在依托它实施各种类型的 BI 项目。

1.2.1 Pentaho BI套件

图 1-1 给出了 Pentaho BI 套件的组成情况。

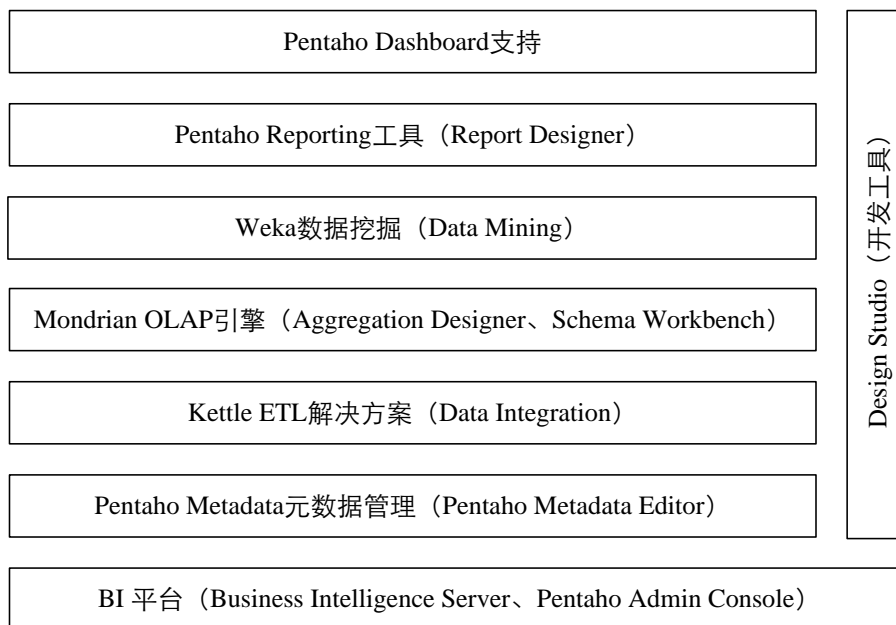


图 1-1 Pentaho BI 套件组成情况

其中，各产品的相关内容如下：

- **BI 平台：**Business Intelligence Server，商业智能服务器。它是整个 Pentaho BI 产品策略的重要基础件，也是 Pentaho 各类产品的重要门户，借助它集成 BI 产品线的其他产品集合。内置在 BI 平台中的 Pentaho Admin Console（管理控制台）是管理整个平台的重要后端软件。
- **Design Studio：**开发工具，目前支持 Action Sequence 的图形化开发工作。
- **Pentaho Metadata 元数据管理：**基于 CWM 规范实施元数据管理，借助内置的 Pentaho Metadata Editor 能够快速实施元数据管理，并将 metadata.xml 部署到 BI 平台中。
- **Kettle ETL 解决方案：**Data Integration，用于各种场景的 ETL 工作，包括数据质量管理（Data Quality Management，DQM）。
- **Mondrian OLAP 引擎：**针对多维分析而提供的既灵活又高性能的 OLAP 引擎，它能够部署到各种环境，而且支持的数据类型多种多样，比如关系数据库、Teradata

等。为简化 OLAP Cube 的定义和维护，Pentaho 提供了 Schema Workbench 工具；为充分改善性能及降低聚合技术的采纳门槛，Pentaho 提供了 Aggregation Designer。

- Weka 数据挖掘 (Data Mining)：内置了各种数据挖掘算法支持。
- Pentaho Reporting 工具：Report Designer，支持各种类型报表的设计、开发工作，并直接将它们部署到 Business Intelligence Server 中。另外，报表的输出结果多种多样，比如 PDF、Excel、HTML、RTF、文本文件等。
- Pentaho Dashboard 支持：仪表盘开发。

从 Pentaho BI 套件的构成情况看，其内置的 BI 能力很广，实力强劲。同业务数据打交道的任何能力都具备了。本书将围绕它们依次展开论述。

1.3 小结

很多 BI 项目的实施结果同预期相差甚远，本章讨论了这其中的不少深层次原因。未来 BI 的发展动向及趋势如何，是否能够着手改变传统 BI 项目实施的尴尬程度，这些都是本章已讨论的内容。

开源 BI 为我们挑明了道路，Pentaho BI 套件是这一领域非常有话语权的产品。从下章开始，将正式进入到 Pentaho BI 套件的探索之中，并借助它实施各种新型 BI 项目。

2 迈入Pentaho BI 3.5 开源套件

本章将开始进入到 Pentaho BI 社区套件（即开源套件）的研究及实践中，商业智能服务器和 Pentaho 管理控制台构成了 Pentaho BI 平台，这里将围绕这一平台的架构和使用展开论述。

2.1 下载及安装Pentaho BI平台

通过 <http://sourceforge.net/projects/pentaho/files/> 网址，我们能够下载到 Pentaho BI 套件的大部分组成部分，比如 Pentaho BI 平台、报表设计器、Kettle ETL 工具等。而 BI 套件的部分内容需要通过其他网址下载，比如在 <http://sourceforge.net/projects/mondrian/files/> 位置可以下载到 Mondrian OLAP 引擎。

图 2-1 展示了 Pentaho BI 服务器及管理控制台的下载入口。








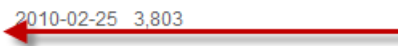







▼ Business Intelligence Server	7.3 GB	2010-02-25	478,020		
▼ 3.5.2-stable	434.3 MB	2010-02-25	6,125		
 biserver-manual-ce-3.5.2.stable.zip	115.7 MB	2010-02-25	653		
 biserver-ce-3.5.2.stable.zip BI Server 3.5.2	172.7 MB	2010-02-25	3,803		
 biserver-ce-3.5.2.stable.tar.gz BI Server 3.5.2	140.4 MB	2010-02-25	1,317	   	
 biserver-ce-3.5.2.stable-javadoc.zip	3.7 MB	2010-02-25	164		
 bi-platform-3.5.2.stable-sources.zip	1.7 MB	2010-02-25	188		

图 2-1 Pentaho BI 服务器及管理控制台的下载

下面给出了图 2-1 中各下载件的详细情况。

各相关下载件的说明

- biserver-manual-ce-3.5.2.stable.zip: 自定义安装 Pentaho BI 平台，资深用户可能会使用到这一工件
- biserver-ce-3.5.2.stable.zip: 内置了 Pentaho BI 服务器及管理控制台的 Windows 版本，当然解压后可以同样用在其他 OS 中
- biserver-ce-3.5.2.stable.tar.gz: 内置了 Pentaho BI 服务器及管理控制台的非

Windows 版本，比如 Linux，当然解压后可以同样用在其他 OS 中

- biserver-ce-3.5.2.stable-javadoc.zip: Pentaho BI 平台对应的 API 规范
- bi-platform-3.5.2.stable-sources.zip: Pentaho BI 平台对应的 Java 源码

这里以 biserver-ce-3.5.2.stable.zip 的使用为例。

2.1.1 初识Pentaho BI服务器

在将 biserver-ce-3.5.2.stable.zip 解压到 D:\后，biserver-ce 和 administration-console 目录将出现，前者就是 BI 服务器，而后者是 Pentaho 管理控制台。

默认时，Pentaho BI 平台会使用内置的 JRE，它位于 D:\biserver-ce\jre 位置。如果用户机器上安装了 JDK，并设置了 JAVA_HOME，则 Pentaho BI 平台会使用用户指定的 JDK。运行如下“start-pentaho.bat”批处理脚本能够启动 Pentaho BI 服务器，它运行在 Apache Tomcat 容器中，并采纳了 HSQLDB 数据库（<http://hsqldb.org/>）。

```
D:\biserver-ce>start-pentaho.bat
```

现在，打开浏览器，并访问 <http://localhost:8080/pentaho>，则将看到登录界面，见图 2-2。

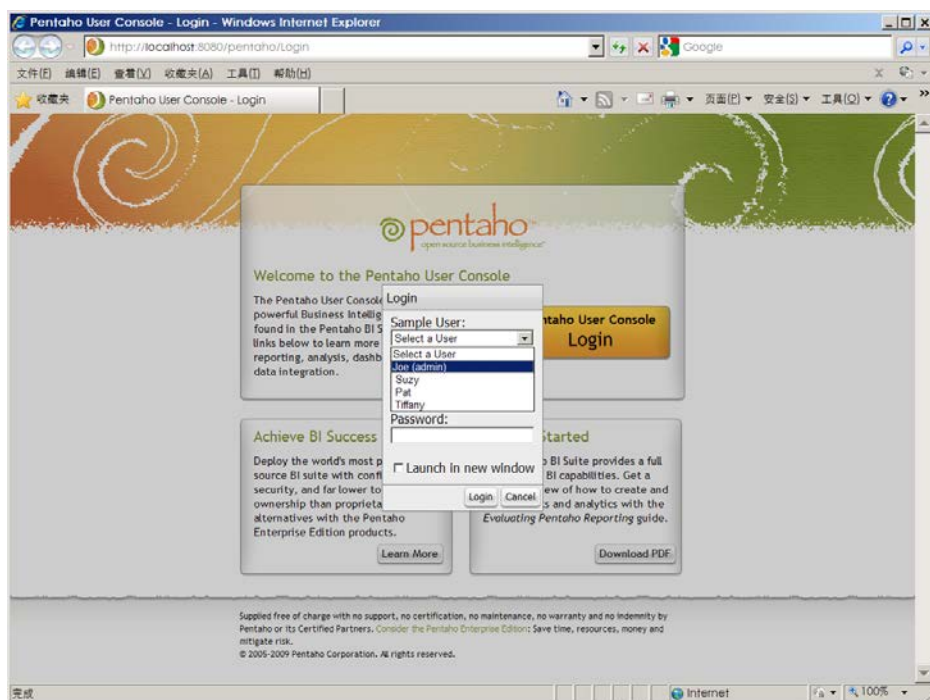


图 2-2 Pentaho BI 服务器登录界面

其登录界面将登录用户集合列举出来了，这使得客户能够更快上手 Pentaho BI 套件。如果需要屏蔽这一列表，则客户可以打开位于 D:\biserver-ce\pentaho-solutions\system 目录中的

pentaho.xml 配置文件，将<login-show-users-list/>的取值置为 false 即可，修改后的配置示例如下，在重启 BI 服务器后，这一新的配置才会生效。

```
<login-show-users-list>false</login-show-users-list>
```

当 joe/password 用户登录后，BI 服务器的主界面将呈现在眼前，见图 2-3。

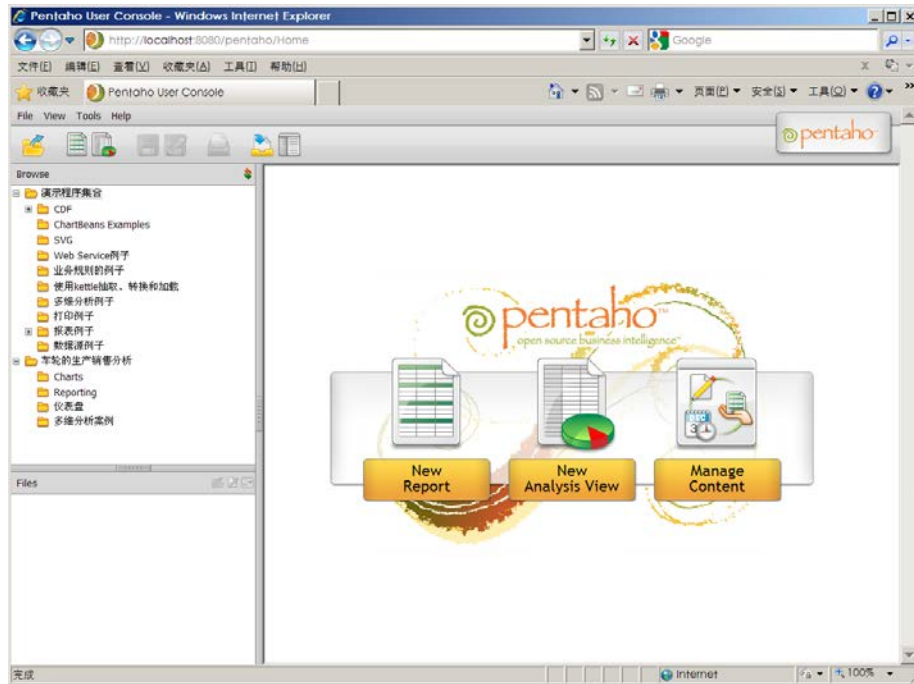


图 2-3 Pentaho BI 服务器主界面

Pentaho BI 套件以 Pentaho BI 服务器为中心，从这一主界面已经能够看出。数据加工、数据展现、数据分析等行为都将通过这一统一的、集成式界面完成，比如客户可以通过它运行并查看报表、运行 ETL 作业、创建各种即席（Ad-Hoc）报表和多维分析。本书后续内容会在相关章节详细介绍到 Pentaho BI 服务器的各个方面，您现在可以根据界面提示进行各种操作，以更进一步认识它。

如果需要停止 Pentaho BI 服务器，则于 D:\biserver-ce 目录下运行“stop-pentaho.bat”批处理脚本即可，示例操作如下。它将同时停止 Pentaho BI 服务器和 HSQLDB 数据库。

```
D:\biserver-ce>stop-pentaho.bat
```

2.1.2 启用Pentaho管理控制台

于 D:\administration-console 目录运行如下“start-pac.bat”批处理脚本能够启动 Pentaho 管理控制台。默认时，它宿主在 Jetty Web 容器中。

```
D:\administration-console>start-pac.bat
```

将浏览器定位到 <http://localhost:8099/> 网址后, 并输入默认的 admin/password 用户, 即可登录到 Pentaho 管理控制台中, 图 2-4 展示了这一控制台。

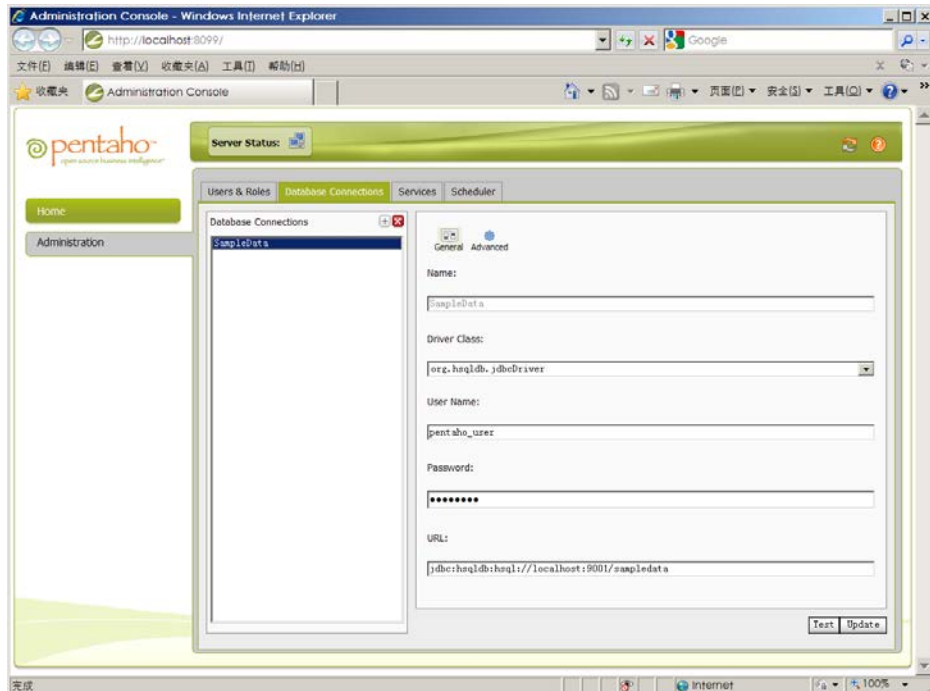


图 2-4 Pentaho 管理控制台的主界面

Pentaho 管理控制台是整个 BI 平台的重要后端软件, 系统管理员通过它能够完成各类操作, 比如维护用户及角色信息、注册新的业务库 (数据库连接)、控制 BI 服务器中的各种敏感信息、使用调度服务等。

如果要停止 Pentaho 管理控制台, 则于 D:\administration-console 目录下运行“stop-pac.bat”批处理脚本即可, 示例操作如下。

```
D:\administration-console>stop-pac.bat
```

