

## 安装和配置 jBPM4

jBPM 需要安装？不是说它是一个框架（Framework）、一堆开放的源代码（Open Source），而非一套“应用程序（Application）”吗，据我所知一般只有应用程序才需要安装呀？

是的，在您深入了解 jBPM 后，可以把它的流程引擎看做一个 Java 工程——若干 Java 类、依赖库和配置文件；jBPM 的流程定义和运行时的上下文需要被存储在关系型数据库中——可以是基于调试目的的内存数据库 HSQLDB（[hsqldb.org](http://hsqldb.org)），也可以是真正的持久化数据库，诸如 MySQL（[www.mysql.com](http://www.mysql.com)），Oracle（[www.oracle.com](http://www.oracle.com)），PostgreSQL（[www.postgresql.org](http://www.postgresql.org)），Sybase（[www.sybase.com](http://www.sybase.com)）；jBPM 的流程设计一般需要一个基于客户端的图形化流程设计器软件，当然在 jBPM4 版本以后也可以在 Web 上做流程设计（这要归功于 Signavio 项目，[www.signavio.com](http://www.signavio.com)）；最终，基于 jBPM 开发出的企业流程应用一般会被部署在一台应用服务器（Application Server）上，以便服务于来自 Web 的访问、监控和管理等。

但是，以上所说的这一切，如果是一名初学者，能很好地掌控吗？

在早期的版本中，要使用 jBPM，很多准备工作都需要自己来做，而且以上每一个步骤和细节都需要自己去关注，例如安装数据库、建表、安装应用服务器、安装图形化流程设计器插件等，是不是会让人感觉到上来就碰了个大钉子？

幸运的是，在我们要介绍的 jBPM4.3 版本中，几乎这一切工作 jBPM 软件包的发布者都帮助您做好了！只要您使用过 JavaEE，Eclipse 和 Ant，就可以通过 Ant build 脚本“一键获取”所需要的 jBPM4 整套开发、运行和管理环境。这就是传说中的“安装”。

## 2.1 jBPM4 安装先决条件

首先，我们要获取 jBPM4 的软件包，可以在 SourceForge.net 上找到它：<http://sourceforge.net/projects/jbpm/>。在本书开始写作的时候，jBPM 的最新发布版本为

### 4.3。

作为一个“生机盎然”的开源项目，jBPM 的版本更迭比较快，约半年左右就会发布一个新的版本，而世事难料（在这里我指的是组织变更或商业并购），如果有一天你发现在上面所提到的 SourceForge.net 上找不到 jBPM 或其最新版本的时候，使用 Google 搜索“jBPM download”是一个比较保险的办法。

事实上，jBPM 在每一次大版本号变迁时的改动才是革命性的，例如 jBPM3 到 jBPM4；而 jBPM 小版本号的变迁则是相对有限的改变，几乎不会需要您重新学习什么，例如 jBPM4.1 到 jBPM4.3，改动不大，只要关注新版本的“*What's new*”即可，也就是说：掌握了 jBPM4.3，jBPM4.X 对您就不在话下啦！

把 jBPM4.3 (jbp4-4.3.zip，不区分操作系统) 下载下来之后，解压到硬盘上的任何一个目录中，这个目录就是您的 jBPM “工作目录”了。

**注意：**这个“工作目录”的绝对路径最好不要包含非英文字符，例如中文；最好不要含有空格。如果您不遵守这两条规则，也许您在 jBPM 开发过程中会遇到些莫名其妙的麻烦。

下面以 `${jbp4.home}` 指代这个“工作目录”。

这个目录中包含如下子目录及文件。

- `doc`：包括用户指南、Javadoc、Schemadoc 以及开发指南。
- `examples`：包括用户指南中使用到的示例流程。
- `install`：包括适用于不同环境的安装脚本。
- `lib`：包括 jBPM 依赖的第三方库和一些特定的归档包。
- `src`：全部 jBPM 源代码。
- `jbp4.jar`：jBPM 源代码归档包文件。
- `migration`：jBPM 升级功能解决方案包（这在本书第二篇的 11.2 流程定义转换工具中会提及）。

在开始使用 jBPM 前，还需要准备如下环境：

- 1) JDK（标准 Java 开发包）5 或更高版本。可以在 <http://java.sun.com/javase/downloads/> 获取到最新版本的 JDK。关于如何安装和设置 JDK 到您的操作系统请参考相关资料，本书不做说明。

- 2) 安装 jBPM 需要执行 Ant 脚本，所以需要 Apache Ant 1.7.0 或更高版本。可以在 <http://ant.apache.org/bindownload.cgi> 获取到最新版本的 Ant。

