

Pentaho 高级安装指南

原文版本	1.1.5
原文链接	
翻译	王娜、覃辉
校对	覃辉
Pentaho 中文讨论组 QQ 群: 12635055 论坛: http://www.bipub.org/bipub/index.asp	

Version 1.1 Milestone 5

December 12, 2005

Copyright © 2006 Pentaho Corporation. Redistribution permitted. All trademarks are the property of their respective owners.

For the latest information, please visit our web site at www.pentaho.org



目录

PENTAHO高级安装指南	- 1 -
翻译.....	- 1 -
王娜、覃辉.....	- 1 -
1 下载PENTAHO BI 平台.....	- 3 -
1.1 下载站点.....	- 3 -
2 预配置安装包.....	- 3 -
3 J2EE DEPLOYMENT DISTRIBUTION	- 3 -
3.1 额外要求.....	- 4 -
3.1.1 配置数据库.....	- 4 -
3.1.2 配置Sample Solutions	- 4 -
3.1.3 Deployment.....	- 5 -
3.1.4 包含Styles Web 应用程序.....	- 6 -
3.1.5 全力尝试它.....	- 6 -
4 应用程序服务器配置.....	- 6 -
4.1 JBOSS 4.0.3	- 6 -
4.1.1 Portal 特征和 Samples.....	- 7 -
4.2 TOMCAT 5.0 配置	- 7 -
4.2.1 Portal 特征和实例.....	- 10 -
4.3 TOMCAT 5.5 配置	- 10 -
4.3.1 Portal 特征和实例.....	- 11 -
4.4 改变服务器端口	- 11 -
4.4.1 编辑Pentaho Web 应用程序的配置文件.....	- 11 -
4.4.2 编辑你的应用程序服务器的配置.....	- 12 -
5 其他可用的包.....	- 12 -
5.1 源代码包	- 12 -
5.2 JAVA ARCHIVE (JAR) 包	- 12 -
5.3 SOLUTIONS 包	- 13 -
5.4 DATA包	- 13 -
5.5 STYLES WEB APPLICATION包.....	- 13 -
5.6 JAVADOC DOCUMENTATION包	- 13 -
6 从源代码构建.....	- 13 -

1 下载 Pentaho BI 平台

1.1 下载站点

Pentaho BI 平台可以自由下载：

- 使用一个web 浏览器，浏览到这个网页：<http://www.sourceforge.net/projects/pentaho>
- 点击 **Files**
- 选择满足你的需要的包。如果你不确定，以下章节会帮助你确定哪个下载适合你。
- 选择你想从哪里下载。如果你比较狂热，可从一些外国站点下载，例如 Russia 或 China。如果你住在 Russia 或 China，可从 North Carolina 的一些站点下载。你可能没时间到处玩，但是至少你的下载可环球旅行。

2 预配置安装包

预配置安装包是一个完整的服务器安装，设计来允许你评估 Pentaho BI 平台的一些特征。它包含一个预配置的应用程序服务器/web 服务器，预配置的预先生成的应用程序数据库，sample 数据和全功能的 samples。也有一个包含了 JRE (仅限于 Windows) 的预配置安装版本。

如果你选择了预配置安装包，你需要下载《[快速启动指南](#)》，也可从<http://www.sourceforge.net/projects/pentaho>下载，它将指导你进行简单的安装，配置和运行平台 samples。

3 J2EE Deployment Distribution

J2EE Deployment Distribution 是一个包，允许你为某种应用程序服务器构造多种不同的 web 应用程序 archives 和/或企业应用程序 archives。

到目前为止，我们有构造以下 archives (通过 Ant targets)的脚本：

- Tomcat 5.0 .war 文件
- **JBoss 4.0.3 .war 文件**
- **JBoss 4.0.3 .ear 文件**

可在 SourceForge 上找到这个包，其名为 **pentaho_j2ee_deployments-<version>.zip**。有关如何从 SourceForge 下载我们的包，请参考[下载 Pentaho BI 平台](#)一节。