2.2 配置 Pentaho BI 平台

下面就 Pentaho BI 平台的常见配置问题分别进行阐述, 这些都是企业生产环境要面对的重要问题。

2.2.1 调整宿主 BI 服务器的 JVM 参数

在生产环境中, 对于 32 位的 JVM 而言, 为了使得 BI 服务器的性能发挥到极致, 一般

都要调整宿主它的 JVM 参数。比如，我们通常需要将 JVM Heap 大小调整到 1G 左右，即“-Xmx1024m”。下面摘录了 start-pentaho.bat 中的示例配置。在调整这些参数后，用户可以借助 JDK 内置的 jconsole 实用工具进行确认。

```
set CATALINA_OPTS=-Xms256m -Xmx768m -XX:MaxPermSize=256m
```

根据 1/2 原则，我们通常要保证-Xmx 参数取值不要超过物理内存的 1/2。比如，物理机器内存为 2G，则-Xmx 取值最好不要超过 1G (2G*1/2)，而-Xms 取值最好不要低于-Xmx 的 1/2，即 512m 左右。不过，不同生产环境的差异性太大，建议用户能够在生产前进行严格的压力及调优测试，并灵活调整 JVM 参数。

值得注意的是，在采纳基于 Intel 架构的物理机器时，而操作系统无论是 Windows 或 Linux，建议能使用 Oracle JRockit 虚拟机(<http://www.oracle.com/technology/products/jrockit>)。同 Sun 或其他厂商的 JVM 相比，同等硬件配置下，基于 JRockit 的 BI 服务器（包括其他类型的 Java EE 应用）的性能能够提升 1 倍左右，这是作者在不少大型项目总结出的数据。而且采用 JRockit 后，客户还能够得到很多附加值，比如它内置的 Mission Control 是用于监控和诊断企业应用性能非常实用的工具。

本节内容主要摘自作者写作的《端到端提升企业级 Java 能力》电子书。

2.2.2 调整BI服务器的日志输出策略

Pentaho BI 服务器默认采用了 Apache Log4j 记录各种日志，开发者具体可参考位于 D:\biserver-ce\tomcat\webapps\pentaho\WEB-INF\classes 位置的 log4j.xml 配置文件。

其配置了“PENTAHOFILE”和“PENTAHOCONSOLE”两个 Appender，并定义了如下内容。开发者可以调整其日志详细程度、日志文件的位置、<category/>定义等。

```
<root>
  <priority value="WARN" />
  <appender-ref ref="PENTAHOCONSOLE" />
  <appender-ref ref="PENTAHOFILE" />
</root>
```

其它细节，开发者可浏览其文件内容，并试图根据自己的意图调整它。

2.2.3 调整宿主BI服务器的Apache Tomcat参数

为提高 BI 服务器的并行吞吐能力，除了调整 JVM 参数外，我们还需要调整宿主它的

Apache Tomcat 参数。下面摘录了 D:\biserver-ce\tomcat\conf 目录中的 server.xml 配置示例。

```
<Connector port="8080" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" redirectPort="8443" acceptCount="100"
    connectionTimeout="20000" disableUploadTimeout="true" />
```

通常，我们需要调整 maxThreads、minSpareThreads、maxSpareThreads、acceptCount 等参数取值。大部分情况下，可以考虑将它们的取值设置成默认的 2 倍左右，即 maxThreads 设置成 300、minSpareThreads 设置成 50、maxSpareThreads 设置成 150、acceptCount 设置成 200。

注意，不同生产环境的差异性较大，建议用户能够在上线前进行严格的压力及调优测试，并灵活调整 Apache Tomcat 参数，毕竟参数不是越大越好，因为这些参数取值要同硬件能力相匹配，否则得不偿失。

如果需要调整 Apache Tomcat 其他参数，用户直接修改同一 server.xml 即可，比如 HTTP 端口。

2.2.4 将Pentaho BI服务器的资料库迁移到Oracle数据库

Pentaho BI 服务器的很多重要信息存储在数据库中，其默认使用 HSQLDB 数据库，即借助它存储自身的资料库，比如 Quartz 调度信息、业务资料库连接信息（数据源）等。下面给出的列表展示了其内置的资料库表集合（未列出 Quartz 对应的表），它们用于支撑 BI 服务器的运行。其中的某些表是用于认证授权操作的，如果 BI 服务器切换成其他安全控制方式，则相关的表不用提供，比如 users、authorities、granted_ authorities 等。本书将在相关的章节仔细阐述它们。

- authorities
- bdparams
- bgcontentid
- contentitem
- contentlocation
- contitemfile
- cplxparams
- datasource

- dtparams
- granted_authorities
- lngparams
- lsparams
- paramtypesmap
- pro_acls_list
- pro_files
- pro_schedule
- pro_subcont_schedlist
- pro_subcontent
- pro_subcontparms
- pro_subs_schedlist
- pro_subscribe
- pro_subscrparms
- rtelement
- ssparams
- user_settings
- users

借助 HSQLDB 内置的管理工具可查看到这些表，比如我们可以在 D:\biserver-ce\data 目录中创建含有如下内容的.bat 批处理文件，并运行它。BI 服务器启动后，HSQLDB 服务器（start_hypersonic.bat）也会被启动。D:\biserver-ce\tomcat\webapps\pentaho\META-INF 目录中的 context.xml 文件提供了 HSQLDB 数据库的相关连接信息。

```
"%JAVA_HOME%/bin/java" -cp lib/hsqldb-1.8.0.jar org.hsqldb.util.DatabaseManager
```

HSQLDB 是不能够支撑真实的企业应用的，生产环境必须替换它。就目前而言，Pentaho BI 服务器的资料库支持 HSQLDB、MySQL 5.x、Oracle 10g、Postgres 8.1.x 等 4 种数据库类型。值得注意的是，BI 服务器内部会借助 Hibernate 操作上述资料库表集合，因此它支持用户扩展使用其他数据库类型，比如 DB2、SQL Server。

这里以切换 HSQLDB 到 Oracle 10g 为例阐述 Pentaho BI 服务器资料库的迁移工作，下节将阐述 MySQL 5.x 迁移工作。

其一，将 Oracle JDBC 驱动拷贝到 D:\biserver-ce\tomcat\webapps\pentaho\WEB-INF\lib 或 D:\biserver-ce\tomcat\common\lib 目录，供 Pentaho BI 服务器访问 Oracle 10g 数据库使用。另外，也需要将 Oracle JDBC 驱动拷贝到 D:\administration-console\jdbc 目录，否则用户不能够正常使用 Pentaho 管理控制台。默认时，Pentaho BI 平台并没有内置 Oracle JDBC 驱动，但内置了针对 HSQLDB、MySQL 5.x、Postgres 8.1.x 的 JDBC 驱动。

其二，初始化 Oracle 10g 数据库，即借助类似 PL/SQL Developer 工具依次执行如下 SQL 脚本集合。期间，它们将创建两个不同 Oracle 用户，一用户（quartz/password）用于存储 Quartz 相关信息，另一用户（hibuser/password）用于存储上述资料库表集合。前者用于存储 Quartz 表，而后者用于存储 Pentaho BI 服务器本身引入的资料库表。

- create_repository_ora.sql: 将创建 pentaho_tablespace 表空间，并新增 hibuser/password 用户，以及 datasource 表。
- create_sample_datasource_ora.sql: 往 datasource 表增加外部业务资料库连接信息。
- create_quartz_ora.sql: 创建 pentaho_user/password 用户、quartz 数据库、Quartz 表等信息。

其三，修改 context.xml 中配置的数据库连接信息，下面给出了配置示例。

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/pentaho" docbase="webapps/pentaho/">
  <Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory"
    maxActive="20" maxIdle="5"
    maxWait="10000" username="hibuser" password="password"
    driverClassName="oracle.jdbc.driver.OracleDriver"
    url="jdbc:oracle:thin:@localhost:1521:ORCL"
    validationQuery="select * from dual"/>

  <Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory"
    maxActive="20" maxIdle="5"
    maxWait="10000" username="quartz" password="password"
    driverClassName="oracle.jdbc.driver.OracleDriver"
    url="jdbc:oracle:thin:@localhost:1521:ORCL"
    validationQuery="select * from dual"/>
</Context>
```

其四，打开 D:\biserver-ce\pentaho-solutions\system\hibernate 中的 hibernate-settings.xml 配置文件，并启用 oracle10g.hibernate.cfg.xml 配置文件，配置示例如下。

```
<config-file>system/hibernate/oracle10g.hibernate.cfg.xml</config-file>
```

其五，D:\biserver-ce\pentaho-solutions\system\hibernate 中的 oracle10g.hibernate.cfg.xml 配置文件也需要调整一下，比如 connection.url (jdbc:oracle:thin:@localhost:1521:ORCL)、connection.username (hibuser)、connection.password (password) 等。

其六，需要修改 applicationContext-spring-security-hibernate.properties 配置文件，它位于 D:\biserver-ce\pentaho-solutions\system 目录，下面给出了配置示例。

```
jdbc.driver=oracle.jdbc.driver.OracleDriver
jdbc.url=jdbc:oracle:thin:@localhost:1521:ORCL
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
```

其七，修改位于 D:\biserver-ce\pentaho-solutions\system\quartz 目录的 quartz.properties 属性文件。当 Quartz 采用 Oracle 存储各种调度信息时，开发者需要启用如下实现类，即将默认的 org.quartz.impl.jdbcjobstore.StdJDBCDelegate 被替换成 OracleDelegate。

```
org.quartz.jobStore.driverDelegateClass=
    org.quartz.impl.jdbcjobstore.oracle.OracleDelegate
```

其八，可选地，用户需要修改 start_hypersonic.bat 中的相关信息，相关默认内容摘录如下。默认时，Pentaho BI 开源服务器内置的 HSQLDB 数据库中持有 3 种不同信息。一方面，hibernate 数据库持有同图 2-5 对应的资料库表集合；另一方面，quartz 数据库持有 Quartz 表，Pentaho BI 服务器借助 Quartz 实现任务调度，比如调度 Kettle ETL 作业、定期运行报表等工作；第三方面，sampledata 数据库持有外部业务资料库信息，供完成报表、ETL 数据加工、多维分析等 BI 工作使用。借助 Pentaho 管理控制台能够获悉 sampledata 数据库的相关连接信息。由于这次的数据迁移工作并没有将 sampledata 数据库迁移到 Oracle 10g，所以除了需要保留它之外，用户可以考虑不激活 hibernate 和 quartz 数据库，即将下面列出的后两行内容删除掉。

```
"%_PENTAHO_JAVA%" -cp %tempclasspath% org.hsqldb.Server
    -database.0 hsqldb\sampledata -dbname.0 sampledata
    -database.1 hsqldb\hibernate -dbname.1 hibernate
    -database.2 hsqldb\quartz -dbname.2 quartz
```

现在，我们便可正常运行 Pentaho BI 服务器和 Pentaho 管理控制台。此时，Pentaho BI 服务器同时宿主在 HSQLDB 和 Oracle 10g 数据库中，前者用于存储外部业务资料库信息，后者用于存储运行 Pentaho BI 服务器所需的资料库信息。

2.2.5 将Pentaho BI服务器的资料库迁移到MySQL数据库

这里以切换 HSQLDB 到 MySQL 5.x 为例阐述 Pentaho BI 服务器资料库的迁移工作。由于 Pentaho BI 服务器内置了 MySQL JDBC 驱动，因此开发者不用再次提供它，除非要使用更新版的驱动替换掉其内置的。

其一，初始化 MySQL 5.x 数据库，具体操作指令如下。期间，它们将创建两个不同 MySQL 用户，一用户(pentaho_user/password)用于存储 Quartz 相关信息，另一用户(hibuser/password)用于存储上述资料库表集合。前者用于存储 Quartz 表，而后者用于存储 Pentaho BI 服务器本身引入的资料库表。其中，create_quartz_mysql.sql 将创建 pentaho_user 用户、quartz 数据库、Quartz 表等信息；create_repository_mysql.sql 将创建 hibuser 用户、hibernate 数据库、datasource 表等信息；create_sample_datasource_mysql.sql 将往 datasource 表增加外部业务资料库连接信息。

```
C:\mysql-5.1.45-win32\bin>mysqld --install
Service successfully installed.

C:\mysql-5.1.45-win32\bin>net start MySQL
MySQL 服务正在启动 .
MySQL 服务已经启动成功。

C:\mysql-5.1.45-win32\bin>mysql -u root -p < D:\biserver-ce\data\mysql5\create_q
uartz_mysql.sql
Enter password:

C:\mysql-5.1.45-win32\bin>mysql -u root -p < D:\biserver-ce\data\mysql5\create_r
epository_mysql.sql
Enter password:

C:\mysql-5.1.45-win32\bin>mysql -u root -p < D:\biserver-ce\data\mysql5\create_s
ample_datasource_mysql.sql
Enter password:

C:\mysql-5.1.45-win32\bin>
```

其二，修改 context.xml 中配置的数据库连接信息，下面给出了配置示例。

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/pentaho" docbase="webapps/pentaho/">
  <Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory"
    maxActive="20" maxIdle="5"
    maxWait="10000" username="hibuser" password="password"
    driverClassName="com.mysql.jdbc.Driver"
```

```

url="jdbc:mysql://localhost:3306/hibernate"
validationQuery="select 1" />

<Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory"
  maxActive="20" maxIdle="5"
  maxWait="10000" username="pentaho_user" password="password"
  driverClassName="com.mysql.jdbc.Driver"
  url="jdbc:mysql://localhost:3306/quartz"
  validationQuery="select 1"/>
</Context>

```

其三，打开 D:\biserver-ce\pentaho-solutions\system\hibernate 中的 hibernate-settings.xml 配置文件，并启用 mysql5.hibernate.cfg.xml 配置文件，配置示例如下。

```
<config-file>system/hibernate/mysql5.hibernate.cfg.xml</config-file>
```

其四，D:\biserver-ce\pentaho-solutions\system\hibernate 中的 mysql5.hibernate.cfg.xml 配置文件也需要调整一下，比如 connection.url（jdbc:mysql://localhost:3306/hibernate）、connection.username（hibuser）、connection.password（password）等。

其五，需要修改 applicationContext-spring-security-hibernate.properties 配置文件，它位于 D:\biserver-ce\pentaho-solutions\system 目录，下面给出了配置示例。

```

jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/hibernate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect

```

其六，可选地，用户需要修改 start_hypersonic.bat 中的相关信息，相关默认内容摘录如下。默认时，Pentaho BI 开源服务器内置的 HSQLDB 数据库中持有 3 种不同信息。一方面，hibernate 数据库持有同图 2-5 对应的资料库表集合；另一方面，quartz 数据库持有 Quartz 表，Pentaho BI 服务器借助 Quartz 实现任务调度，比如调度 Kettle ETL 作业、定期运行报表等工作；第三方面，sampledata 数据库持有外部业务资料库信息，供完成报表、ETL 数据加工、多维分析等 BI 工作使用。借助 Pentaho 管理控制台能够获悉 sampledata 数据库的相关连接信息。由于这次的数据迁移工作并没有将 sampledata 数据库迁移到 MySQL 5.x，所以除了需要保留它之外，用户可以考虑不激活 hibernate 和 quartz 数据库，即将下面列出的后两行内容删除掉。

```

"%_PENTAHO_JAVA%" -cp %tempclasspath% org.hsqldb.Server
  -database.0 hsqldb\sampledata -dbname.0 sampledata
  -database.1 hsqldb\hibernate -dbname.1 hibernate

```

```
-database.2 hsqldb\quartz -dbname.2 quartz
```

现在，我们便可正常运行 Pentaho BI 服务器和 Pentaho 管理控制台。此时，Pentaho BI 服务器同时宿主在 HSQLDB 和 MySQL 5.x 数据库中，前者用于存储外部业务资料库信息，后者用于存储运行 Pentaho BI 服务器所需的资料库信息。

2.2.6 保护Pentaho管理控制台

Pentaho 管理控制台直接管理着 BI 服务器的各种重要信息，因此需要保护好它。默认时，admin/password 用户能够登录到 Pentaho 管理控制台中，如果这一帐号被非法者窃取到，则后果不堪设想。因此，我们有必要修改这一默认系统管理员帐号。