## 2.2 快速开始吧

如果作为一名初学者，拿到 jBPM 后，您最想做什么？没错，快速地安装好，开始运行。

下面的范例将以最简单的方式帮助您快速开始使用 jBPM。

提示：如果您有下载过 `apache-tomcat-6.0.20.zip` 或 `jboss-5.0.0.GA.zip` (Tomcat 还是 JBoss？视您想要运行 jBPM 的应用服务器类型而定，二者选其一即可)，可以把它放到 `${jbpn.home}/install/downloads` 目录下。这样可以避免安装脚本从网络上下载这些 jBPM 所需的软件。同理适用于 `eclipse-jee-galileo-win32.zip`，或在 Linux 平台下的 `eclipse-jee-galileo-linux-gtk(-x86_64).tar.gz`，或在 Mac OS X 平台下的 `eclipse-jee-galileo-macosx-carbon.tar.gz`。

按步骤来：

- 1) 打开命令控制台（即 Windows 下的 `cmd`），进入目录 `${jbpn.home}/install`。
- 2) 运行脚本（当然需要配置好 Ant 命令的路径）`ant demo.setup.tomcat` 或者 `ant demo.setup.jboss`。

没了，就这么简单的两步。

实际上这两步帮您做了如下工作：

- 1) 把 Tomcat 安装到 `${jbpn.home}/apache-tomcat-6.0.20` 目录下。
- 2) 把 jBPM 安装到 Tomcat 中。
- 3) 安装 HSQLDB，并在后台启动。
- 4) 创建数据库表结构。
- 5) 在后台启动 Tomcat。
- 6) 根据示例（来自 `examples` 目录）创建一个 `examples.bar` 业务流程归档，并把它发布到 jBPM 数据库中。
- 7) 从 `${jbpn.home}/install/src/demo/example.identities.sql` 初始化用户和组。

- 8) 安装 Eclipse 到 `${jbpn.home}/eclipse`。
- 9) 启动 Eclipse。
- 10) 安装 jBPM Web 控制台。
- 11) 安装 Signavio Web 设计器。

当这些都完成后, Tomcat (或 JBoss, 由您之前运行的 `demo.setup.*`脚本决定) 会在后台启动。

一旦 Eclipse 启动成功, 您可以在其上安装 GPD (图形化流程设计器), 使用这个基于 Eclipse 的客户端软件去进行流程建模, 如何安装请参考 2.9 安装图形化流程设计器 (GPD)。

或者您可以通过 Signavio web 设计器进行流程建模: <http://localhost:8080/jbpmeditor/p/explorer>。

这时候, 您也可以使用 jBPM 控制台: <http://localhost:8080/jbpm-console/>, 利用表 2-1 中所列用户之一进行登录。

表 2-1 jBPM 控制台用户

用户名	密码
alex	password
mike	password
peter	password
mary	password

注意: jBPM4.3 控制台目前存在一个问题——对于一些比较慢的机器, 在初始流程报表时, 控制台的失效时间太短了, 所以当您第一次请求流程报表时, 会出现超时, 控制台会崩溃。注销, 然后再次登录, 就可以避过这个问题。同时, 这个问题已经提交到了官方 JIRA – JBPM-2508。

## 2.3 安装脚本详解

在上一节中我们使用了 jBPM 安装脚本中名为 `demo.setup.tomcat` 的目标任务 (target), 如您所知, 这是一个 Ant build target, 接下来我们将完整解读 jBPM4.3 的安

装脚本。

jBPM 软件包目录中包含了一个 `install` 目录，此目录中有一个 Ant 的 `build.xml` 文件，您可以使用它来把 jBPM 按照需要安装到应用环境中。

**注意：最好严格按照这组安装脚本进行安装和发布 jBPM 配置文件。当然，我们可以根据需要进行自定义 jBPM 的安装和配置，但这需要您的经验和谨慎。**

如何调用安装脚本？打开命令行，进入 `${jbp.home}/install` 目录。使用 `ant-p` 命令您可以看到这个安装脚本里所有可以使用的目标任务。其参数都设置了默认值，以便快速执行，下面给出了可用目标任务的概况。