如果你想将平台配置到以上列出的应用程序服务器中的一个，就下载这个包。这也是一个好的起点，如果你想将之配置到一个没有列出的应用程序服务器时。你需要的所有 Pentaho 文件均存在于这个包中，你所需要做的就是编写你自己的 Ant target，并添加你的环境必需的配置文件。

我们有一个预配置的 JBoss 应用程序服务器，其中已经为你设置和配置好了每件事物(参考之前的章节)。如果想尽快省力的配置运行，我们推荐使用预配置安装包。如果想在自己的环境

中设置平台，可选择使用 J2EE Deployment Distribution。

注意：我们主要的应用程序服务器测试环境是 **JBoss 4.0.3**。Pentaho community 和开发组欢迎各种反馈—不管是面临的问题，还是仅仅想让我们知道你已经成功配置了环境—关于在其它应用程序服务器上的配置。请使用<http://www.pentaho.org/discussion/installation.html> 上的论坛共享你的经历。

3.1 额外要求

除了 **pentaho_j2ee_deployments-<version>.zip**，你还需要下载以下的包：

- **pentaho_data-<version>.zip**：用于 samples 和测试实例的 Hypersonic 数据库。
- **pentaho_solutions-<version>.zip**：Samples 和测试实例包括所有必须的资源。

你也将需要确保你已经有：

- 这个下载需要已经安装了一个 Java SDK 或 JRE。Pentaho BI 平台已在 Java SDK 1.4 上构建和测试过，可用于很多平台 <http://java.sun.com/j2ee/1.4/download.html#sdk>。
- 确保环境变量 **JAVA_HOME** 指向 JDK 被安装的目录。
- 这个下载也需要安装了 Apache Ant，开放源码的基于 Java 的构建工具。你将需要 Ant 的 1.6.2 版本或更高。你可从 <http://ant.apache.org/> 下载 Ant。
- 确保 Ant 的 bin 目录在你的系统 path 上可用。

这一节描述外围内容的(数据和 sample solutions)手动配置，这是必需的，如果你想使用一个配置设置 sample solutions，而不是使用我们提供的预配置安装的话。

我们建议你在将平台配置到你的应用程序服务器之前，设置 solutions 和数据库。一些应用程序服务器需要在配置期间，数据库是 alive 的，否则配置可能失败。

3.1.1 配置数据库

- 解压 **pentaho_data-<version>.zip**。
- 在资源浏览树的根目录，你应能见到一个 startup 和 shutdown 脚本。
- 为启动数据库服务器，运行 start-hypersonic (.bat 用于 Windows, .sh 用于 *nix) 脚本。
- 为停止数据库服务器，运行 stop-hypersonic (.bat 用于 Windows, .sh 用于 *nix) 脚本。

3.1.2 配置 Sample Solutions

- **这很重要！**将 **pentaho_solutions-<version>.zip** 解压到一个目录，它是你的应用程序服务器的根目录的一个同级目录。
- 只要 **pentaho-solutions** 目录如上所述被配置为应用程序服务器的同级目录，web 应用程序不需要做任何配置上的改变就可以找到 **pentaho-solutions**。
- 如果应用程序不能找到 sample solutions，你可以输入一个配置参数指向他们。

- 抽取 web.xml 文件(从 .war)
- 查找值为 **solution-path** 的<param-name> 元素。
- 直接在<param-name>元素后，输入一个<param-value> 元素，将它的值设置为 **pentaho-solutions** 目录的绝对路径。
- 实例：<param-value>d:\pentaho\pentaho-solutions</param-value>
- 重新构建 .war 文件。

3.1.3 Deployment

为配置 Pentaho BI 平台，首先，你必须有一个 J2EE 兼容的 archive 文件，它包含平台的所有文件和资源。我们提供了 **pentaho_j2ee_deployments-<version>.zip** 帮助构建一些最常使用的 archives。