位于 D:\administration-console\resource\config 目录中的 login.properties 属性文件存储了 admin/password 帐号信息，以及 admin 所属角色集合，其内容摘录如下。此时，password 被进行了加密处理。

```
admin:
OBF:1v2j1uum1xtvlzejlzer1xtnluvk1vlv,server-administrator,content-administrator,adm
in
```

从本章前面内容获悉，Pentaho 管理控制台运行在 Jetty Web 容器中，而上述密文信息正是借助 Jetty 的实用类生成的，下面给出了操作示例。用户可以根据自身情况设定各自的系统管理员帐号信息。

```
D:\administration-console>java -cp lib/jetty-6.1.2.jar;lib/jetty-util-6.1.9.jar
org.mortbay.jetty.security.Password admin password
password
OBF:1v2j1uum1xtvlzejlzer1xtnluvk1vlv
MD5:5f4dcc3b5aa765d61d8327deb882cf99
CRYPT:advwtv/9yU5yQ
```

另外，在某些时候，系统管理员希望启用 HTTPS，以保护他同 Pentaho 管理控制台间的交互。通过修改位于 D:\administration-console\resource\config 目录中的 console.properties 属性文件能够达到这一目的，下面给出了相关配置。将“console.ssl.enabled”的取值置为 true 后，我们便启用了 HTTPS。此后，系统管理员将通过 <https://localhost:8043> 访问管理控制台。console.properties 还包括了其他一些同 SSL 相关的重要配置信息，比如服务器 X.509 证书的存放位置、HTTPS 端口。注意，Pentaho 管理控制台提供的默认服务器证书已经过期，它存储在 D:\administration-console\resource\config 目录的 keystore 中。

```
console.ssl.enabled=false
```

有关 SSL X.509 证书的更多细节，读者可以参考作者写作的《实战 Spring Security 3.x：快速构建企业级安全》电子图书。

2.3 小结

本章大致介绍了 Pentaho BI 套件的核心组件，Pentaho BI 服务器和管理控制台。本书后续章节要经常同它们打交道，所以要重视这些内容。关于 BI 平台的具体操作，我们将结合特定的 BI 场景展开论述。

下章内容将进入到 Kettle ETL 解决方案中。

3 数据加工王者—Kettle

ETL 在 BI 领域占据着重要的位置，几乎所有的 BI 项目都要实施数据加工行为，比如为统一各业务系统的数据口径、对业务数据进行粗和精加工。在功能及性能表现上，Kettle 是一款重量级的 ETL 产品；而在安装、使用及部署方面，它又是一款非常轻便的产品。本章将围绕 Kettle 的各个方面进行阐述。

3.1 ETL及Kettle概述

ETL (Extract、Transform、Load，抽取、转换、装载)，它是 BI 项目中最常见、基础的数据加工行为。构建数据仓库期间，各类业务系统的数据需要经过严格的 ETL 过程，才能够进入到数据仓库中，进而为后续的数据展现、分析提供支撑。通常，由于企业的各业务系统数据口径不一致，比如不同应用存储性别的方式存在差异性、银行应用中不同币种的统一、零售应用中商品计价方式的统一等，使得 BI 项目必须实施 ETL 工作，否则在含糊、不准确的数据上进行各种数据行为是徒劳的、没有意义的。

在另外一些场合，企业往往需要对 TB 级别的数据进行各种数据聚合、粗和精加工。比如，在制作即席报表期间，用户希望这些报表的运行时间越短越好，然而如果报表使用到的数据粒度很细、数据量很大，则要控制好报表的运行时间估计够呛。此时，我们往往需要对数据进行各种层次的聚合操作，比如可以将“日”级别存储的数据预先聚合成按周、月、季度的数据。将来，运行报表的时间将得到有效控制，毕竟 RDBMS 能够更快速响应客户提交的 SQL 请求了。同样地，进行即席 OLAP 多维分析期间，数据聚合及粗、精加工环节不可或缺。有关数据聚合的相关阐述，本书第 5 章内容将会有相关专题讨论它。

为有效实施 ETL 工作，我们可借助 Kettle，即 Pentaho 数据集成解决方案。

3.1.1 基于“流”架构的Kettle

Kettle (<http://kettle.pentaho.org/>)，这是一款历史悠久的 ETL 产品，存在一支稳定的开发团队在发展着它，而 Matt Casters 一直在负责 Kettle 的发展。

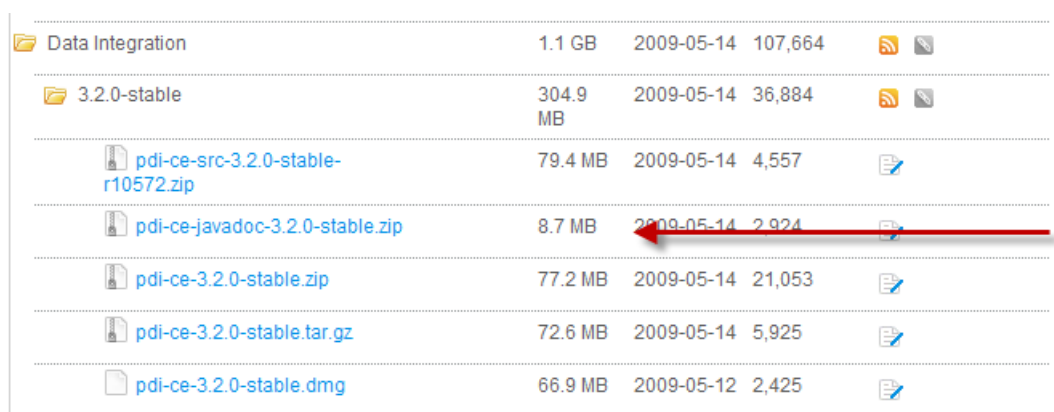
在使用 Kettle 的这些年里，最让我印象深刻的是它的数据加工性能，毕竟大部分 ETL 项目涉及的数据量都很大。记得在 Kettle 官方论坛有这么一个比方，说 Kettle 加工数据的过

程正如河流里水的流动过程。一切能够并行加工数据的地方都是并行进行的，而且数据都是从上游流向下游，河床里水的流动又何尝不是呢？某些场合，Kettle 胜过河水的流动，因为它允许用户随意拓宽或收窄“河床”，比如动态添加或减少新的 ETL 执行引擎。

接下来，让我们一起去感受数据在 Kettle 中的流动。

3.1.2 下载及安装 Kettle

用户将浏览器定位到 <http://sourceforge.net/projects/pentaho/files/> 网址后，并找到图 3-1 类似内容，Kettle 下载入口便呈现在眼前。



Data Integration	1.1 GB	2009-05-14	107,664	
3.2.0-stable	304.9 MB	2009-05-14	36,884	
pdi-ce-src-3.2.0-stable-r10572.zip	79.4 MB	2009-05-14	4,557	
pdi-ce-javadoc-3.2.0-stable.zip	8.7 MB	2009-05-14	2,924	
pdi-ce-3.2.0-stable.zip	77.2 MB	2009-05-14	21,053	
pdi-ce-3.2.0-stable.tar.gz	72.6 MB	2009-05-14	5,925	
pdi-ce-3.2.0-stable.dmg	66.9 MB	2009-05-12	2,425	

图 3-1 Kettle 的下载

其中，pdi-ce-3.2.0-stable.zip 是 Kettle 正式发布版，而 pdi-ce-src-3.2.0-stable-r10572.zip 是对应的源码，当我们需要深入研究或扩展 Kettle 时，这一源码是不可或缺的。

<http://source.pentaho.org/svnkettleroot/Kettle>，如果需要，我们也可以从这一 SVN 库获得各个版本的 Kettle 源码（包括最新代码快照），并手工构建或自定义自身的 Kettle 版本。比如，下面给出了手工构建 Kettle 的操作示例。用户要准备好 Ant（<http://ant.apache.org/>）。

```
D:\springsource\workspace\Kettle>ant
```

大约几分钟后，构建好的 Kettle 被存放在 D:\springsource\workspace\Kettle\distrib 目录，其内容同解压后的 pdi-ce-3.2.0-stable.zip 类似，这里将它解压到 D:\ 目录，即 D:\data-integration 位置将持有 Kettle。

为配合 ETL 工作的顺利实施，Kettle 内置了大量的实用工具，比如用于设计转换和作业的 Spoon IDE、执行转换的 Pan、执行作业的 Kitchen、添加新 ETL 执行引擎的 Carte 等。

3.2 Spoon—设计转换及作业集成开发环境

本节将围绕 Spoon IDE 的使用展开论述。

3.2.1 启动Spoon

单击 D:\data-integration 目录中的 Kettle.exe 或 Spoon.bat，用户便能启动 Spoon IDE。在此之前，用户需要安装并配置好 JDK，比如于 D:\jdk1.6.0_18 位置安装好 Java SE 6.0。Spoon 是用于开发、调试、测试 ETL 转换和作业的 IDE。启动 Spoon 期间，图 3-2 界面会出现在用户面前。Kettle 允许用户将 ETL 转换和作业存储到文件、RDBMS 等地方，这一界面允许用户选择适合自身的存储方式，即资源库的选择。我们暂时采用文件方式存储 ETL 转换和作业，即“没有资源库”。本章后续内容会重点讨论资源库问题。



图 3-2 启动界面

单击“没有资源库”按钮后，Spoon IDE 的主界面将呈现在用户面前，即图 3-3。

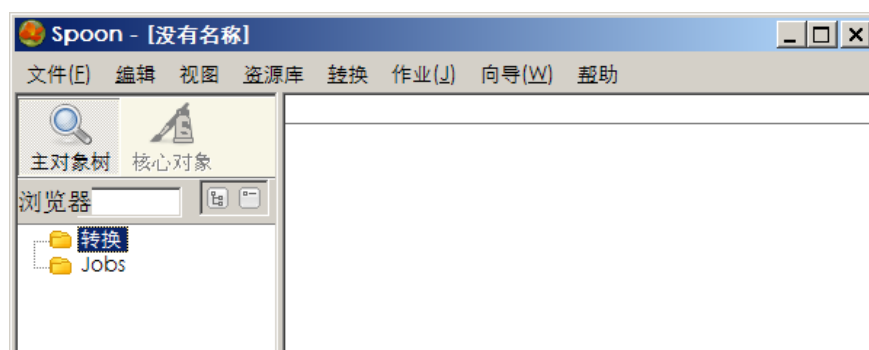


图 3-3 Spoon IDE 主界面

在 Kettle ETL 解决方案中,主要存在两种 ETL 工件:转换(Transformation)和作业(Job)。

ETL 转换,专注于数据加工本身,比如装卸数操作、数据编码转换; ETL 作业,专注于流程控制,比如执行若干 ETL 转换、将加工后的文件借助 SSH2 传输出去等。通常, ETL 作业会包含若干 ETL 转换,并控制它们的执行,而且作业会以一定周期执行,比如每周二执行、每隔 3 小时执行等。

3.2.2 从Kettle内置的ETL转换和作业示例谈起

为加快 Kettle 的入门和使用,其内置了大量的 ETL 转换(.ktr)和作业(.kjb)示例,具体位置存在于 D:\data-integration\samples 目录,比如 transformations 子目录内置了大量的转换示例,而 jobs 子目录内置了大量的作业示例。这些示例展示了各个 Kettle 内置组件的使用。这里以“Text File Output - Number formatting.ktr”转换为例,图 3-4 展示了打开这一转换后的 Spoon IDE。

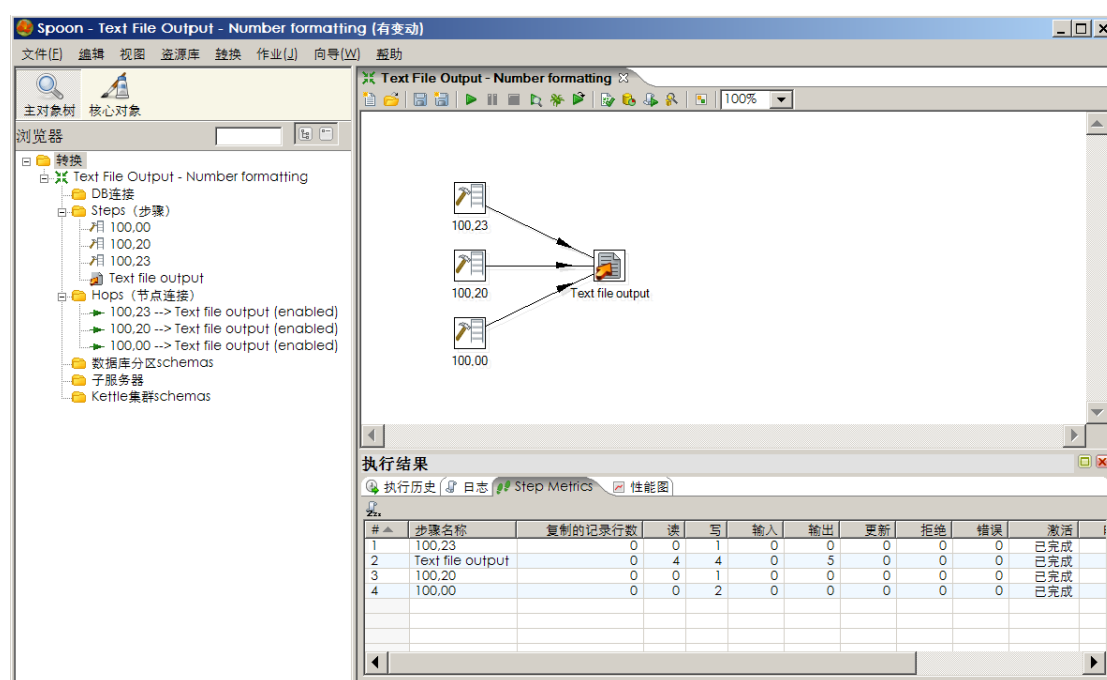


图 3-4 打开“Text File Output - Number formatting.ktr”转换

在“Text File Output - Number formatting.ktr”转换中,它借助“生成记录”组件(Step)生成了一些数据,并作为转换的输入,随后借助“文本文件输出”组件将这一转换的数据加工结果输出到一文本文件中,即位于 D:\data-integration\samples\transformations\output 目录的 number_formatting_sample.txt 文件。在运行(F9)这一转换后, number_formatting_sample.txt

文件将持有如下类似内容。通过按住“Shift”键，可以将上下游的 ETL 转换组件连接起来。

```
srinu
100.23
100.00
100.00
100.20
```

图 3-5 展示了 Kettle 内置的、服务于 ETL 转换的组件分类集合。本书附录 A 详细阐述了这些分类中各个组件的使用。这些组件基本上能够满足各种 ETL 数据加工场景，一旦不行，则用户可以考虑扩展它们，甚至借助 Kettle 暴露的各种接口开发新的 ETL 转换组件。

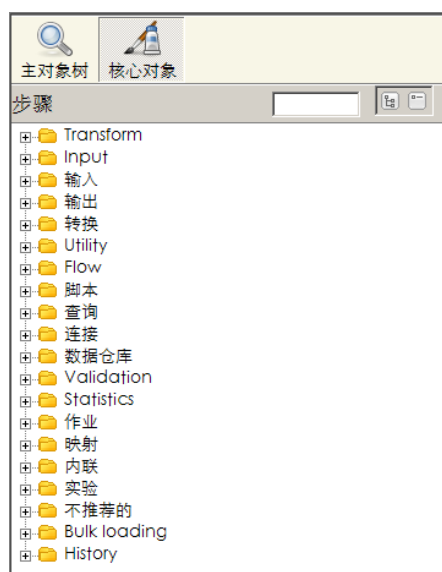


图 3-5 Kettle 内置的 ETL 转换组件分类集合

通常，为实现某一 ETL 数据加工目的，用户的选择很多，不同组件能够实现同样的目的，比如就文本文件的输入而言，就存在 CSV file input、Fixed file input、文本文件输入等组件。这就要求用户在选用它们之前要了解到这些组件的各自习性。在数据读取性能方面，文本文件输入组件很差，它不如 CSV file input 和 Fixed file input 组件，因为后两者启用了 Java NIO 技术实现。