- `demo.setup.jboss`：依次执行——（下载）安装 JBoss，把 jBPM 安装到 JBoss 中，启动 JBoss，创建 jBPM 数据库表结构，部署示例，加载示例身份认证信息，（下载）安装并启动 Eclipse。
- `demo.setup.tomcat`：依次执行——（下载）安装 Tomcat，把 JBoss 安装到 Tomcat 中，启动 Tomcat，创建 jBPM 数据库表结构，部署示例，加载示例身份认证信息，（下载）安装并启动 Eclipse。
- `clean.cfg.dir`：清除配置——删除 `${jbp.home}/install/generated/cfg` 目录。
- `create.cfg`：基于当前的参数在 `${jbp.home}/install/generated/cfg` 目录中创建一组配置。
- `create.jbpm.schema`：在目标数据库中创建 jBPM 表结构。
- `create.user.webapp`：根据您当前的配置，在 `${jbp.home}/install/generated/user-webapp` 目录中创建一个最简的 jBPM JavaEE Web 应用程序。这个应用程序包括完整的 `WEB-INF` 目录，所有的 jBPM 依赖库都在其 `lib` 目录中。您可以将这个目标任务理解为创建一个“jBPM Blank Project”。
- `delete.jboss`：删除已安装的 JBoss 应用服务器。
- `delete.tomcat`：删除已安装的 Tomcat 应用服务器。
- `demo.teardown.jboss`：删除 jBPM 数据库中的表并停止 JBoss 服务。
- `demo.teardown.tomcat`：停止 Tomcat 服务，以及 HSQLDB 服务（如果 HSQLDB 服务正在运行的话）。
- `drop.jbpm.schema`：从数据库中删除 jBPM 的表结构。
- `get.eclipse`：下载 Eclipse，如果在 `${jbp.home}/install/downloads` 目录中不存在的话。

- `get.jboss` : 下载与当前版本 jBPM 匹配的 JBoss AS (JBoss Application Server 即 JBoss 应用服务器), 如果 `downloads` 目录中不存在的话。
- `get.tomcat` : 下载与当前版本 jBPM 匹配的 Tomcat, 如果 `downloads` 目录中不存在的话。
- `hsqldb.databasesmanager` : 启动 HSQLDB 数据库管理控制台。
- `install.eclipse` : 安装 Eclipse, 执行下载 (如果 `downloads` 目录中不存在的话) 解压操作。
- `install.jboss` : 安装 JBoss 应用服务器, 执行下载 (如果 `downloads` 目录中不存在的话) 解压操作。
- `install.jbpm.into.jboss` : 安装 jBPM 到 JBoss 中。
- `install.tomcat` : 安装 Tomcat 应用服务器到 `${tomcat.distro.dir}` (在 `build.xml` 中配置), 执行下载 (如果 `downloads` 目录中不存在的话) 解压操作。
- `install.jbpm.into.tomcat` : 把 jBPM 安装到 Tomcat 中。
- `install.examples.into.tomcat` : 部署所有的示例流程到 Tomcat。
- `install.signavio.into.jboss` : 把 Signavio 流程设计器应用安装到 JBoss。
- `install.signavio.into.tomcat` : 把 Signavio 流程设计器应用安装到 Tomcat 中。
- `load.example.identities` : 装载示例用户和用户组数据到数据库中。
- `reinstall.jboss` : 删除之前的 JBoss 安装, 并重新安装 JBoss。
- `reinstall.jboss.and.jbpm` : 删除之前的 JBoss 安装, 并重新安装 JBoss。然后把 jBPM 安装到 JBoss 中。
- `reinstall.tomcat` : 删除之前安装的 Tomcat, 并重新安装 Tomcat。
- `reinstall.tomcat.and.jbpm` : 删除之前安装的 Tomcat, 并重新安装 Tomcat。然后把 jBPM 安装到 Tomcat 中。
- `start.eclipse` : 启动默认位置上的 Eclipse IDE。
- `start.jboss` : 启动 JBoss 服务, 并等待 JBoss 启动完成, JBoss 服务作为后台进程运行。
- `start.tomcat` : 启动 Tomcat 服务, 并等待 Tomcat 启动完成, Tomcat 服务作为后台进程运行。
- `stop.jboss` : 通知 JBoss 服务停止, 但并不等待停止完成。
- `stop.tomcat` : 通知 Tomcat 服务停止, 但并不等待停止完成。
- `upgrade.jbpm.schema` : 更新数据库中的 jBPM 表结构到当前版本。

以下将从配置文件和依赖库这两个角度, 向您介绍一些使用安装脚本应该注意的事项。

## 2.3.1 关于配置文件

通过上述目标任务，我们了解到安装脚本能为我们下载安装 Eclipse, JBoss, Tomcat……简直无所不能，但细心的读者会发现：数据库软件的安装呢？

没错，脚本没有能力完成所有的事，如果要成功运行 jBPM，您必须要修改一些配置文件，例如，数据库对应的配置文件在目录 `${jbpn.home}/install/jdbc` 中，这个目录中列出了 jBPM 官方支持数据库类型的相应配置，根据您对数据库的选择配置相应的 `.properties` 文件吧，例如默认的 `mysql.properties` 内容如下：

```
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/jbpmdb
jdbc.username=jbpm
jdbc.password=jbpm
```

假设您的 MySQL 服务位于 192.168.0.8 的 3308 端口，库名为 `jbpmdb`，用户名为 `root`，密码为 `root`，则您需要修改配置为：

```
jdbc.driver=com.mysql.jdbc.Driver
# 数据库访问地址
jdbc.url=jdbc:mysql://192.168.0.8:3308/jbpmdb
# 数据库用户名和密码
jdbc.username=root
jdbc.password=root
```