如果你正使用 JBoss 或 Tomcat 作为你的应用程序服务器，这个包将为你构建 archive。如果你正使用一个不同的应用程序服务器，你将需要编写你自己的 Ant 任务来构建你需要的 archive，并添加特定于你的应用程序服务器的配置文件。如果你愿意的话，我们很高兴将你的应用程序服务器的 Ant 任务放进我们的包中。参与 Pentaho 的细节信息请参考 <http://www.pentaho.org/contributions>。

pentaho_j2ee_deployments-<version>.zip 将构建两种不同的 archives - **.war 文件或.ear 文件**。**.ear 文件**有在选定的应用程序服务器上成功运行 Pentaho 平台 demo 和 samples 所必需的所有文件。**.war 文件**仅仅包含 Pentaho web 应用程序，在这种情况下，在你配置 Pentaho .war 文件后，还有更多要完成的工作。

1. 解压 **pentaho_j2ee_deployments-<version>.zip** 文件进一个工作目录。对于我们这里的实例，我们将使用 d:\work。

2. 确定你想构建哪个archive。你的选择有：

- 一个Tomcat 5.0 .war 文件
- 一个JBoss 4.0.3 .war 文件
- 一个JBoss 4.0.3 .ear 文件

3. 打开一个终端窗口(*nix)，或是一个命令行窗口(Windows)。

4. 切换目录到你解压配置包的工作目录。

5. 执行适合于你选中的 archive 的 Ant 命令。命令是：**ant <ant_target>**

这里<ant_target> 是以下值：

1. **build-tomcat5.0-war**
2. **build-jboss4.0.3-war**
3. **build-jboss4.0.3-ear**

因此，例如如果你想为 Windows 上的 Tomcat 构建一个 .war 文件，命令应如下：

```
D:\work> ant build-tomcat5.0-war
```

等待构建脚本执行完。可在 `/dist/<appserver>` 子目录下找到生成的 .ear 或 .war 文件，它在工作目录被创建。在我们的实例中，`<appserver>` 是 tomcat 目录。

既然你已经为你的应用程序服务器构建了合适的 archive，你可以配置平台。配置过程特定于你使用的应用程序服务器，我们假设你已经对之很熟悉了。

重要！ 确保查看了这篇文档中的应用程序服务器特定的提示，以防你的配置需要额外的步骤(如果有的话)。

记住！如果你在配置平台时遇到了任何的问题，将你的问题发布到我们的论坛：

<http://www.pentaho.org/discussion/installation.html>，很可能别人已经找到了一个解决方法，或可以帮助解决。也可发布你的成功案例，以及你怎么解决的！

3.1.4 包含 Styles Web 应用程序

如果你将平台配置为一个 .war 文件，你可能会漏掉另一个主要的组件—**pentahostyles. War**。Pentaho-styles.war 提供了平台 demo 和 samples 必需的所有 styles 和 images。不必担心这个，如果你 built 和 deployed 一个 .ear 文件—ear 文件包括了平台需要的所有额外 web 资源。

当没有配置 pentaho-styles.war 应用程序时，samples 也可以运行，但你将注意到找不到一些图标和 images，samples 的展示可能有点不正常。这是因为，为了简单的和可维护的 style 定制，我们已经将平台的 content styling 迁移进单独的 styles web 应用程序。

为纠正 samples 的展示，仅将 pentaho-styles.war 配置进和平台相同的应用程序服务器(这个 .war 打包在 **pentaho_styles_war-<version>.zip** 中)。

3.1.5 全力尝试它

那应该完成这个功能！查看下面的应用程序服务器提示，记得在启动应用程序服务器之前，启动数据库。samples URL 如下：<http://localhost:8080/pentaho/Navigate>。主机名依赖于你的应用程序服务器是安装于本地还是远程，当必需时进行相应的调整。

4 应用程序服务器配置

4.1 JBoss 4.0.3

如果你正配置 Pentaho BI 平台到一个 vanilla JBoss 4.0.3，你应按照如下提示。我们建议你从 **pentaho_j2ee_deployments-<version>.zip** 开始构建你的 .war 文件或 .ear 文件。

Pentaho web 应用程序中数据源的命名依赖于 JNDI，因此为了成功运行平台和 samples，你需要定义 Jboss 下的这些数据源。JBoss 4.0.3 中的标准是使用一个带有 **-ds.xml** 命名约定的 xml 文件。如果你将配置一个 .ear 文件，你不必定义这些 datasources，但是如果你将配置一

个.war，你可以使用我们在 **jboss_pentaho_war-<latest version>.zip** 包的根目录处为你提供的 datasource 定义：