当然，用户也需要去斟酌各个组件的具体使用选项，因为启用参数的不同使得数据加工的性能也会存在很大差别，注意这种差别是惊人的，数据加工速度可能从 2 分钟提高到 10 秒。这话不是没有依据的，作者在不少项目中就经常碰到这种问题。就拿上述“Text File Output - Number formatting.ktr”转换使用到的文本文件输出组件来说，图 3-6 暴露了其内置的“Fast data dump”选项。当大量被加工后的数据需要落地到文本文件中，而且文本文件输出组件明显成为瓶颈时，启用这一选项所带来性能上的提升可能是用户不敢想的。道理也

很简单，启用“Fast data dump”选项后，ETL 转换的执行将避免大量的 CPU 运算，直接将 byte[] 字节流落地到文本文件中，这既保证了内容的正确性，又省去了大量的 CPU 运算周期。类似地，启用 CSV file input 和 Fixed file input 组件的“Lazy conversion”选项能够加快文件的读取速度。



图 3-6 文本文件输出组件内置的“Fast data dump”选项

值得注意的是，用户需要从整个转换的角度考虑各组件选项的使用，千万不能够“只见水花、不见河流”。在开发 Kettle ETL 转换期间，用户一定要多实践、多思考，并总结不同 ETL 转换组件的使用心得，比如将 Kettle 内置的转换示例多运行几次，并试图修改它们中的部分内容。这里再以“CSV Input - Reading customer data.ktr”转换为例，见图 3-7。

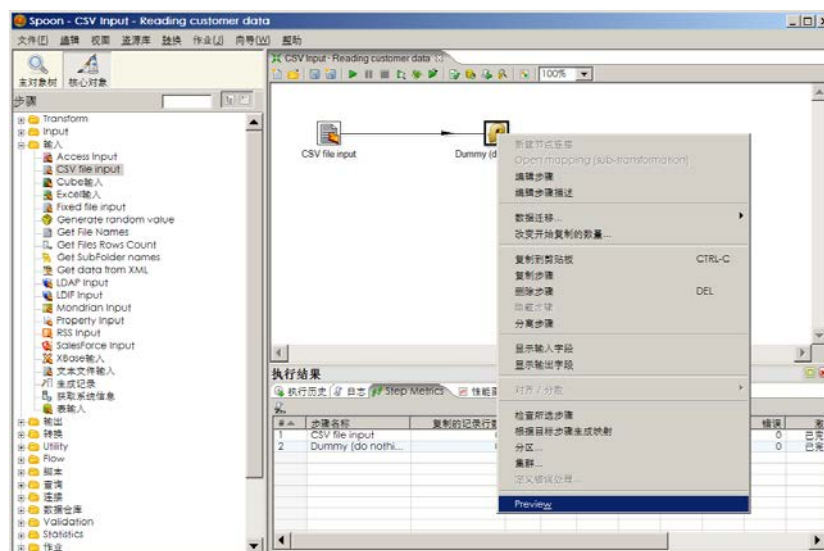


图 3-7 预览 (F10) “CSV Input - Reading customer data.ktr” ETL 转换

这一转换使用到一“Dummy(do nothing)”特殊组件，即“空操作（什么也不做）”组件。它会将当前 ETL 转换上游传递过来的已加工数据直接输出在界面上（或流向下游）。通常，

用户可以借助这类组件进行转换的调试工作，比如 ETL 转换的运行结果同预期不一致时，借助 Dummy 组件能够快速定位到问题的根源。也正因为 Dummy 组件的特殊目的，用户需要以预览（F10）方式运行其所在的转换。在另外一些场景，我们可借助 Dummy 组件进行上游数据的汇总操作，并将汇总后的数据流向整个转换的下游。

接下来研究“Database - generic driver usage.ktr” ETL 转换示例，见图 3-8。这一转换需要使用到 RDBMS，因为它使用到表输入（Table input）组件。

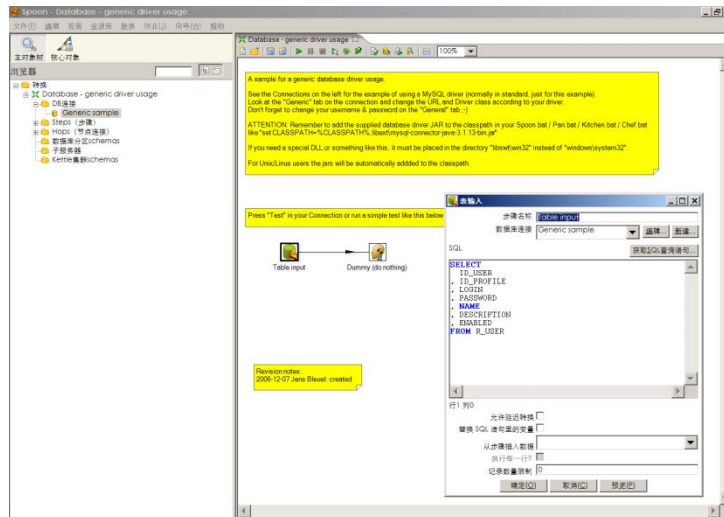


图 3-8 “Database - generic driver usage.ktr” ETL 转换

通过左边浏览器中“DB 连接”项，用户能够调整具体的数据库连接设置，见图 3-9。比如，用户可以采用不同数据库（从 MySQL 切换到 Oracle）、切换访问数据库的方式（JDBC 或 ODBC 或 OCI 或 JNDI 等）。

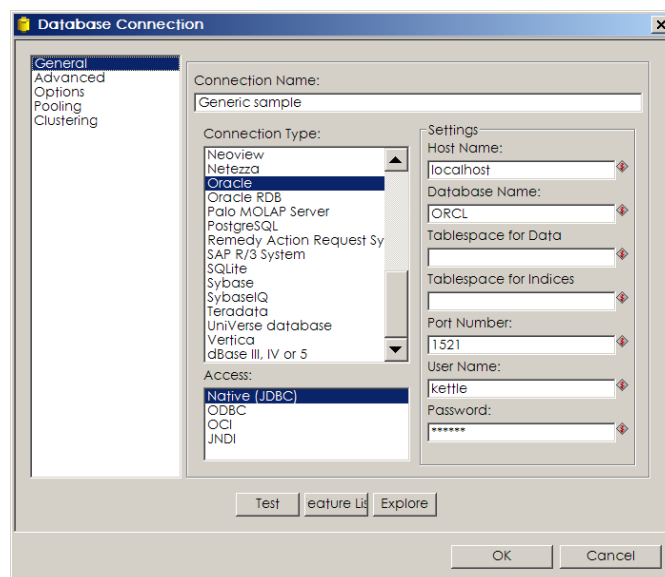


图 3-9 配置“数据库连接”

接下来，我们来研究 Kettle 内置的 ETL Job 示例，比如 Evaluate result rows.kjb 作业，见图 3-10。它使用到 Create result rows.ktr 转换。通过 F9 快捷键能够启动 ETL 作业的运行。

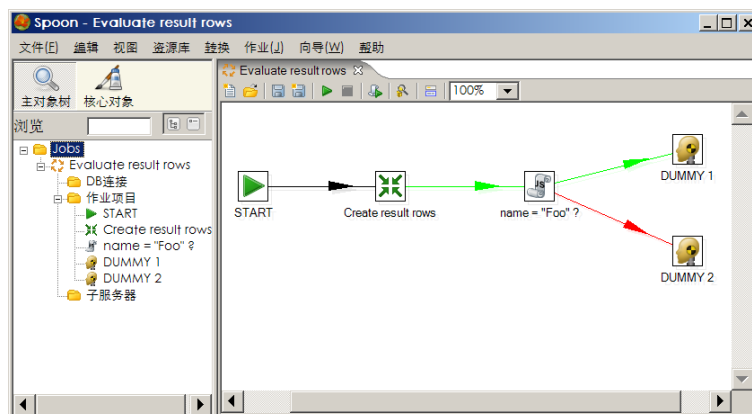


图 3-10 “Evaluate result rows.kjb” 作业

通常，ETL Job 会使用到 START 组件，借助它启动作业的运行。而且，它还允许用户自定义当前作业的调度时机，比如图 3-11 要求这一作业每隔 5 秒钟运行 1 次。

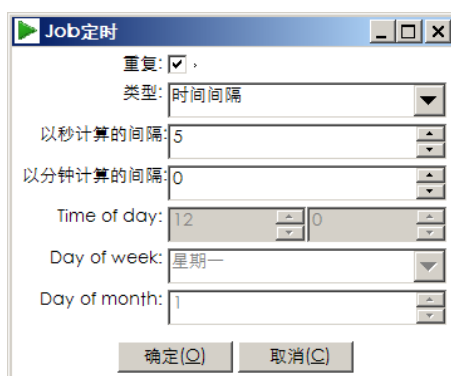


图 3-11 START 组件

图 3-12 展示了 Kettle 内置的、服务于 ETL 作业的组件分类集合。本书附录 A 详细阐述了这些分类中各个组件的使用。这些组件基本上能够满足各种 ETL 数据加工场景，一旦不行，则用户可以考虑扩展它们，甚至借助 Kettle 暴露的各种接口开发新的 ETL 作业组件。

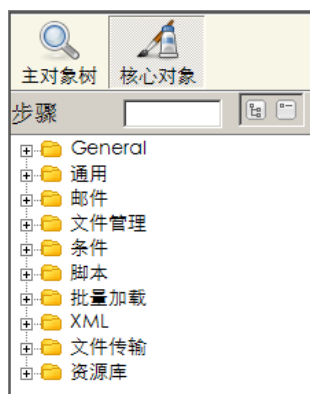


图 3-12 Kettle 内置的 ETL 作业组件分类集合

为在 ETL 作业中引用到具体的 ETL 转换，用户需要使用到 Transformation 作业组件，它位于“通用”分类组件中。

3.2.3 监控 ETL 转换的执行性能

在执行 ETL 转换期间，用户可以启用 Spoon 内置的性能监控支持。这一特性用来监控当前 ETL 转换中各组件（步骤，Step）的执行情况，见图 3-13。默认时，它会每隔 1 秒钟收集 1 次性能数据。



图 3-13 启用 ETL 转换的性能监控支持

图 3-14 给出了“Text File Output - Number formatting.ktr” ETL 转换的执行结果，它展示了相应的执行性能。作者调整了 Kettle 内置的这一转换，主要是大大增加了“生成记录”组件生成的数据量，并启用了 Text file output 组件的“Fast data dump”特性。

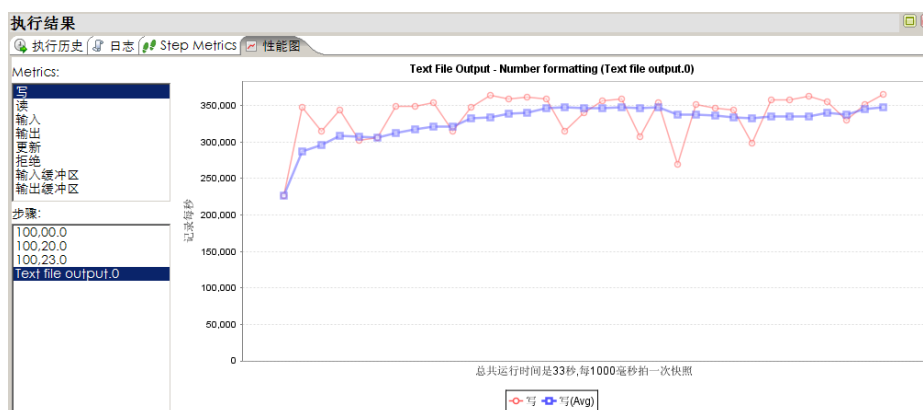


图 3-14 监控“Text File Output - Number formatting.ktr” ETL 转换的执行性能

可以看出，此时的 Text file output 组件“写”文本文件的速度较为平稳。在实际开发 ETL 转换期间，图中展示的性能指标对于用户诊断和提升数据加工性能非常有用处。比如，在从 RDBMS 进行卸数操作时，用户能够知道 ETL 转换的哪一环节（组件）存在瓶颈，并采取进一步行动。再比如，在加工大量数据文件时，如果文件的落地操作过于频繁，则“读”指

标一定会凸显出来，进而让用户减少文件的落地次数。因此，用户要合理使用好 Spoon IDE 的这一特性。

3.2.4 调整宿主Spoon IDE的JVM内存

某些 ETL 转换和作业的执行需要耗费大量的内存。在开发及测试它们期间，用户需要调整 Spoon IDE 使用的 JVM，比如它占用的 JVM 内存情况。

如果是通过 Kettle.exe 可执行程序启动 Spoon IDE 的，则需要调整 Kettle.l4j.ini 配置文件，比如将内置的-Xmx256M 调整成-Xmx768M 或更大。

如果是通过 Spoon.bat 脚本文件启动 Spoon IDE 的，则需要调整 Spoon.bat 本身，调整内容同上述一致。

3.3 将转换和作业进行外在化管理

默认时，Kettle ETL 转换和作业直接用文件保存，这显然不适合大规模团队。如果考虑将这些文件存放到 SVN 或其它 SCM 配置工具中，则也是一种选择。或者，Kettle 允许用户将 ETL 转换和作业存储到其它位置，比如 RDBMS 中。

3.3.1 存储到数据库中—以Oracle为例

为了用数据库取代文件系统存储 ETL 转换和作业，用户需要在启动 Spoon IDE 时创建一新的资源库。在创建资源库前，用户要提供一 Oracle 数据库连接用户，比如 kettle/kettle。然后，创建好相应的数据库连接（比如 kettle），再并给出资源库的名称，比如 kettle-repos。最后，用户需要单击图 3-15 中给出的“创建或更新”按钮，并完成资源库的创建工作。此时，Spoon 会在 Oracle kettle/kettle 用户中自动创建大量的表，它们用来存储 ETL 转换和作业。

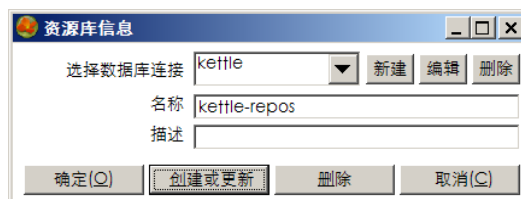


图 3-15 创建一“kettle-repos”资源库

随后，用户可以以 admin/admin 帐号登录到 kettle-repos 资源库中，并启动 Spoon IDE，进而完成各种 ETL 转换和作业的开发工作，见图 3-16。



图 3-16 admin/admin 帐号连接到 kettle-repos 资源库

图 3-17 展示了 kettle-repos 资源库中持有的数据库表集合（kettle/kettle）。

+	R_CLUSTER
+	R_CLUSTER_SLAVE
+	R_CONDITION
+	R_DATABASE
+	R_DATABASE_ATTRIBUTE
+	R_DATABASE_CONTYPE
+	R_DATABASE_TYPE
+	R_DEPENDENCY
+	R_DIRECTORY
+	R_JOB
+	R_JOBENTRY
+	R_JOBENTRY_ATTRIBUTE
+	R_JOBENTRY_COPY
+	R_JOBENTRY_TYPE
+	R_JOB_ATTRIBUTE
+	R_JOB_HOP
+	R_JOB_NOTE
+	R_LOG
+	R_LOGLEVEL
+	R_NOTE
+	R_PARTITION
+	R_PARTITION_SCHEMA
+	R_PERMISSION
+	R_PROFILE
+	R_PROFILE_PERMISSION
+	R_REPOSITORY_LOG
+	R_SLAVE
+	R_STEP
+	R_STEP_ATTRIBUTE
+	R_STEP_DATABASE
+	R_STEP_TYPE
+	R_TRANSFORMATION
+	R_TRANS_ATTRIBUTE
+	R_TRANS_CLUSTER
+	R_TRANS_HOP
+	R_TRANS_NOTE
+	R_TRANS_PARTITION_SCHEMA
+	R_TRANS_SLAVE
+	R_TRANS_STEP_CONDITION
+	R_USER
+	R_VALUE
+	R_VERSION