下面的安装脚本参数也可根据需要自定义。

- `database`：默认值是 `HSQldb`。可选值为 `mysql`, `oracle` 和 `postgresql`。
- `jboss.version`：默认值是 `5.0.0.GA`。可选值是 `5.1.0.GA`。

如果想要自定义这些参数值，可以在运行 Ant 安装脚本时使用 `-D` 指令，例如：

```
ant -Ddatabase=postgresql demo.setup.tomcat
```

在 `install` 目录的 `build.xml` 文件中，您可以发现所有的安装参数：

```
<!-- DEVELOPER SPECIFIC CONFIGURATIONS -->
看这里，没错！您可以建立一个 build.properties 文件，在其中统一设定自定义的参数值
<property file="${user.home}/.jbpm4/build.properties" />
<!-- USER CUSTOMIZABLE PROPERTIES -->
就是在这里。database 参数可以指定安装的目标数据库类型，对应相应的 .properties 文件
<property name="database" value="hsqldb" /> <!-- {hsqldb | mysql | oracle
| postgresql} -->
<echo message="database..... ${database}" />
```

```

<property name="tx" value="standalone" /> <!-- {standalone | jta | spring}
-->
<echo message="tx..... ${tx}" />
<property name="mail.smtp.host" value="localhost" />
<echo message="mail.smtp.host... ${mail.smtp.host}" />
<!-- INTERNAL PROPERTY DEFAULTS -->
<property name="jbpm.version" value="4.3" />
<property name="jbpm.parent.dir" value="../../" />
<property name="jbpm.home" value="${jbpm.parent.dir}/jbpm-${jbpm.version}" />
<property name="hibernate.connection.type" value="jdbc" /> <!-- jdbc |
datasource -->
<property name="logging" value="jdk" /> <!-- jdk | none -->
<property name="cfg.dest.dir" value="${jbpm.home}/install/generated/cfg" />
<property name="install.src.dir" value="${jbpm.home}/install/src" />
<property name="mail.cfg" value="default" />
<property name="jdbc.properties.dir" value="${jbpm.home}/install/jdbc" />
<property file="${jdbc.properties.dir}/${database}.properties" />
<property name="examples.file" value="${jbpm.home}/examples/target/
examples.jar"/>
<property name="tomcat.version" value="6.0.20" />
.....

```

其中“INTERNAL PROPERTY DEFAULTS”部分设定了jBPM安装属性的一些默认值，您可以根据需要变更之，例如可以将最后一行的Tomcat版本改为“6.0.24”，当然，这些修改需要您的知识和经验作为保证。

## 2.3.2 关于依赖库

通过以上介绍，我们了解到jBPM提供了自动安装的Ant脚本。这些脚本会将正确的依赖库和正确的配置文件安装到正确的位置。但如果想在应用中定制属于自己的jBPM，那么不可避免地需要了解jBPM依赖库的细节和如何集成其他第三方库和框架……关于这方面的知识，您可以在第9章jBPM4扩展研发先决条件中了解到。

## 2.4 安装到JBoss

执行安装脚本中的install.jbpm.into.jboss目标任务会把jBPM安装到JBoss应用服务器中。这会把jBPM安装成为一个JBoss的服务，因此这台JBoss上所有应用程序都可以使用这个安装的jBPM流程引擎。



如同前文所说，您可以在运行脚本任务时通过如下参数来指定 JBoss 的安装路径：

```
-Djboss.home=PathToYourJBossInstallation
```

安装成功后，在 JBoss 环境中，应用程序可以通过 JNDI 查询服务获取 ProcessEngine（流程引擎）对象：

```
new InitialContext().lookup("java:/ProcessEngine")
```

同样，也可以通过 Configuration.getProcessEngine() 获取 ProcessEngine 对象。

## 2.5 安装到 Tomcat

安装脚本的 install.jbpm.into.tomcat 目标任务会把 jBPM 安装到 Tomcat 应用服务器中。这个任务对 Tomcat 主要做了如下“手脚”：