- **PentahoHibernate-ds.xml**
- **quartz-ds.xml**
- **sampledata-ds.xml**
- **shark-ds.xml**

为了适当呈现平台用户接口，Pentaho 平台依赖于 jaxen 库的 version 1.1beta4 和 dom4j 库的 version 1.6.1。默认的，JBoss 4.0.3 自带了 dom4j 1.5.2，但并不包含 jaxen 库。

以下是操作步骤：

1. 停止服务器，如果它正运行。
2. 从 <http://jaxen.org/releases.html> 下载 jaxen 1.1 beta 4，或使用 **pentaho_j2ee_deployments-<version>.zip** 中的 jaxen.jar。
3. 从 <http://dom4j.org/download.html> 下载 dom4j 1.6.1，或使用 **pentaho_j2ee_deployments-<version>.zip** 中的 dom4j_1.6.1.jar。
4. 将 jaxen.jar 文件复制进 <jboss_home>/lib 目录。
5. 从 <jboss_home>/lib 目录删除 dom4j.jar 文件。
6. 将 dom4j_1.6.1.jar 重命名为 dom4j.jar，将新的 dom4j.jar 文件复制进 <jboss_home>/lib 目录。
7. 配置 .war 或 .ear。如果配置一个 .war，确保你将之命名为 pentaho.war。
8. 如果配置 pentaho.war 文件，将如上所示的 *-ds.xml 文件添加进 <jboss_home>/server/default/deploy 目录。如果你不使用默认的服务器，那么将文件复制进你的服务器的 deploy 目录。
9. 确保你的数据库正在运行。
10. 重启服务器。

4.1.1 Portal 特征和 Samples

JBoss 4.0.3 应用程序服务器没有自带 JBoss Portal，因此任何 Pentaho portal 特征或 samples 均不工作。随后将提供在一个预存在的 Jboss Portal 上设置 Pentaho 平台的指令。

这些关于 JBoss 的 pentaho.war 文件的提示仅应用于平台的 .war archive。你可以减少很多麻烦，如果你为 JBoss 构建 pentaho.ear，其为你合并了几个步骤，否则的话，它将是手动的。

4.2 Tomcat 5.0 配置

如果你正将 Pentaho BI 平台配置进 Tomcat 5.0，你需要使用以下提示。我们建议你从 **pentaho_j2ee_deployments-<version>.zip** 开始构建你的 .war 文件。

为了访问平台和 samples 的默认数据库，你必须在 Tomcat 5.0 的 **common/lib** 目录中添加 **hsqldb.jar** (Hypersonic 数据库 drivers)。

Pentaho web 应用程序数据源的命名依赖于 JNDI，因此为了成功运行平台和 samples，你需要为 Tomcat 定义这些数据源，Tomcat 5.0 的标准是使用 server.xml 文件里的 xml 条目。

以下是操作步骤：

1. 停止 Tomcat 服务器。
2. 将 hsqldb.jar 从 **pentaho-third-party** 目录复制进 **<tomcat-home>/common/lib** 目录，这里 **<tomcat-home>** 是你的 Tomcat 应用程序服务器的根目录。你可从 pentaho-third-party 目录找到 hsqldb.jar 文件，它是 'pentaho_j2ee_deployments-<version>.zip' 的部分。
3. 打开 **server.xml** 文件，位于 **<tomcat-home>/conf** 目录。
4. 小心的复制如下列出的 xml，在 server.xml 文件的 **<DefaultContext></DefaultContext>** 节点之间黏贴它们。如果你没有 **<DefaultContext></DefaultContext>** 节点，创建它们，并在其间黏贴新的xml。
5. 保存和关闭 server.xml 文件。
6. 启动你的数据库，如果它还没有启动的话。
7. 重启 Tomcat 服务器。