图 3-17 Kettle 资源库持有的表集合

3.4 Kettle内置的ETL相关辅助工具

为加快 Kettle ETL 解决方案的快速实施，Kettle 内置了大量的辅助工具。接下来，我们一一研究它们。

3.4.1 Pan—执行转换

Pan 命令行用于执行 ETL 转换。至于 Kettle ETL 转换的位置，Pan 并不作假定，比如存储在文件中，或者在数据库中。下面给出了 Pan 内置的命令行选项。

```
D:\data-integration>Pan.bat
Options:
  /rep      : 资源库名称
  /user     : 资源库用户名
  /pass     : 资源库密码
  /trans    : 要启动的转换名称
  /dir      : 目录(不要忘了前缀 /)
  /file     : 要启动的文件名(转换所在的 XML 文件)
  /level    : 日志等级(基本, 详细, 调试, 行级, 错误, 没有)
  /logfile  : 要写入的日志文件
  /listdir  : 列出资源库里的目录
  /listtrans : 列出指定目录下的转换
  /listrep  : 列出可用资源库
  /exprep   : 将资源库里的所有对象导出到 XML 文件中
  /norep    : 不要将日志写到资源库中
  /safemode : 安全模式下运行: 有额外的检查
  /version  : 显示版本, 校订和构建日期
  /param    : Set a named parameter <NAME>=<VALUE>. For example -param:FOO=bar
  /listparam : List information concerning the defined named parameters in the
specified transformation.
```

下面展示了/file 选项的使用，它将触发 DistinctCount.ktr 转换的执行。

```
Pan /file D:\data-integration\samples\transformations\DistinctCount.ktr
```

如果 DistinctCount 转换存储在 kettle-repos 资源库中，则可以借助如下选项集合运行它。

```
Pan /rep kettle-repos /user admin /pass admin /trans DistinctCount
```

3.4.2 Kitchen—执行作业

Kitchen 命令行用于执行 ETL 作业。至于 Kettle ETL 作业的位置，Kitchen 并不作假定，比如存储在文件中，或者在数据库中。下面给出了 Kitchen 内置的命令行选项。

```
D:\data-integration>Kitchen.bat
Options:
  /rep      : Repository name
  /user     : Repository username
  /pass     : Repository password
  /job      : The name of the job to launch
  /dir      : The directory (dont forget the leading /)
  /file     : The filename (Job XML) to launch
  /level    : The logging level (Basic, Detailed, Debug, Rowlevel, Error, Nothing)
  /logfile  : The logging file to write to
  /listdir  : List the directories in the repository
  /listjobs : List the jobs in the specified directory
  /listrep  : List the available repositories
  /norep    : Do not log into the repository
  /version  : show the version, revision and build date
  /param    : Set a named parameter <NAME>=<VALUE>. For example -param:FOO=bar
  /listparam : List information concerning the defined parameters in the specified job.
  /export   : Exports all linked resources of the specified job. The argument is the
name of a ZIP file.
```

下面展示了/file 选项的使用，它将触发 Evaluate result rows.kjb 作业的执行。

```
Kitchen /file "D:\data-integration\samples\jobs\evaluate-result-rows\Evaluate result
rows.kjb"
```

如果 Evaluate result rows.kjb 作业存储在 kettle-repos 资源库中，则可以借助如下选项集合运行它。

```
Kitchen /rep kettle-repos /user admin /pass admin /job "Evaluate result rows"
```

3.4.3 Carte—添加新的ETL执行引擎

Carte 类似于 Pentaho 管理控制台，它们都宿主在 Jetty Web 容器中，但各自承担的使命不同。Carte 用于远程执行 Kettle ETL 转换和作业。下面给出了 Carte 内置的命令行选项。

```
D:\data-integration>Carte.bat
Usage: Carte [Interface address] [Port]

Example: Carte 127.0.0.1 8080
Example: Carte 192.168.1.221 8081

Example: Carte /foo/bar/carte-config.xml
Example: Carte http://www.example.com/carte-config.xml
```

图 3-18 通过执行“Carte.bat localhost 8081”命令行启动了新的 ETL 执行引擎，即宿主了 Kettle ETL 引擎的 Jetty Web 容器。

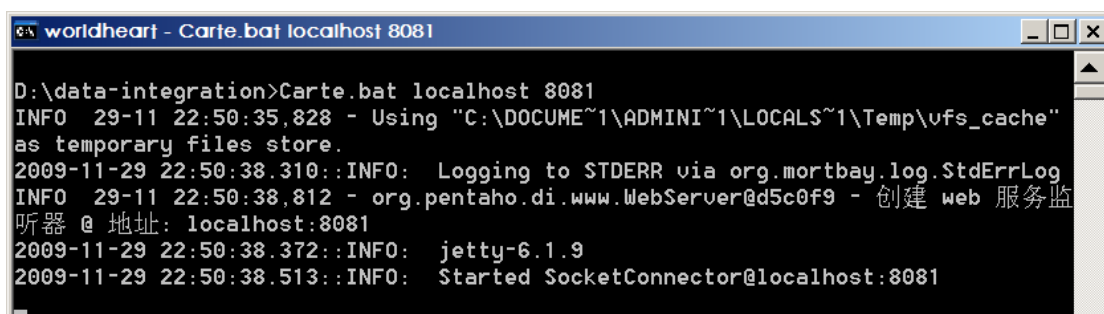


图 3-18 启动新的 ETL 执行引擎

为在 Spoon IDE 中测试 ETL 转换或作业, 用户可以选择让它们在远程 Carte 实例中执行。在这之前, 我们必须在对应的转换或作业中配置子服务器, 见图 3-19。为保护 Carte 实例, Carte 启用了安全性认证, 默认的登录用户是 cluster/cluster。



图 3-19 为 ETL 转换或作业的执行添加子服务器

最后, 在选择执行 ETL 转换或作业的方式时, 用户需要选中“远程执行”, 并选中合适的远程机器, 见图 3-20。



图 3-20 在远程机器中执行 ETL 转换或作业

此时, 透过运行 Carte 的 DOS 控制台, 用户能够看到 ETL 转换或作业的执行。或者, 用户可以打开浏览器, 并定位到 <http://localhost:8081/> 位置, 输入 cluster/cluster 登录帐号后,

便能够操控到其中的各 ETL 转换或作业，见图 3-21。

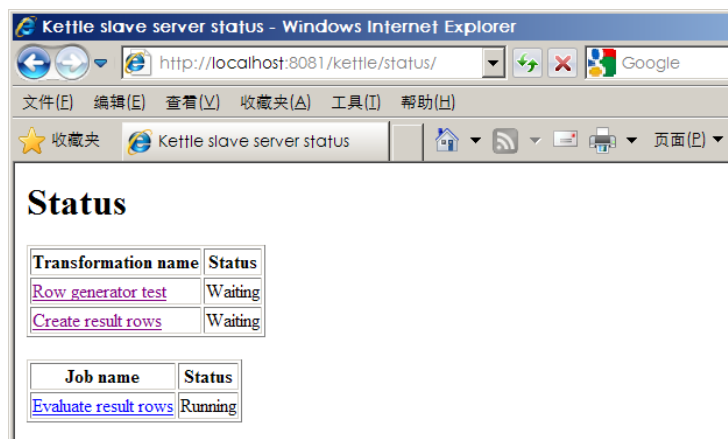


图 3-21 Carte Web 操作界面

3.4.4 Encr加密工具

如果需要调整 Kettle 资源库或 Carte 的登录用户，比如用户密码，则可借助 Kettle 内置的 Encr 加密工具。

默认时，Carte 将登录用户存储在 D:\data-integration\pwd 的 kettle.pwd 文件中。比如，下面对“password”密码进行了处理，如果将生成的“OBF:1v2jluumlxtvlzejlzerlxtnluvklv1v”替换 kettle.pwd 中的对应内容，则以后要同 Carte 打交道必须用 cluster/password 帐号。甚至，如果需要，用户可以直接将用户名直接替换成其他的。

```
D:\data-integration>Encr.bat -carte password
OBF:1v2jluumlxtvlzejlzerlxtnluvklv1v
```

为修改 Kettle 资源库中 admin 帐号的密码，用户可以执行如下命令行。随后，将生成的“2be98afc86aa7f2e4bb18bd63c99dbdde”密文覆盖 Oracle kettle/kettle 数据库中 R_USER 表的对应数据。此后，只有 admin/password 帐号才能够连接到 Kettle 资源库了。

```
D:\data-integration>Encr.bat -kettle password
Encrypted 2be98afc86aa7f2e4bb18bd63c99dbdde
```

Encr 加密工具非常实用。

3.5 基于集群并发加工大批量数据

3.5.1 静态集群模式

3.5.2 动态集群模式

3.6 与Pentaho BI服务器的集成

3.7 自定义及扩展Kettle

3.8 Kettle最佳实践

3.8.1 善待Kettle内置的变量集合

3.9 其他ETL解决方案

这里主要以 IBM DataStage 和 Spring Batch 为例。

3.9.1 同IBM DataStage的对比

作者看到，IBM DataStage 在国内外 BI 项目的出现还是很频繁的，它毕竟是行业内很有分量的一款 ETL 工具。作为 IBM 数据集成解决方案的拳头产品，我们也需要时常关注它的发展，在某种程度上，IBM DataStage 在引领 ETL 领域朝前发展。

有关 IBM DataStage 的更多内容, 本书不想详细讨论, 但有几方面的内容需要交代一下。

其一, 同 Kettle 相比, 作者认为 IBM DataStage 过于笨重。无论是它的安装过程, 还是运行时对机器物理资源的消耗来看。

其二, 同 Kettle Spoon IDE 相比, IBM DataStage 内置的 ETL 设计器在功能上并没有占据优势。比如, Spoon 对 ETL 工件的开发、测试、调试、性能监控提供了端到端的解决方案, 而且非常轻量。

其三, 部署 Kettle ETL 工件是一件非常轻松的事情, 整个 Kettle ETL 解决方案可以在同一台机器上完成, 这在效率优先的今天尤为重要。

其四, IBM DataStage 没有传说中的那么神。比如, ETL 作业支持中文方面存在欠缺、偶尔会出现 ETL 作业长时间执行不下去的问题。可见, 是软件就有 Bug。相比之下, 用户可以掌控开源 Kettle 的一切。

可以这么认为, 80% 的 ETL 项目应该变得更加敏捷, 而 Kettle 可帮助客户做到这一点, 这一点作者坚信不疑!

3.9.2 Spring Batch—另一种风格的ETL解决方案

在使用 Kettle 实施 ETL 项目时, 用户几乎不用编写任何 (Java) 代码, 借助 Spoon IDE 能够完成各种复杂程度的 ETL 转换和作业。与此同时, 如果打算将 Kettle 执行引擎 (包括 kettle-core.jar、kettle-db.jar、kettle-engine.jar) 嵌入到企业应用中, 则尽管要编写一些集成代码, 但 ETL 转换和作业仍然可以用 Spoon IDE 设计。因此, 业务人员可以是潜在的 Kettle 用户, 并且他们可以去设计和开发各种 ETL 工件。

相比之下, 借助 Spring Batch (<http://static.springsource.org/spring-batch/>) 实施 ETL 项目的用户显得没有这么幸运。因为他们必须掌握 Java EE 开发知识。Spring Batch 采纳 Spring 编程模型开发 ETL 作业等工件, 这使得它无疑是最为灵活的 ETL 解决方案之一, 因为用户可以在任意场合灵活使用它构建出各种风格的 ETL 行业应用。

附录 B 针对 Spring Batch 给出了较多篇幅, 开发者不应该错过这一内容的阅读。

3.10 小结

Kettle 是企业级、ETL 能力很强的产品, 加上它的使用简单、部署方便, 进而得到了广

大用户的欢迎。本章对 **Kettle** 进行了全方位阐述，无论是简单 ETL 场景，还是复杂的集群环境，它都能够胜任。甚至，用户还可以对它进行二次开发，借助 **Kettle** 丰富的扩展能力能够满足用户的各种需求。本书附录 A 还针对 **Kettle** 内置的各种组件的使用进行了详细阐述。

在进入 **Pentaho** 报表工具的世界前，我们来仔细探讨一下 **Action Sequence**，它是 **Pentaho BI** 解决方案中的中坚力量。

4 Action Sequence—集大成者

Action Sequence 能够完成各种复杂程度的 BI 工作，它是 Pentaho BI 解决方案中非常重要、基础的特性。本章将围绕它展开论述。

4.1 Action Sequence概述

Action Sequence 字面意思已经告诉我们，它能够以一定顺利执行一系列动作，从而完成某项 BI 工作，比如执行完某 Kettle ETL 转换后，将执行结果通过邮件发送出去。下面摘录了 GetPDIEEnvironment.xaction 的内容，位于 D:\biserver-ce\pentaho-solutions\bi-developers\etl 中。GetPDIEEnvironment.xaction 正是一 Action Sequence，这是一标准的 XML 文档。

```
<?xml version="1.0" encoding="UTF-8"?>
<action-sequence>
  <name>SampleTransformation.xaction</name>
  <title>%title</title>
  <version>1</version>
  <logging-level>debug</logging-level>
  <documentation>
    <author>Jens Bleuel</author>
    <description>%description</description>
    <help>%help</help>
    <result-type>rule</result-type>
    <icon>HelloETL.png</icon>
  </documentation>

  <inputs/>

  <outputs>
    <rule-result type="result-set"/>
  </outputs>

  <resources>
    <transformation-file>
      <solution-file>
        <location>GetPDIEEnvironment.ktr</location>
        <mime-type>text/plain</mime-type>
      </solution-file>
    </transformation-file>
  </resources>

  <actions>
    <action-definition>
      <component-name>KettleComponent</component-name>
      <action-type>Execute Kettle Transformation</action-type>
```

```

<action-inputs/>
<action-resources>
  <transformation-file type="resource"/>
</action-resources>
<action-outputs>
  <transformation-output type="result-set" mapping="rule-result"/>
</action-outputs>
<component-definition>
  <importstep><![CDATA[result]]></importstep>
</component-definition>
</action-definition>
</actions>
</action-sequence>

```

图 4-1 展示了这一 Action Sequence 的执行结果。可以看到，Action Sequence 是能够被 Pentaho BI 服务器直接执行的。

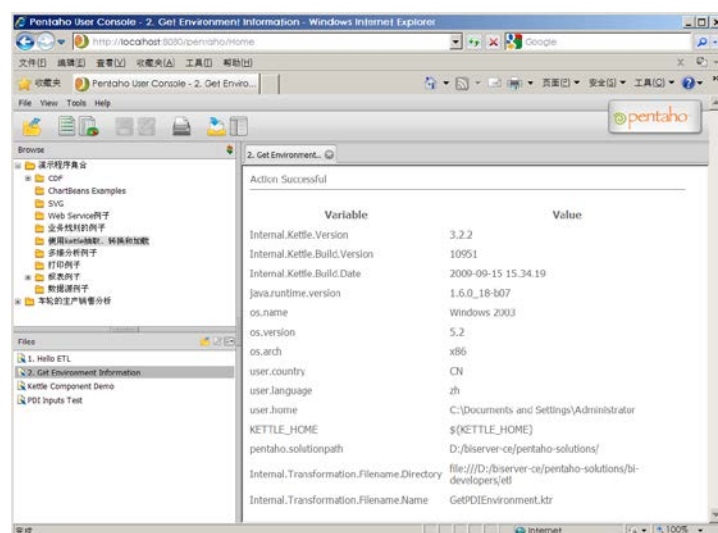


图 4-1 运行 GetPDEnvironment.xaction

GetPDEnvironment.xaction 对应的 URL 如下。直接在浏览器中敲入这一地址也能够访问到它，但必须事先登录到 Pentaho BI 服务器中。

<http://localhost:8080/pentaho/ViewAction?&solution=bi-developers&path=etl&action=GetPDEnvironment.xaction>

借助 Pentaho Design Studio (简称 PDS) 开发工具，开发者能够快速完成 Action Sequence 的制作，下面先来了解一下这一开发工具。

4.1.1 Pentaho Design Studio 开发工具

透过 <http://sourceforge.net/projects/pentaho/files/> 网址，开发者能够下载到 PDS。具体见

图 4-2。PDS 需要宿主到 Eclipse 中，即它是以 Eclipse 插件形式存在的，开发者可以下载内置了 Eclipse 的版本，比如 pds-ce-win-3.5.0.stable.zip（针对 Windows）。或者，可以下载仅含 PDS 插件的压缩包，比如 org.pentaho.designstudio.editors.actionsequence_3.5.0.stable.zip。