- 1) 复制了若干 jar 包文件到 \${tomcat.home}/lib 目录中，这包括 jBPM 本身的库及其依赖的第三方库，当然还有相应的数据库驱动程序包。
- 2) 分别将 Signavio 流程设计器、jBPM 控制台、GWT 控制台服务这 3 个应用程序安装到 \${tomcat.home}/webapps 目录，对应的 war 包文件是 jbpmeditor.war, jbpm-console.war, gwt-console-server.war。
- 3) 安装用于 jBPM 控制台流程分析报表的 BIRT（开源报表项目）模板及其依赖文件到 \${tomcat.home}/birt 目录。
- 4) 最后，在 \${tomcat.home}/conf/server.xml 文件中配置用于 jBPM 控制台用户身份认证的数据源。

```
.....
<Engine name="Catalina" defaultHost="localhost">
  <Host name="localhost" appBase="webapps" unpackWARs="true"
    autoDeploy="true" xmlValidation="false" xmlNamespaceAware="false">
    <!--
      Current workaround for using the JBPM identity tables on Tomcat.
      Setting the Realm on the context with path /jbpm-console did not seem
      to work. Note that this workaround means that the complete localhost
      domain will use this realm !
      这里就是 jBPM 控制台为了用户身份认证而在 Tomcat 上设置的数据源
    -->
    <Realm className="org.jbpm.integration.tomcat6.JbpmConsoleRealm"
      driverName="com.mysql.jdbc.Driver"
      connectionUrl="jdbc:mysql://localhost:3306/jbpmdb"
      connectionName="root"
```

```
connectionPassword="root" />
</Host>
</Engine>
.....
```

## 2.6 基于 Web 的 Signavio 流程设计器

从 jBPM3 开始，这个著名的开源项目就因为没有基于浏览器的图形化流程设计器而饱受用户诟病，没错，进入了 Web 时代，流程设计器没有理由总是停留在 CS (Client Server) 阶段。有很多公司认识到了这一点，因此市场上的 jBPM Web 流程设计器并不少见，但它们或不开放源代码、或基于特定的业务、或存在支持升级问题。总之，没有 RedHat JBoss jBPM 的官方认证和支持，您用着总会不放心，不是吗？

现在您不用担心了。从 Version 4.1 开始，jBPM 官方发布包绑定了一个完全开源的基于 Web 的 BPM 流程设计器，代号为 Signavio。

### 2.6.1 jBPM Web 流程设计器简介

Signavio Web 流程建模工具是和 JBoss jBPM 团队、德国的 Signavio 公司和 Hasso Plattner Institute (HPI 软件工程研究所) 紧密协作的成果。Signavio 项目基于 Web 建模工具 Oryx，Oryx 是由 HPI 主持的开源项目。HPI 和 Signavio 公司都会持续地在 Oryx 项目和 Signavio 项目中投入人员和资金的支持。关于这两个项目，您可以在 Google Code 上找到：<http://code.google.com/p/signavio-oryx-initiative/>。

图 2-1 所示便是 Signavio 流程设计器的使用界面，它能很好地支持 IE 和 Firefox 浏览器。

使用 Signavio 可以让业务流程分析人员通过浏览器建立业务流程模型。Signavio 输出的流程文件格式正是 jPDL。这意味着 Signavio 设计出的流程定义文件可以直接导入到 Eclipse GPD，反之亦然。流程定义文件会保存在硬盘上，位于 \$jbpm\_home/signavio-repository 中，这个参数在安装脚本中有默认值。

**注意：** Signavio 是基于 web 的业务流程建模工具，绑定在 jBPM 中，是 100% 开源的（基于 MIT 开放源代码许可证）。同时，Signavio 公司也提供商业版的同名工具，毫无疑问，商业版的 Signavio 有更多的功能。

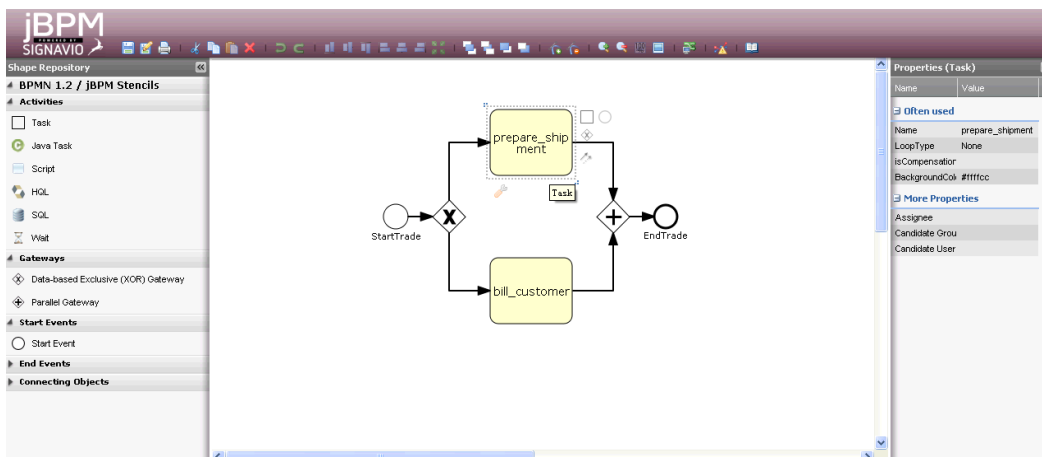


图 2-1 Signavio——基于 Web 的流程设计器

## 2.6.2 独立安装 Signavio

如本章开始所提及的，您可以使用 `$jbpn_home/install` 中的 `demo.setup.tomcat`（或 `jboss`）脚本任务把 Signavio 安装到您的 Web 容器（Tomcat 或 JBoss）中去。