Tomcat 5.0的 Sample JNDI Datasource Mappings 如下：

```
<Resource name="jdbc/SampleData" auth="Container" type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/SampleData">
  <parameter><name>factory</name><value>org.apache.commons.dbcp.BasicDataSourceFactory</value></parameter>
  <parameter><name>maxActive</name><value>20</value></parameter>
  <parameter><name>maxIdle</name><value>5</value></parameter>
  <parameter><name>maxWait</name><value>10000</value></parameter>
  <parameter><name>username</name><value>pentaho_user</value></parameter>
  <parameter><name>password</name><value>password</value></parameter>
  <parameter><name>driverClassName</name><value>org.hsqldb.jdbcDriver</value></parameter>
  <parameter><name>url</name><value>jdbc:hsqldb:hsq://localhost/sampledata</value></parameter>
</ResourceParams>
```

```
<Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/Hibernate">
  <parameter><name>factory</name><value>org.apache.commons.dbcp.BasicDataSourceFactory</value></parameter>
  <parameter><name>maxActive</name><value>20</value></parameter>
  <parameter><name>maxIdle</name><value>5</value></parameter>
  <parameter><name>maxWait</name><value>10000</value></parameter>
  <parameter><name>username</name><value>hibuser</value></parameter>
  <parameter><name>password</name><value>password</value></parameter>
  <parameter><name>driverClassName</name><value>org.hsqldb.jdbcDriver</value></parameter>
  <parameter><name>url</name><value>jdbc:hsqldb:hsq://localhost/hibernate</value></parameter>
</ResourceParams>
```

```
<Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/Quartz">
  <parameter><name>factory</name><value>org.apache.commons.dbcp.BasicDataSourceFactory</value></parameter>
  <parameter><name>maxActive</name><value>20</value></parameter>
  <parameter><name>maxIdle</name><value>5</value></parameter>
  <parameter><name>maxWait</name><value>10000</value></parameter>
  <parameter><name>username</name><value>pentaho_user</value></parameter>
  <parameter><name>password</name><value>password</value></parameter>
  <parameter><name>driverClassName</name><value>org.hsqldb.jdbcDriver</value></parameter>
  <parameter><name>url</name><value>jdbc:hsqldb:hsq://localhost/quartz</value></parameter>
</ResourceParams>
```

```
<Resource name="jdbc/Shark" auth="Container" type="javax.sql.DataSource"/>
<ResourceParams name="jdbc/Shark">
  <parameter><name>factory</name><value>org.apache.commons.dbcp.BasicDataSourceFactory</value></parameter>
  <parameter><name>maxActive</name><value>20</value></parameter>
  <parameter><name>maxIdle</name><value>5</value></parameter>
```

```
<parameter><name>maxWait</name><value>10000</value></parameter>
<parameter><name>username</name><value>sa</value></parameter>
<parameter><name>password</name><value></value></parameter>
<parameter><name>driverClassName</name><value>org.hsqldb.jdbcDriver</value></parameter>
<parameter><name>url</name><value>jdbc:hsqldb:hsq://localhost/shark</value></parameter>
</ResourceParams>
```

4.2.1 Portal 特征和实例

Tomcat 5.0 没有自带 JSR168 兼容的 portlet 容器，因此任何 Pentaho portal 特征或 samples 均不起作用。我们希望很快可以解决在 Tomcat 中设置 portals 的问题，我们一定发布信息 ASAP。

4.3 Tomcat 5.5 配置

如果你正将 Pentaho BI 平台配置进 Tomcat 5.5，你需要使用以下提示。我们建议你从 **pentaho_j2ee_deployments-<version>.zip** 开始构建你的 .war 文件。

为了访问平台和 samples 的默认数据库，你必须在 Tomcat 5.5 的 **common/lib** 目录中添加 **hsqldb.jar** (Hypersonic 数据库 drivers)。

Pentaho web 应用程序数据源的命名依赖于 JNDI，因此为了成功运行平台和 samples，你需要为 Tomcat 定义这些数据源。我们发现将这些 resources 添加进嵌套在 <Host>元素中的 <Context>元素就可以了。下面我们提供了一些 sample xml。