▼ Design Studio	3.9 GB	2009-10-15	99,797	
▼ 3.5.0-stable	496.2 MB	2009-10-15	6,607	
org.pentaho.designstudio.editors.actionsequence_3.5.0.stable.zip	23.0 MB	2009-10-15	1,307	
pds-ce-win-3.5.0.stable.zip	148.9 MB	2009-10-15	3,694	
pds-ce-mac-3.5.0.stable.tar.gz	167.2 MB	2009-10-15	319	
pds-ce-linux-3.5.0.stable.tar.gz	157.3 MB	2009-10-15	1,287	

图 4-2 PDS 的下载

下载 org.pentaho.designstudio.editors.actionsequence_3.5.0.stable.zip 后，直接将它解压到现有的 Eclipse IDE(STS)中，并重启 Eclipse，即完成了 PDS 的安装。开发者可以通过 Eclipse 菜单、Wizard、工具栏触发 Action Sequence 的新建工作，具体见图 4-3。

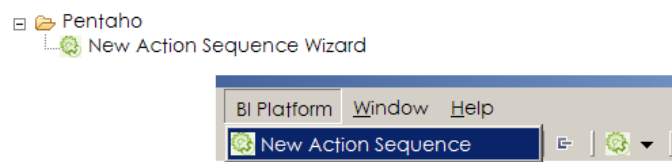


图 4-3 新建 Action Sequence 的菜单、Wizard 及工具栏

新建 Action Sequence 前，开发者需要新建一 Eclipse 工程。或者，开发者可以将 Pentaho BI 服务器内置的 bi-platform-sample-solution 项目（D:\biserver-ce\pentaho-solutions 位置）导入到 Eclipse 中。好了，这一 Eclipse 工程便可用于存储新建的 Action Sequence 了。

现在，开发者可以去试图打开 D:\biserver-ce\pentaho-solutions\bi-developers\etl 位置的 GetPDIEnvironment.xaction。图 4-4 展示了它。

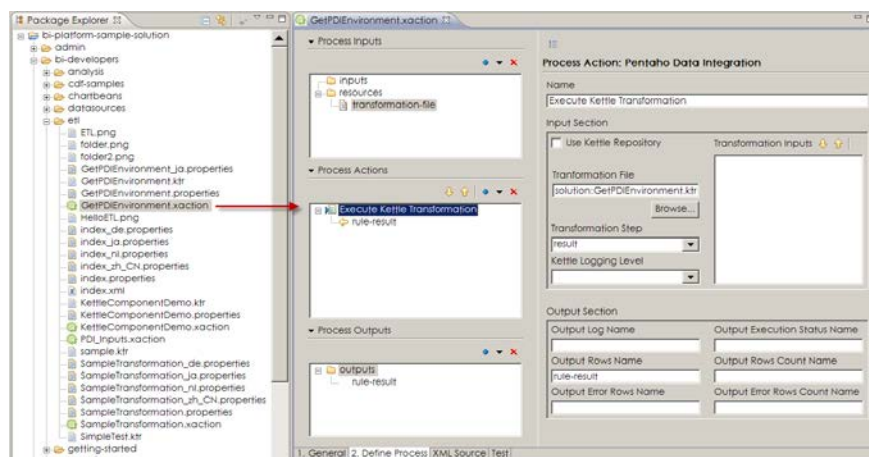


图 4-4 打开的 GetPDIEnvironment.xaction

借助 PDS，能够完成对 Action Sequence 的新建、修改、测试等操作。

4.2 深入到 Action Sequence 中

本节内容将深入围绕 Action Sequence 展开阐述。

4.2.1 Action Sequence 定义

Action Sequence 是以 XML 形式存在的一组动作（.xaction）。它主要由一般性设置、输入、资源、动作集合、输出等内容构成。所谓一般性设置，即图 4-5 所展示的 PDS 界面。这里主要是设置 Action Sequence 的概要信息，比如图标、描述等。

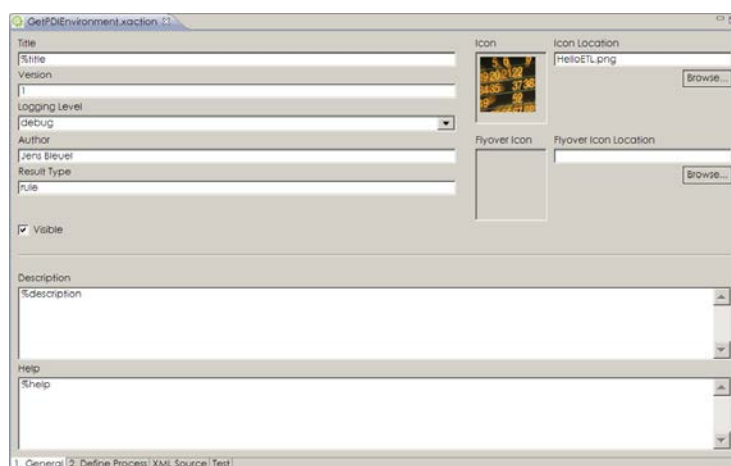


图 4-5 PDS 展示的一般性设置

图 4-6 展示了 Action Sequence 主体内容的定义。

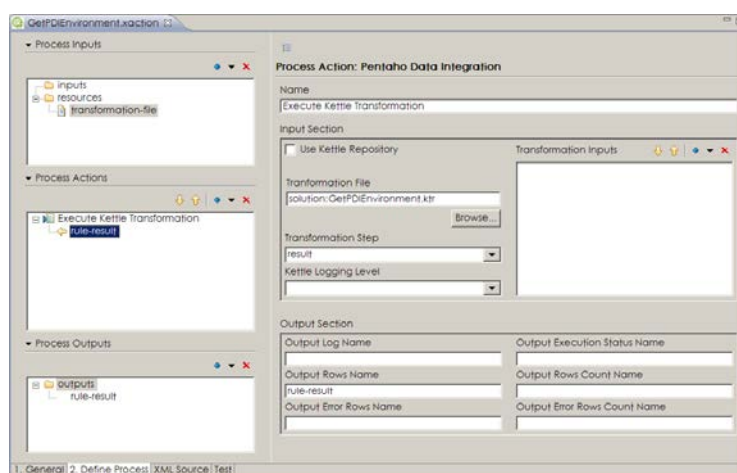


图 4-6 定义 Action Sequence 主体

不同 Action Sequence 的这一操作界面存在很大的差异性，因为各自引用的组件内容不同。比如，GetPDIEnvironment.xaction 便引用到 KettleComponent 组件，由于不同组件暴露的属性信息不一样，因此 PDS 提供的图形化界面也会不一样。

4.2.2 测试 Action Sequence

创建或修改.xaction 定义文件后，我们往往需要测试一下它的行为是否同预期的一致。然而，在测试 Action Sequence 之前，我们需要刷新一下 Pentaho BI 服务器的 Repository 缓存，具体见图 4-7。或者，通过管理控制台也能够达到这一目的。

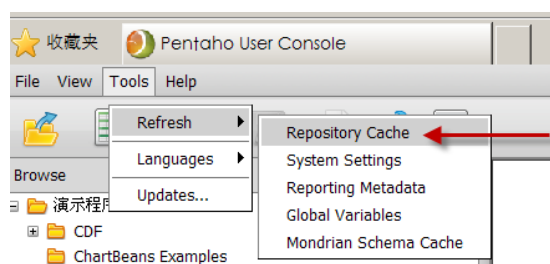


图 4-7 刷新 Repository 缓存

测试 Action Sequence 的途径很多，比如登录到用户控制台中、直接在 PDS 中。图 4-8 展示了 PDS 中是如何测试 Action Sequence 的。

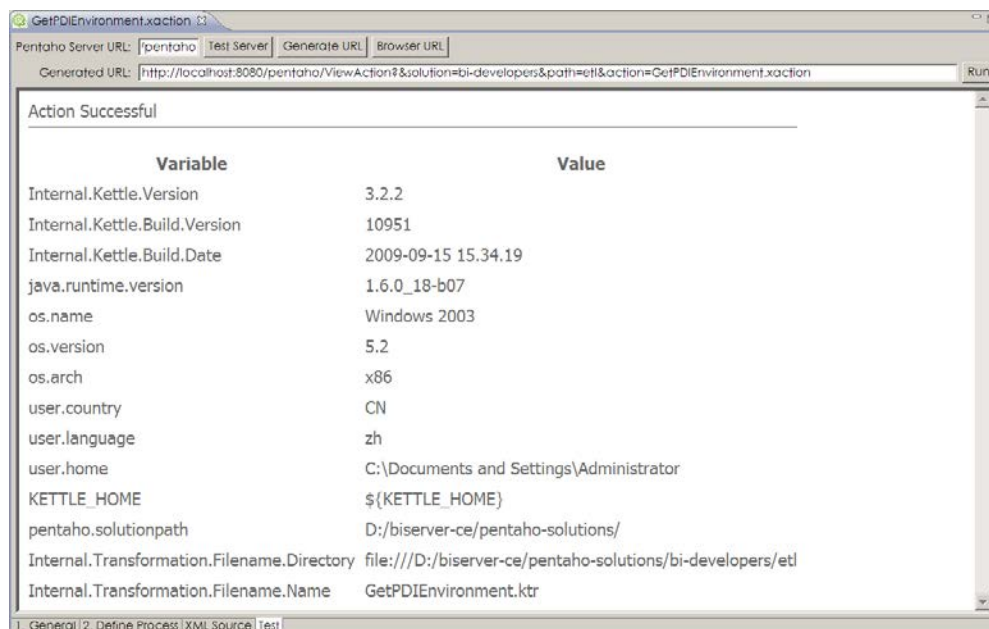


图 4-8 测试 Action Sequence

我们需要在“Pentaho Server URL”输入框中输入类似 <http://localhost:8080/pentaho> 的服

务器地址信息，然后通过单击“Generate URL”按钮便能够生成.xaction 对应的 URL，进而开始完成 Action Sequence 的测试工作。

4.2.3 组件集合

在 Action Sequence 定义中，<component-name/>元素引用的组件是最为基础的内容，比如 KettleComponent。客观地说，这些组件都存在对应的 Java 类。Pentaho BI 服务器于 org.pentaho.platform.engine.services.runtime(pentaho-bi-platform-engine-services-3.5.0.stable.jar) 包的 plugins.properties 属性文件中集结了其内置的组件集合，即它们可供 Action Sequence 复用，具体见下面的列表，以供查阅。

```
# Notes on this file:
# If the provided classname begins with an exclamation point (!),
# then that is an indicator to the plugin factory code to display
# a deprecation warning to the Pentaho console.
#
# Map the Short Names
ContentOutputComponent =
    org.pentaho.platform.plugin.action.deprecated.ContentOutputComponent
ContentRepositoryCleaner =
    org.pentaho.platform.plugin.action.builtin.ContentRepositoryCleaner
HelloWorldComponent =
    org.pentaho.platform.plugin.action.examples.HelloWorldComponent
ResultSetCompareComponent =
    org.pentaho.platform.plugin.action.datatransforms.ResultSetCompareComponent
ResultSetExportComponent =
    org.pentaho.platform.plugin.action.datatransforms.ResultSetExportComponent
ResultSetFlattenerComponent =
    org.pentaho.platform.plugin.action.datatransforms.ResultSetFlattenerComponent
ResultSetCrosstabComponent =
    org.pentaho.platform.plugin.action.datatransforms.ResultSetCrosstabComponent
SecureFilterComponent =
    org.pentaho.platform.plugin.action.builtin.SecureFilterComponent
SubActionComponent =
    org.pentaho.platform.plugin.action.builtin.SubActionComponent
TemplateComponent =
    org.pentaho.platform.plugin.action.builtin.TemplateComponent
BIRTReportComponent =
    org.pentaho.platform.plugin.action.eclipsebirt.BIRTReportComponent
EmailComponent =
    org.pentaho.platform.plugin.action.builtin.EmailComponent
JasperReportsComponent =
    org.pentaho.platform.plugin.action.jasperreports.JasperReportsComponent
JavascriptRule =
    org.pentaho.platform.plugin.action.javascript.JavascriptRule
ChartComponent =
```

```

    org.pentaho.platform.plugin.action.jfreechart.ChartComponent
ChartBeansComponent =
    org.pentaho.platform.plugin.action.chartbeans.ChartComponent
JFreeReportComponent =
    org.pentaho.platform.plugin.action.jfreereport.JFreeReportComponent
JFreeReportGeneratorComponent =
    org.pentaho.platform.plugin.action.jfreereport.JFreeReportGeneratorComponent
ReportWizardSpecComponent =
    org.pentaho.platform.plugin.action.jfreereport.ReportWizardSpecComponent
KettleComponent =
    org.pentaho.platform.plugin.action.kettle.KettleComponent
MDXDataComponent =
    org.pentaho.platform.plugin.action.mdx.MDXDataComponent
MDXLookupRule =
    org.pentaho.platform.plugin.action.mdx.MDXLookupRule
XMLALookupRule =
    org.pentaho.platform.plugin.action.xmla.XMLALookupRule
HQLLookupRule =
    org.pentaho.platform.plugin.action.hql.HQLLookupRule
TestComponent =
    org.pentaho.platform.plugin.action.builtin.TestComponent
UtilityComponent =
    org.pentaho.platform.plugin.action.deprecated.UtilityComponent
MondrianModelComponent =
    org.pentaho.platform.plugin.action.mondrian.MondrianModelComponent
PivotViewComponent =
    org.pentaho.platform.plugin.action.mondrian.PivotViewComponent
PrintComponent =
    org.pentaho.platform.plugin.action.builtin.PrintComponent
JobSchedulerComponent =
    org.pentaho.platform.scheduler.action.JobSchedulerComponent
SchedulerAdminComponent =
    org.pentaho.platform.scheduler.action.SchedulerAdminComponent
SQLDataComponent =
    org.pentaho.platform.plugin.action.sql.SQLDataComponent
SQLLookupRule =
    org.pentaho.platform.plugin.action.sql.SQLLookupRule
SQLExecute =
    org.pentaho.platform.plugin.action.sql.SQLExecute
XQueryLookupRule =
    org.pentaho.platform.plugin.action.xml.xquery.XQueryLookupRule
MQLRelationalDataComponent =
    org.pentaho.platform.plugin.action.pentahometadata.MQLRelationalDataComponent
OpenFlashChartComponent =
    org.pentaho.platform.plugin.action.openflashchart.OpenFlashChartComponent
PojoComponent =
    org.pentaho.platform.engine.services.solution.PojoComponent

# Map the v1.2+ Names
org.pentaho.plugin.core.ContentOutputComponent
= !org.pentaho.platform.plugin.action.deprecated.ContentOutputComponent
org.pentaho.plugin.core.ContentRepositoryCleaner
= !org.pentaho.platform.plugin.action.builtin.ContentRepositoryCleaner