这里还有几种把 Signavio 独立安装到 Web 容器中的方式：

- 使用 `$jbpn_home/install` 中的 `install.signavio.into.tomcat(or jboss)` 脚本任务，安装到默认的 Web 容器（Tomcat 或 JBoss）中。
- 复制 `$jbpn_home/install/src/signavio/jbpmeditor.war` 到您的目标 Web 容器中。

**注意：**不同的 Web 容器可能会存在些兼容性的问题，例如，Signavio 安装到默认的 Tomcat6 后，在流程设计中输入中文字符保存后，再打开，可能会造成无法加载的问题。

## 2.6.3 配置 Signavio

Signavio 配置很简单，大多数参数在 `web.xml` 中修改即可，您可以在 `jbpmeditor.war/WEB-INF/` 目录中找到。其中最重要的是 `fileSystemRootDirectory` 参数。这个参数的值必须为一个物理上存在的本地目录，它指定了流程定义文件（即

\*.jpd.xml 文件) 存储的位置, 以下是此参数的 web.xml 片段:

```
.....  
<context-param>  
  <description>Filesystem directory that is used to store models</description>  
  <param-name>fileSystemRootDirectory</param-name>  
  
<param-value>C:/opensource/jboss/jbpm-4.3/signavio-repository</param-value>  
</context-param>  
.....
```

如果使用\$jbpm\_home/install 中提供的安装脚本, fileSystemRootDirectory 会被默认地设置在\$jbpm\_home/signavio-repository 目录中。

## 2.7 用户自定义 jBPM Web 应用程序

如果您希望创建一个基于 jBPM 的 Web 应用程序供您自由发挥, 可以使用 create.user.webapp 这个脚本任务。正如 2.3.1 关于配置文件中说明的, 这将在\${jbpm.home}/install/generated/user-webapp 目录中创建一个最简的 jBPM JavaEE Web 应用程序……可以理解为创建一个“jBPM Blank Project”。

提示: 如果您在 JBoss AS 或其他包含 jta.jar 的应用服务器上部署了您的自定义 jBPM Web 应用程序, 需要把文件 \${jbpm.home}/install/generated/user-webapp/WEB-INF/lib/jta.jar 删除。

## 2.8 安装 jBPM 数据库

安装脚本包含了安装数据库的必要操作, 例如创建表, 删除表也是可选的。

使用任何数据库操作的前提条件是在\${jbpm.home}/install/jdbc 目录的.properties 文件中指定连接参数。

下面以 MySQL 数据库系统为例进行说明。

## 2.8.1 新数据库安装

在一张白纸上写字很容易！同样，安装一个全新的 jBPM 数据库也不难。

- 执行 `${jbpn.home}/install` 目录下的脚本任务 `create.jbpm.schema` 单独创建 jBPM 数据库表，当然，先要在 DBMS 上建立好“库”。作为创建表和约束的一部分，JBPM4\_PROPERTY 表会被初始化一些种子数据，例如当前的 jBPM 引擎版本（`key db.version`）和 ID 生成器版本（`key next.dbid`）。
- 执行 `drop.jbpm.schema` 脚本任务删除表结构。注意这个操作会删除 jBPM 数据库表中的所有数据。

## 2.8.2 升级旧的数据库

升级数据库，兼容旧版本会麻烦点。需要注意的是，本书使用的是 jBPM4.3 版本，升级只在 jBPM4 系列中进行。至于从 jBPM3 升级到 jBPM4 的数据库，最好根据您的业务单独做个 ETL（Extract, Transform, Load，数据的萃取转换装载，数据挖掘的前置工作）。

执行 `${jbpn.home}/install` 目录下的 `upgrade.jbpm.schema` 脚本任务开始升级。

升级包含两步操作：

- 1) 添加新版本中增加的表、列以及约束。
- 2) 插入种子数据。

jBPM4.3 版本前的一些数据库升级要点：

- 从 4.0 到 4.1 版本，表 JBPM4\_VARIABLE 添加了一个新列 `CLASSNAME_` 用来支持设置流程变量的值的自定义类型和 Hibernate 类型映射。这个列是可以为 `null` 的，因为这个功能在 4.0 中没有支持，所以支持没有初始值。
- 从 4.1 到 4.2 版本，jBPM 表现出了强大的兼容性潜力，这包括 JBPM4\_PROPERTY 表的横空出世。
  - 一个新的表 JBPM4\_PROPERTY 被用来保存流程引擎环境相关的数据。
  - jBPM 版本保存在 JBPM4\_PROPERTY 表中，`key db.version` 字段被用来在未来的发布版本中作为区分标识。
  - ID 生成策略是完全跨数据库系统的。下一个有效的 ID 是通过搜索所有包含主键的表计算出的，它被保存在 JBPM4\_PROPERTY 表中的 `key`

next.dbid 字段。

- 流程语言被设置为 `jpdl-4.0`，作用于所有已经存在的流程定义，对应 `JBPM4_DEPLOYPROP` 表中的 `key langid` 字段。jPDL 解析器根据 `langid` 属性来读取流程定义，以此支持向后兼容。