以下是操作步骤：

1. 停止 Tomcat 服务器。
2. 将 hsqldb.jar 从 **pentaho-third-party** 目录复制进 <tomcat-home>/common/lib 目录，这里 <tomcat-home> 是你的 Tomcat 应用程序服务器的根目录。你可从 pentaho-third-party 目录找到 hsqldb.jar 文件，它是 ‘pentaho_j2ee_deployments-<version>.zip’ 的部分。
3. 打开 **server.xml** 文件，位于 <tomcat-home>\conf 目录。
4. 小心的将如下列出的 xml 添加进 server.xml 文件中的 <host></Host>元素。
5. 保存和关闭 server.xml 文件。
6. 启动你的数据库，如果它还没有启动的话。
7. 重启 Tomcat 服务器。

注意：在这个过程中，我们假设你已经将 pentaho.war 放进了 webapps 目录，且 Tomcat 也

被设置成自动配置新 webapps。

Tomcat 5.5的 Sample JNDI Datasource Mappings 如下:

```
<Context path="/pentaho" docbase="webapps/pentaho/">
  <Resource name="jdbc/SampleData" auth="Container"
type="javax.sql.DataSource" maxActive="20"
maxIdle="5" maxWait="10000" username="pentaho_user"
password="password"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
driverClassName="org.hsqldb.jdbcDriver"
url="jdbc:hsqldb:hsqldb://localhost/sampledatab" />
  <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="org.hsqldb.jdbcDriver"
url="jdbc:hsqldb:hsqldb://localhost/hibernate" />
  <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="org.hsqldb.jdbcDriver"
url="jdbc:hsqldb:hsqldb://localhost/quartz" />
  <Resource name="jdbc/Shark" auth="Container" type="javax.sql.DataSource"
factory="org.apache.commons.dbcp.BasicDataSourceFactory"
maxActive="20" maxIdle="5"
maxWait="10000" username="sa" password=""
driverClassName="org.hsqldb.jdbcDriver"
url="jdbc:hsqldb:hsqldb://localhost/shark" />
</Context>
```

4.3.1 Portal 特征和实例

Tomcat 5.5 没有自带 JSR168 兼容的 portlet 容器, 因此任何 Pentaho portal 特征或 samples 均不起作用。我们希望很快可以解决在 Tomcat 中设置 portals 的问题, 我们一定发布信息 ASAP。

4.4 改变服务器端口

默认的, 平台 web 应用程序被配置使用 8080 端口。如果你已经有另一个服务器使用了 8080 端口, 希望应用程序使用一个不同的端口的话, 按照以下步骤进行操作。

4.4.1 编辑 Pentaho Web 应用程序的配置文件

抽取 web.xml 文件(从.war, 位于pentaho.war/WEB-INF/web.xml)** 如果你使用预配置安装包, web.xml 位于pentahodemo/jboss/server/default/deploy/pentaho.war/WEB-INF/web.xml。

添加或编辑 **base-url** 上下文参数:

```
<context-param>
  <param-name>base-url</param-name>
  <param-value>http://server-name:port/pentaho</param-value>
</context-param>
```

将 **server-name** 改成用于访问服务器的 **name** 或 **IP** 地址。如果你仅仅在相同的机器上运行服务器和 web 浏览器，你可将之设置成 **localhost**。将 **port** 改成你想使用的端口号。

4.4.2 编辑你的应用程序服务器的配置

设置服务器使用的端口本身是一个应用程序服务器配置任务。每个应用程序服务器配置端口的的方式都是不同的。因为我们的参考实现运行于 Jboss，我们在这里包含了改变 Jboss 的端口的指令。对于其他的应用程序服务器，如何配置端口请参考你的服务器的文档。

对于 JBoss 4.0.3:

编辑 `<jboss-home>/server/default/deploy/jbossweb-tomcat55.sar/server.xml`。** 如果你使用预配置安装包，`server.xml` 位于

`pentahodemo/jboss/server/default/deploy/jbossweb-tomcat55.sar/server.xml`。

在文件顶部的几行中，你可以看到：

```
<!-- A HTTP/1.1 Connector on port 8080 -->
<Connector port="8080" address="{jboss.bind.address}"
```