```



```
org.pentaho.plugin.core.HelloWorldComponent
= !org.pentaho.platform.plugin.action.examples.HelloWorldComponent
org.pentaho.plugin.core.ResultSetCompareComponent
= !org.pentaho.platform.plugin.action.datatransforms.ResultSetCompareComponent
org.pentaho.plugin.core.ResultSetExportComponent
= !org.pentaho.platform.plugin.action.datatransforms.ResultSetExportComponent
org.pentaho.plugin.core.ResultSetFlattenerComponent
= !org.pentaho.platform.plugin.action.datatransforms.ResultSetFlattenerComponent
org.pentaho.plugin.core.ResultSetCrosstabComponent
= !org.pentaho.platform.plugin.action.datatransforms.ResultSetCrosstabComponent
org.pentaho.plugin.core.SecureFilterComponent
= !org.pentaho.platform.plugin.action.builtin.SecureFilterComponent
org.pentaho.plugin.core.SubActionComponent
= !org.pentaho.platform.plugin.action.builtin.SubActionComponent
org.pentaho.plugin.core.TemplateComponent
= !org.pentaho.platform.plugin.action.builtin.TemplateComponent
org.pentaho.plugin.eclipsebirt.BIRReportComponent
= !org.pentaho.platform.plugin.action.eclipsebirt.BIRReportComponent
org.pentaho.plugin.email.EmailComponent
= !org.pentaho.platform.plugin.action.builtin.EmailComponent
org.pentaho.plugin.jasperreports.JasperReportsComponent
= !org.pentaho.platform.plugin.action.jasperreports.JasperReportsComponent
org.pentaho.plugin.javascript.JavascriptRule
= !org.pentaho.platform.plugin.action.javascript.JavascriptRule
org.pentaho.plugin.jfreechart.ChartComponent
= !org.pentaho.platform.plugin.action.jfreechart.ChartComponent
org.pentaho.plugin.jfreereport.JFreeReportComponent
= !org.pentaho.platform.plugin.action.jfreereport.JFreeReportComponent
org.pentaho.plugin.jfreereport.JFreeReportGeneratorComponent
= !org.pentaho.platform.plugin.action.jfreereport.JFreeReportGeneratorComponent
org.pentaho.plugin.jfreereport.ReportWizardSpecComponent
= !org.pentaho.platform.plugin.action.jfreereport.ReportWizardSpecComponent
org.pentaho.plugin.kettle.KettleComponent
= !org.pentaho.platform.plugin.action.kettle.KettleComponent
org.pentaho.plugin.mdx.MDXDataComponent
= !org.pentaho.platform.plugin.action.mdx.MDXDataComponent
org.pentaho.plugin.mdx.MDXLookupRule
= !org.pentaho.platform.plugin.action.mdx.MDXLookupRule
org.pentaho.plugin.xmla.XMLALookupRule
= !org.pentaho.platform.plugin.action.xmla.XMLALookupRule
org.pentaho.plugin.hql.HQLLookupRule
= !org.pentaho.platform.plugin.action.hql.HQLLookupRule
org.pentaho.plugin.misc.TestComponent
= !org.pentaho.platform.plugin.action.builtin.TestComponent
org.pentaho.plugin.misc.UtilityComponent
= !org.pentaho.platform.plugin.action.deprecated.UtilityComponent
org.pentaho.plugin.olap.MondrianModelComponent
= !org.pentaho.platform.plugin.action.mondrian.MondrianModelComponent
org.pentaho.plugin.olap.PivotViewComponent
= !org.pentaho.platform.plugin.action.mondrian.PivotViewComponent
org.pentaho.plugin.print.PrintComponent
= !org.pentaho.platform.plugin.action.builtin.PrintComponent
org.pentaho.plugin.quartz.JobSchedulerComponent
```

```

= !org.pentaho.platform.plugin.action.quartz.JobSchedulerComponent
org.pentaho.plugin.quartz.SchedulerAdminComponent
= !org.pentaho.platform.plugin.action.quartz.SchedulerAdminComponent
org.pentaho.plugin.sql.SQLDataComponent
= !org.pentaho.platform.plugin.action.sql.SQLDataComponent
org.pentaho.plugin.sql.SQLLookupRule
= !org.pentaho.platform.plugin.action.sql.SQLLookupRule
org.pentaho.plugin.sql.SQLExecute
= !org.pentaho.platform.plugin.action.sql.SQLExecute
org.pentaho.plugin.xquery.XQueryLookupRule
= !org.pentaho.platform.plugin.action.xml.xquery.XQueryLookupRule
org.pentaho.plugin.mql.MQLRelationalDataComponent
= !org.pentaho.platform.plugin.action.pentahometadata.MQLRelationalDataComponent

# map the pre-v1.x names
org.pentaho.component.ContentOutputComponent
= !org.pentaho.platform.plugin.action.deprecated.ContentOutputComponent
org.pentaho.component.ContentRepositoryCleaner
= !org.pentaho.platform.plugin.action.builtin.ContentRepositoryCleaner
org.pentaho.component.HelloWorldComponent
= !org.pentaho.platform.plugin.action.examples.HelloWorldComponent
org.pentaho.component.ResultSetCompareComponent
= !org.pentaho.platform.plugin.action.datatransforms.ResultSetCompareComponent
org.pentaho.component.ResultSetExportComponent
= !org.pentaho.platform.plugin.action.datatransforms.ResultSetExportComponent
org.pentaho.component.ResultSetFlattenerComponent
= !org.pentaho.platform.plugin.action.datatransforms.ResultSetFlattenerComponent
org.pentaho.component.SecureFilterComponent
= !org.pentaho.platform.plugin.action.builtin.SecureFilterComponent
org.pentaho.component.SubActionComponent
= !org.pentaho.platform.plugin.action.builtin.SubActionComponent
org.pentaho.component.TemplateComponent
= !org.pentaho.platform.plugin.action.buildin.TemplateComponent
org.pentaho.birt.BIRTReportComponent
= !org.pentaho.platform.plugin.action.eclipsebirt.BIRTReportComponent
org.pentaho.component.EmailComponent
= !org.pentaho.platform.plugin.action.builtin.EmailComponent
org.pentaho.jasper.JasperReportsComponent
= !org.pentaho.platform.plugin.action.jasperreports.JasperReportsComponent
org.pentaho.component.JavascriptRule
= !org.pentaho.platform.plugin.action.javascript.JavascriptRule
org.pentaho.component.ChartComponent
= !org.pentaho.platform.plugin.action.jfreechart.ChartComponent
org.pentaho.jfree.JFreeReportComponent
= !org.pentaho.platform.plugin.action.jfreereport.JFreeReportComponent
org.pentaho.kettle.KettleComponent
= !org.pentaho.platform.plugin.action.kettle.KettleComponent
org.pentaho.component.MDXDataComponent
= !org.pentaho.platform.plugin.action.mdx.MDXDataComponent
org.pentaho.component.MDXLookupRule
= !org.pentaho.platform.plugin.action.mdx.MDXLookupRule
org.pentaho.component.TestComponent
= !org.pentaho.platform.plugin.action.builtin.TestComponent

```

```
org.pentaho.component.UtilityComponent
= !org.pentaho.platform.plugin.action.deprecated.UtilityComponent
org.pentaho.component.MondrianModelComponent
= !org.pentaho.platform.plugin.action.mondrian.MondrianModelComponent
org.pentaho.component.PivotViewComponent
= !org.pentaho.platform.plugin.action.mondrian.PivotViewComponent
org.pentaho.component.PrintComponent
= !org.pentaho.platform.plugin.action.builtin.PrintComponent
org.pentaho.component.JobSchedulerComponent
= !org.pentaho.platform.scheduler.action.JobSchedulerComponent
org.pentaho.component.SchedulerAdminComponent
= !org.pentaho.platform.scheduler.action.SchedulerAdminComponent
org.pentaho.component.SQLDataComponent
= !org.pentaho.platform.plugin.action.sql.SQLDataComponent
org.pentaho.component.SQLLookupRule
= !org.pentaho.platform.plugin.action.sql.SQLLookupRule
org.pentaho.component.XQueryLookupRule
= !org.pentaho.platform.plugin.action.xml.xquery.XQueryLookupRule
com.pentaho.component.JFreeReportGeneratorComponent
= !org.pentaho.platform.plugin.action.jfreereport.JFreeReportGeneratorComponent
```

至于组件 Java 类的位置，可以主要参考 `pentaho-bi-platform-plugin-actions-3.5.0.stable.jar` 包。它们都继承于共同的 `org.pentaho.platform.engine.services.solution.ComponentBase` 基类。如有兴趣，可以去研究一下组件源码。

Pentaho BI 服务器内置了大量的组件，这使得借助 Action Sequence 完成各种复杂 BI 工作成为可能。事实上，我们还可以提供自身的组件集合，以扩展 Pentaho BI 服务器。

4.3 于复杂BI场景中进行Action Sequence实战

本节内容将结合某银行 ETL 调度场景阐述 Action Sequence 实战。

4.3.1 银行ETL调度场景概述

4.3.2 Action Sequence的创建过程

4.3.3 运行并验证Action Sequence的执行

4.4 小结

Action Sequence 有效整合了 Pentaho BI 套件，借助它，我们能够完成各种复杂程度的 BI 工作。Pentaho BI 套件内置了大量的 Action Sequence 示例，这些示例展示了各种 BI 行为，非常值得读者去研究、分析。本书的许多场合都离不开 Action Sequence 的使用。

下章内容将进入到 Pentaho 报表工具的阐述中。

5 Pentaho报表工具—数据展现解决方案

报表工具，这是老生常谈的问题了。Pentaho 报表工具对报表开发提供了一流的支持。

5.1 Pentaho数据展现解决方案概述

Pentaho 针对数据展现提供了完整的解决方案。比如，借助 Pentaho Report Designer(PRD)，最终用户能够快速完成报表的制作。与此同时，我们可以以多种不同方式运行制作好的报表，比如通过 Pentaho BI 服务器，或者将它嵌入到其它企业应用中（即嵌入式 Pentaho 报表）。

事实上，我们也可以直接通过 Pentaho 用户控制台完成即席报表的制作，即摆脱 PRD。不过，制作即席报表需要有报表模型的帮助，在 Pentaho 元数据编辑器的帮助下，我们能够快速完成报表模型的梳理。

5.1.1 Pentaho元数据编辑器概述

现如今，对于大规模报表制作而言，报表模型是不可或缺的。Pentaho Metadata Editor，这是专门用来制作报表模型的工具，简称 PME。图 5-1 展示了 PME 的主界面。

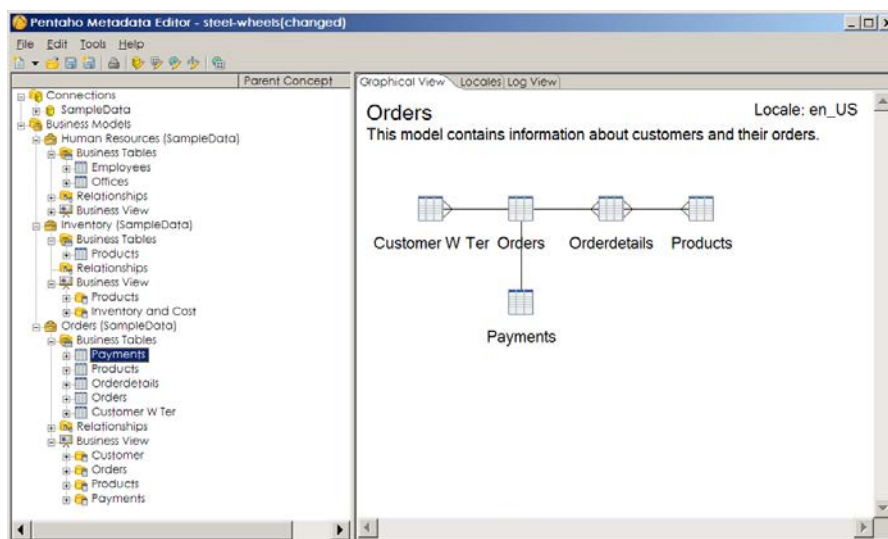


图 5-1 PME 主界面

借助报表模型，能够有效屏蔽底层的 RDBMS 信息，比如物理表、物理字段、物理视图等底层信息，甚至包括各种 SQL 语句、格式问题、币种问题、区域问题等。由于报表模型是面向业务人员，因此业务人员拿到报表模型后能够完成各种复杂程度报表的制作，这类角

色可以不了解 SQL、RDBMS 等底层数据库知识。

5.2 Pentaho Report Designer

本节内容将围绕 Pentaho Report Designer 展开阐述。

5.2.1 PRD 的下载及安装

开发者可通过 <http://sourceforge.net/projects/pentaho/files/> 网址下载到 PRD，具体见图 5-2。

▼ Report Designer	1.1 GB	2010-03-04	59,896	
▼ 3.6.0-stable	164.5 MB	2010-03-04	1,655	
prd-source-3.6.0-stable.zip	591.0 KB	2010-03-04	291	
prd-ce-mac-3.6.0-stable.tar.gz	54.6 MB	2010-03-04	167	
prd-ce-3.6.0-stable.zip	54.7 MB	2010-03-04	918	
prd-ce-3.6.0-stable.tar.gz	54.6 MB	2010-03-04	279	

图 5-2 下载 PRD

比如，我们将下载的 prd-ce-3.6.0-stable.zip 解压到 D:\report-designer 位置，并执行这一目录中的 report-designer.bat 批处理文件，PRD 的主界面将呈现在我们面前，具体见图 5-3。

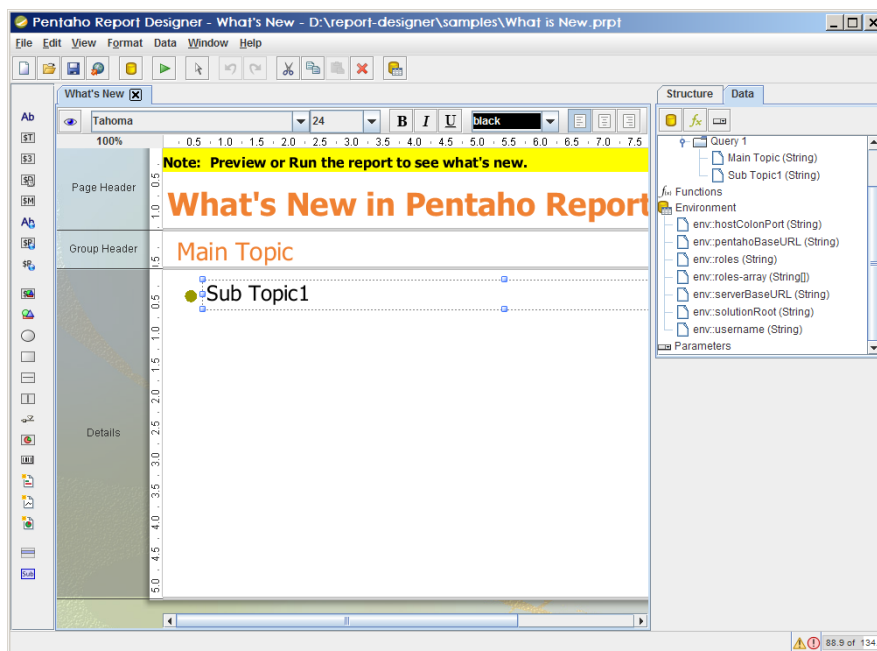


图 5-3 PRD 主界面

这样一来，即完成了 PRD 的下载和安装工作。

5.2.2 借助PRD完成报表的制作

5.3 借助PME梳理报表模型

PME 用于辅助梳理报表模型，其重要性不言而喻。

5.3.1 PME的下载及安装

开发者可通过 <http://sourceforge.net/projects/pentaho/files/> 网址下载到PME, 具体见图 5-4。










▼ Pentaho Metadata	587.1 MB	2010-03-18	32,516		
▼ 3.5.0-stable	141.3 MB	2010-03-18	6,814		
 pme-ce-mac-3.5.0-stable.zip	48.9 MB	2010-03-18	300		
 pme-ce-3.5.0-stable.tar.gz	45.7 MB	2009-10-15	1,543		
 pme-ce-3.5.0-stable.zip	45.7 MB	2009-10-15	3,865		
 pme-ce-3.5.0-stable-sources.zip	293.2 KB	2009-10-15	513		
 pme-ce-3.5.0-stable-javadoc.zip	725.5 KB	2009-10-15	593		

图 5-4 下载 PME

注意，我们这里将 pme-ce-3.5.0-stable.zip 解压到 D:\metadata-editor 位置，执行这一目录中的 metadata-editor.bat 批处理文件即可启用 PME。

5.3.2 使用PME

5.3.3 PRD中报表模型的使用

5.4 Pentaho即席报表

5.4.1 揭秘metadata.xmi

5.4.2 即席报表的制作

5.5 嵌入式Pentaho报表引擎

5.5.1 操作型BI报表

5.5.2 嵌入式报表的研发过程

5.6 Pentaho数据展现最佳实践

本节内容将围绕同 Pentaho 数据展现相关的若干主题展开探讨。

5.6.1 中文问题

借助 Pentaho Report Designer（嵌入式 Pentaho 报表引擎）制作报表期间，为预览 PDF 格式报表，用户往往要遭受中文问题的折磨，即中文显示不正常或者根本就显示不出来。

事实上，在 Pentaho 报表引擎内部，它是借助 iText 生成 PDF 报表的。

5.7 小结

Pentaho 报表工具功能强大，能够宿主在各种场景中，包括操作型 BI 领域。

下章内容将针对著名 OLAP 引擎—Mondrian 展开论述。

6 Mondrian OLAP引擎—多维数据分析利器

随着 BI 类项目的逐渐推广、使用，越来越多的企业需借助多维数据分析利器捕捉商业数据中的有用信息，从而抓住各种商机，并发现企业经营中存在的各种不足。不同于传统数据展现工作，多维数据分析工作能够多层次、多角度分析数据。本章内容将深入到著名的 OLAP 引擎，即 Mondrian 中，它被各种企业、BI 套件广泛使用着。

6.1 OLAP概述

6.1.1 多维建模及数据仓库设计

6.1.2 Mondrian OLAP引擎

6.2 使用Mondrian

6.2.1 下载Mondrian OLAP引擎

开发者需要去 <http://sourceforge.net/projects/mondrian/files/> 下载 Mondrian OLAP 引擎。图 6-1 展示了相应的下载信息。













▼ mondrian	2.5 GB	2010-02-25	202,236	 
▼ mondrian-3.1.6.13364	129.4 MB	2010-02-25	4,354	 
RELEASE.txt	10.8 KB	2010-02-25	429	
mondrian-3.1.6.13364.zip <small>Mondrian 3.1.6</small>	59.9 MB	2010-02-25	3,588	   
mondrian-3.1.6.13364-src.zip	5.4 MB	2010-02-25	181	 
mondrian-3.1.6.13364-derby.zip	64.1 MB	2010-02-25	156	 

图 6-1 下载 Mondrian

其中，mondrian-3.1.6.13364-src.zip 含有源码、文档等信息；mondrian-3.1.6.13364.zip 和 mondrian-3.1.6.13364-derby.zip 持有可供直接部署的 Mondrian 正式发布版，它们都包含了 mondrian-3.1.6.13364-src.zip，而且它们都内置了一 OLAP Cube 示例，前者采用微软 Access 文件存储相关数据，后者采用 Apache Derby (<http://db.apache.org/derby/>) 数据库存储相关数据。

6.2.2 初探Mondrian OLAP

6.2.3 Mondrian OLAP使用案例研究

6.3 借助PSW设计OLAP Cube

6.3.1 下载Pentaho Schema Workbench

开发者需要去 <http://sourceforge.net/projects/mondrian/files/> 下载 PSW 设计器。图 6-2 展示了相应的下载信息。






▼ schema workbench	167.5 MB	2010-02-26	39,252	 
▼ 3.1.6-stable	32.7 MB	2010-02-26	1,553	 
 psw-ce-3.1.6.13364.zip	16.4 MB	2010-02-26	1,232	
 psw-ce-3.1.6.13364.tar.gz	16.3 MB	2010-02-26	321	

图 6-2 下载 Pentaho Schema Workbench

下载后，我们可以将 psw-ce-3.1.6.13364.zip 解压到 D:\schema-workbench 位置，并执行其中的 workbench.bat 批处理文件，即可完成 PSW 的启动工作，见图 6-3。

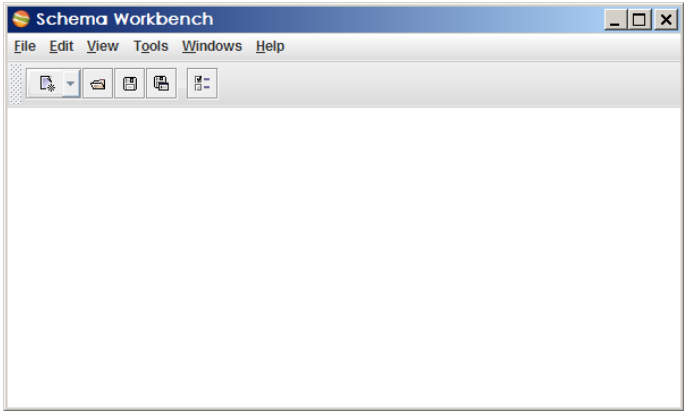


图 6-3 Pentaho Schema Workbench 主界面

下面将详细介绍 PSW 的各种功能，并借助它设计 OLAP Cube。

6.3.2 初探PSW

6.3.3 PSW使用案例研究

6.4 Mondrian技术架构探讨

6.5 与Pentaho BI服务器的集成

6.6 借助Pentaho Aggregation Designer提升数据分析性能

在实施多维数据分析期间，借助数据聚合技术能够大幅度提升数据分析的性能。如果数据集合技术被合理使用，则性能上的提升可能是成倍的，甚至可能是几十倍、几百倍。而借助 Pentaho 聚合设计器（Aggregation Designer，PAD），我们能够快速实施数据聚合。

6.6.1 数据聚合概述

6.6.2 PAD的下载和安装

同样地，<http://sourceforge.net/projects/mondrian/files/>网址内置了 PAD 的下载入口，具体见图 6-4。

▼ aggregation designer	188.7 MB	2009-05-08	20,492	 
▼ 1.1.1-stable	77.7 MB	2009-05-08	11,029	 
 pad-ce-1.1.1.zip	38.8 MB	2009-05-08	9,339	
 pad-ce-1.1.1.tar.gz	38.8 MB	2009-05-08	1,690	

图 6-4 下载 PAD

下载 pad-ce-1.1.1.zip 后，可以将它解压到 D:\aggregation-designer 位置，并执行这一目录中的 startaggregationdesigner.bat 批处理文件，即可成功启动 PAD，具体见图 6-5。

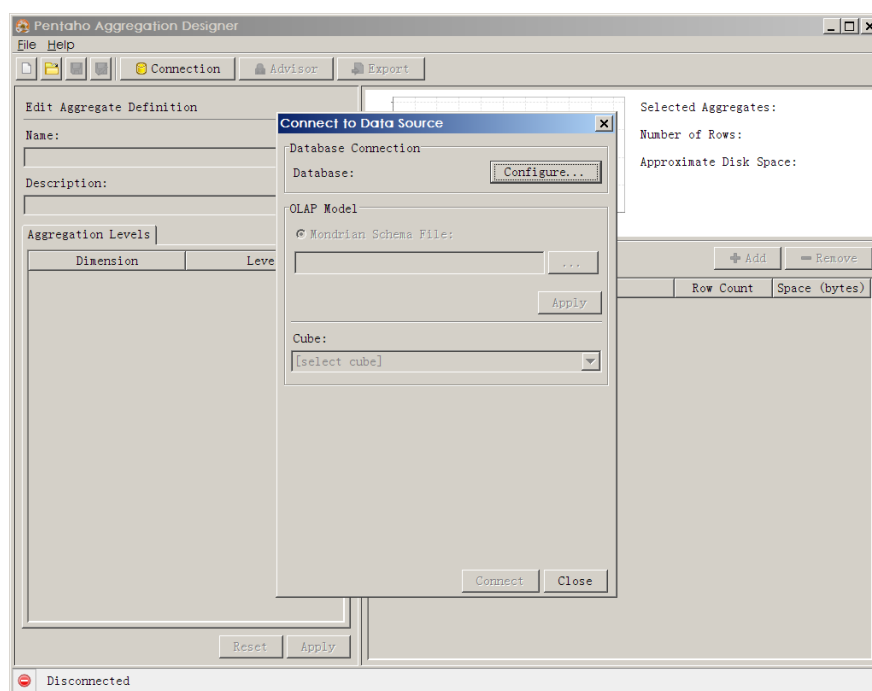


图 6-5 PAD 主界面

有关 PAD 的具体使用，下节内容将会详细介绍。

6.6.3 PAD使用案例研究

6.7 小结

Mondrian OLAP 引擎轻量、功能强大、性能好、部署灵活，等等，这些特性使得它得到了企业用户的广泛部署和关注。

下章内容将进入到基于 Weka 的数据挖掘解决方案中。

7 基于Weka的数据挖掘解决方案

从目前来看，数据挖掘是最为高级的数据分析行为之一。

7.1 数据挖掘概述

7.1.1 Weka介绍

Weka (<http://www.cs.waikato.ac.nz/~ml/weka/>)，是 Pentaho 数据挖掘解决方案的基石。

图 7-1 展示了 Weka 项目主页。事实上，<http://weka.pentaho.org/> 已经成了 Weka 的主战场，因为 Pentaho 团队接管了这一项目的后续研发。



图 7-1 Weka 项目主页

7.2 采纳Weka进行数据挖掘

7.2.1 下载Weka

开发者可在 <http://sourceforge.net/projects/weka/files/> 网址下载到 Weka 正式发布版，具体见图 7-2。






▼ weka-3-7	43.0 MB	2010-01-12	5,445		
▼ 3.7.1	23.4 MB	2010-01-12	2,128		
CHANGELOG-3-7-1	116.9 KB	2010-01-12	73		
weka-3-7-1.zip	others	23.3 MB	2010-01-12	2,055	

图 7-2 下载 Weka

下载 weka-3-7-1.zip 后，可以将它解压到 D:\weka-3-7-1 位置，并通过如下的命令行能够启动 Weka GUI Chooser。

```
D:\weka-3-7-1>java -jar weka.jar
```

图 7-3 展示了 Weka GUI Chooser 主界面。

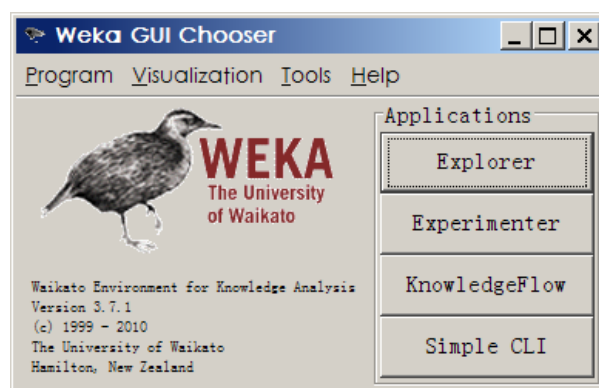


图 7-3 Weka GUI Choose 主界面

有关 Weka 的深入介绍，本章后续内容会一一阐述到。

7.2.2 Weka使用案例研究

7.3 小结

数据挖掘是一项非常高级的数据分析行为，Pentaho 数据挖掘解决方案使得这一行为成为了可能，能够提供 DM 解决方案的 BI 厂商不多。

下章内容将进入到 Pentaho 仪表盘工具的研究和探讨中。

8 Pentaho仪表盘工具

现如今，仪表盘（Dashboard）在企业应用中扮演着非常重要的作用，尤其是 BI 类项目。Dashboard 能够高效集成各种 BI 内容，并以较简单、统一的视图呈现给 BI 用户。而且，各种不同层次的 BI 用户能够定制适合自己的仪表盘。本章内容将仔细阐述 Pentaho 内置的仪表盘工具。

8.1 Pentaho Dashboard工具概述

Pentaho Dashboard 工具基于 CDF（Community Dashboard Framework）项目架构而成，而且默认时，CDF 内置在 Pentaho BI 服务器中。

8.1.1 Community Dashboard Framework介绍

CDF 宿主在 <http://code.google.com/p/pentaho-cdf/>位置，具体见图 8-1。

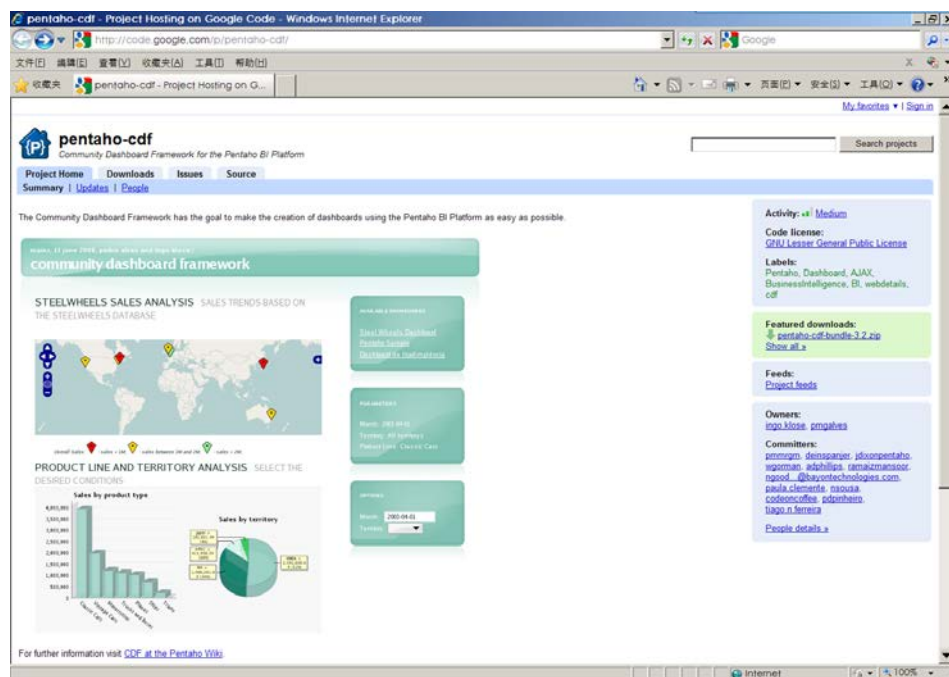


图 8-1 CDF 项目主页

8.1.2 借助Flash展现

8.2 小结

Pentaho 仪表盘工具是必不可少的 BI 利器。它能够快速而有效地整合各种 BI 内容，并以较简约的统一视图展现给最终用户。

下章内容将进入到 Pentaho BI 套件高级特性的讨论中。

9 Pentaho BI套件高级特性讨论

9.1 配置新的解决方案库

9.1.1 Solution概述

9.1.2 实践Solution

9.2 基于元数据的架构思路

9.3 基于领域模型的安全性管理

9.4 小结

Pentaho BI 套件涉及的知识点非常多，本章内容对一些主要的内容进行了非常深入的探讨和研究。

10 附录A：Kettle组件权威指南

为加快转换和作业的开发，Kettle 内置了大量实用组件，本附录将一一介绍它们的使用和相关技巧。

10.1 专注转换的组件集合

10.1.1 输入组件

10.1.2 输出组件

10.1.3 转换组件

10.1.4 实用（Utility）组件

10.1.5 流程控制（Flow）组件

10.1.6 脚本组件

10.1.7 查询组件

10.1.8 连接组件

10.1.9 数据仓库组件

10.1.10 校验（Validation）组件

10.1.11 统计（Statistics）组件

10.1.12 作业组件

10.1.13 映射组件

10.1.14 内联组件

10.1.15 批量装载（Bulk Loading）组件

10.2 专注作业的组件集合

10.2.1 通用组件

10.2.2 邮件组件

10.2.3 文件管理组件

10.2.4 条件组件

10.2.5 脚本组件

10.2.6 批量加载组件

10.2.7 XML 组件

10.2.8 文件传输组件

10.2.9 资源库组件

11 附录B: Spring Batch

11.1 为ETL而战

11.2 Spring Batch概述

11.3 实践Spring Batch

12 附录C：相关资料

12.1 图书

- 《Pentaho Solutions: Business Intelligence and Data Warehousing with Pentaho and MySQL》，WILEY。作者，Roland Bouman、Jos van Dongen。2009.8
- 《Pentaho Reporting 3.5 for Java Developers》，Packt Publishing Ltd。作者，Will Gorman。2009.9
- 《Pentaho 3.2 Data Integration: Beginner's Guide》，Packt Publishing Ltd。作者，María Carina Roldán。2010.4
- 《Mastering Data Warehouse Aggregates: Solutions for Star Schema Performance》，WILEY。作者，Christopher Adamson。2006.7

12.2 网站

- Pentaho 官方网站：<http://www.pentaho.com/>
- Pentaho 社区 Wiki：<http://wiki.pentaho.com/display/COM/Community+Wiki+Home>。这是重要的知识宝库，我们应该时常去这里研读、实践，以获得第一手的 Pentaho 知识。如有可能，您还可以贡献自身的 Pentaho 经验
- Pentaho 社区论坛：<http://forums.pentaho.org>。对 Pentaho 的任何疑问，您都可以透过它获得答案
- Matt Casters 的博客：<http://www.ibridge.be/>。他是 Kettle 项目创始人，现为 Pentaho Data Integration 产品主架构师。很有价值的地方
- Julian Hyde 的博客：<http://julianhyde.blogspot.com>。他是 Mondrian 项目创始人
- Jaspersoft 官方网站：<http://www.jaspersoft.com/>