改变文本 `port="8080"`，指向你想使用的端口号。如果你愿意的话，你也可以改变其他连接器的端口。

5 其他可用的包

5.1 源代码包

Pentaho 源代码包是一个 Pentaho 平台 Java 源代码和子系统配置文件的 archive。需要和源代码(到目前为止)一起分发的配置文件是 Hibernate 配置文件(`hibernate.cfg.xml`) 和 Quartz scheduler 配置文件(`quartz.properties`)。可在 SourceForge 上找到这个包，名为 **pentaho_src-<version>.zip**。有关如何从 SourceForge 下载我们的包的指令请参考[下载 Pentaho BI 平台一节](#)。

源代码包可以附加进二进制库，用于 debugging 和 tracing，或为了你的特定需要而修改和重编译。对于这个源代码包，我们不提供构建脚本或构建文档。如果你对源代码的构建脚本感兴趣，你应该从 SourceForge.net 下载 Pentaho SDK。

5.2 Java Archive (JAR) 包

Java Archive (JAR) distribution 是一个 Pentaho 平台二进制类和子系统配置文件的 archive。可在 SourceForge 上找到这个包，名为 **pentaho_jar-<version>.zip**。有关如何从 SourceForge 下载我们的包的指令请参考[下载 Pentaho BI 平台一节](#)。

平台 Java Archive (JAR) 包是有用的，如果你想在一个 standalone 应用程序中利用 platform API 编码，或想在你自己的 web 应用程序中包括这些类。

5.3 Solutions 包

solutions 包是可被用作你自己的 solutions 的起点的 sample 和 test solution 文档的一个 archive, 和/或可被放进你的平台的自定义配置中。可在 SourceForge 上找到这个包, 名为 **pentaho_solutions-<version>.zip**。有关如何从 SourceForge 下载我们的包的指令请参考[下载 Pentaho BI 平台一节](#)。

有关如何构建 Pentaho solutions 和理解 solution document schemas 的更多信息, 请参考 [Solution Building 指南](#), 它也可以从 SourceForge.net 上我们项目的站点上下载。

5.4 Data 包

data 包是几个 Hypersonic 数据库的 archive, 它包含了运行我们的预配置安装包和 solutions 的所有必需的应用程序和 sample 数据。可在 SourceForge 上找到这个包, 名为 **pentaho_data-<version>.zip**。有关如何从 SourceForge 下载我们的包的指令请参考[下载 Pentaho BI 平台一节](#)。

下载 data 包, 如果你想从源码构建平台, 或是想运行 sample solutions。

5.5 Styles Web Application 包

styles web application 包含格式化平台接口的展示的 images 和 cascading style sheets。我们将展示分割成它自己的 web 应用程序, 为了平台的简单和可维护的定制, 这样当你升级平台自身时, 它不会被覆盖。

如果你想配置平台 samples, 在从源码构建后, 或是需要一个地方启动你自己的定制, 那么你应该获取这个包。有关如何从 SourceForge 下载我们的包的指令请参考[下载 Pentaho BI 平台一节](#)。

5.6 Javadoc Documentation 包

Javadoc 文档包是 Pentaho 平台 Java 类的 Javadoc。可在 SourceForge 上找到这个包, 名为 **pentaho-javadoc-<version>.zip**。有关如何从 SourceForge 下载我们的包的指令请参考[下载 Pentaho BI 平台一节](#)。

6 从源代码构建

从头开始构建 Pentaho BI 平台的人, 不同的人有不同的原因。一些人想能够从头构建平台以更好的掌握体系结构, 如果愿意的话, 到处看看。一些想利用他们喜欢的 IDE 在代码内 debug 或 trace。另一些为了他们特定的目的, 需要将定制的增加或改变合并进代码。

为满足所有这些情况下的需求, 我们提供了 Pentaho SDK, 其为开发人员提供了一套可以放进一个 IDE 的文件, 并相对安逸的开始编译和 stepping Pentaho 源代码。你可从我们的 Sourceforge 站点下载 Pentaho SDK。关于如何从 SourceForge 下载我们的包的指令, 请参考[下](#)

载 Pentaho BI 平台一节。

如果(after painstakingly following instructions in the SDK guide) 你仍然没有获得你最终的目标文件，寻求帮助吧！帮助仅是一个论坛，位于

http://www.pentaho.org/index.php?option=com_simpleboard&Itemid=275&func=showcat&catid=14