## 2.9 安装图形化流程设计器 ( GPD )

所谓图形化流程设计器，即 **Graph Process Designer**（以下简称 **GPD**），使用户能够通过图形拖曳、属性设置等可视化的方式进行业务流程设计，建立并展现业务流程模型。这个模型在 **jBPM4** 中一般为 `.jpdl.xml` 文件，遵循 **jPDL** 规范，此文件即“流程定义”文件，在运行时由工作流引擎解释执行，生成“流程实例”……

**jBPM** 的经典 **GPD** 一直使用 **Eclipse**（**Eclipse** 是著名的跨平台自由集成开发环境，即 **Integrated Development Environment, IDE**。您可以在 [www.eclipse.org](http://www.eclipse.org) 上了解关于 **Eclipse** 的更多信息）作为其客户端支撑平台，本节将介绍如何获取和安装 **Eclipse**，并将 **GPD** 插件安装到 **Eclipse IDE** 之上。

### 2.9.1 获取 Eclipse

使用 **jBPM4.3 GPD**，需要 **Eclipse IDE for Java EE Developers 3.5.0** 版本（**jBPM4.0** 依赖于 **Eclipse 3.4** 版本）。

可以选择之前介绍的脚本安装或手工下载安装这个 **Eclipse** 版本，大小约 163 MB。

注意，**Eclipse** 传统版本无法满足 **GPD** 的使用要求，因为它没有 **XML** 编辑器，但是 **Eclipse IDE for Java Developers 3.5.0** 可以使用 **GPD**。

### 2.9.2 在 Eclipse 中安装 GPD 插件

使用 **Eclipse** 软件升级（**Software Update**）功能安装 **GPD** 是非常简单的。文件 `install/src/gpd/jbpm-gpd-site.zip` 就是 **GPD** 的站点更新存档（**archived update site**）。

在 **Eclipse** 里添加 **GPD** 站点更新存档的步骤：

- 1) 选择 **Help**→**Install New Software** 命令。
- 2) 单击 **Add** 按钮。

- 3) 在 Add Site 对话框中, 单击 Archive 按钮。
- 4) 找到 install/src/gpd/jbpm-gpd-site.zip 文件, 并单击 Open 按钮。
- 5) 在 Add Site 对话框中单击 OK 按钮, 返回到 Install 对话框。
- 6) 选中出现的 jPDL 4 GPD Update Site 选项。
- 7) 单击 Next 按钮, 然后单击 Finish 按钮。
- 8) 接受软件许可证协议。
- 9) 询问时重启 Eclipse。

图 2-2 所示是主要操作步骤的示意图。

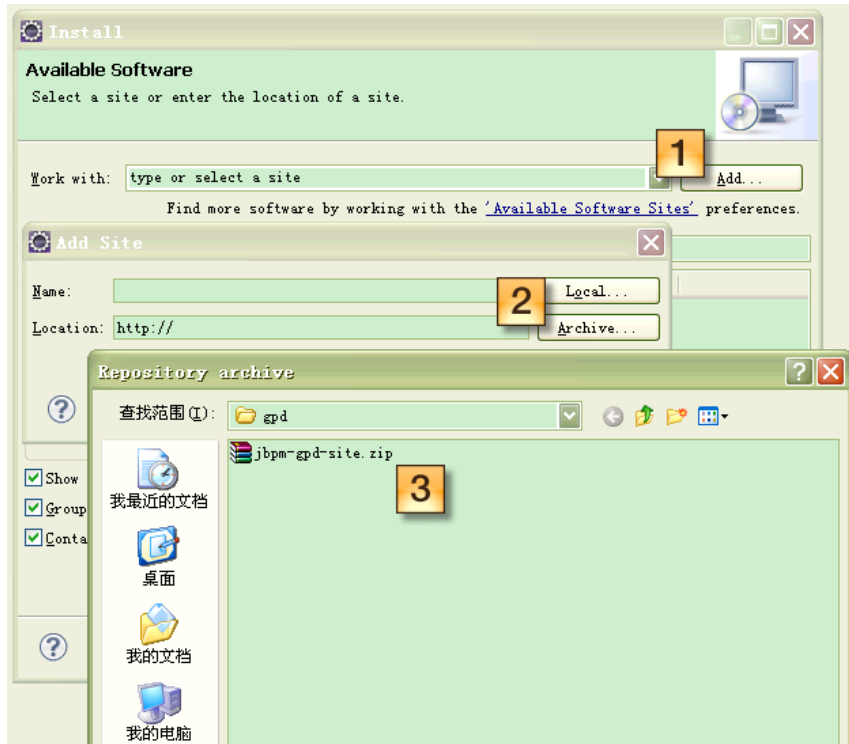


图 2-2 在 Eclipse 里添加 GPD 站点更新存档

### 2.9.3 配置 jBPM 运行环境

为了方便地基于 jBPM 开发流程应用, 需要在 Eclipse 中配置其运行环境, 步骤如下:

- 1) 选择 Window→Preferences 命令。
- 2) 选择 JBoss jBPM→jBPM 4→Runtime Locations 选项。
- 3) 单击 Add 按钮。
- 4) 在 Edit Location 对话框中输入一个名称，例如 jbpm43，然后单击 Search 按钮。
- 5) 在 Browse For Folder 对话框中，选择 jBPM 安装目录，然后单击 OK 按钮。
- 6) 在 Edit Location 对话框中单击 OK 按钮。

图 2-3 所示是主要操作步骤的示意图。

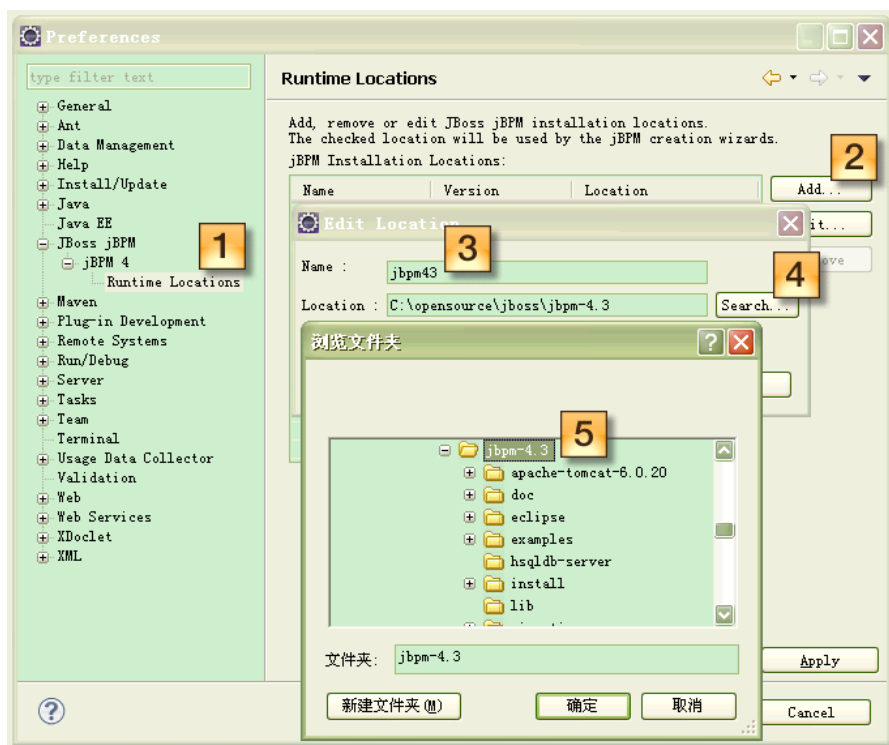


图 2-3 在 Eclipse 中配置 jBPM 运行环境

接下来为您的工作空间定义一个 jBPM 用户库（User Libraries）吧，它可以被用来引用 jBPM 的所有依赖库文件。如果您新建一个 jBPM 工程，只需将这个 jBPM 用户库添加到 build path 下即可。

- 1) 选择 Window→Preferences 命令。
- 2) 选择 Java→Build Path→User Libraries 选项。



- 3) 单击 **New** 按钮。
- 4) 输入名称 **jBPM Libraries**。
- 5) 单击 **Add JARs** 按钮。
- 6) 找到 **jBPM** 安装目录下的 **lib** 目录。
- 7) 选择 **lib** 目录下的所有 **jar** 文件，并单击 **Open** 按钮。
- 8) 选中刚才新建的 **jBPM Libraries**。
- 9) 重新单击 **Add JARs** 按钮。
- 10) 在 **jBPM** 的安装目录下选择 **jbpm.jar** 文件。
- 11) 单击 **Open** 按钮。
- 12) 在 **jbpm.jar** 下选中 **Source attachment**。
- 13) 单击 **Edit** 按钮。
- 14) 在 **Source Attachment Configuration** 对话框中，单击 **External Folder** 按钮。
- 15) 找到 **jBPM** 安装目录下的 **src** 目录。
- 16) 单击 **Choose** 按钮，为 **jbpm.jar** 关联源代码。
- 17) 单击两次 **OK** 按钮关闭所有对话框，完成。

图 2-4 所示是主要操作步骤的示意图。

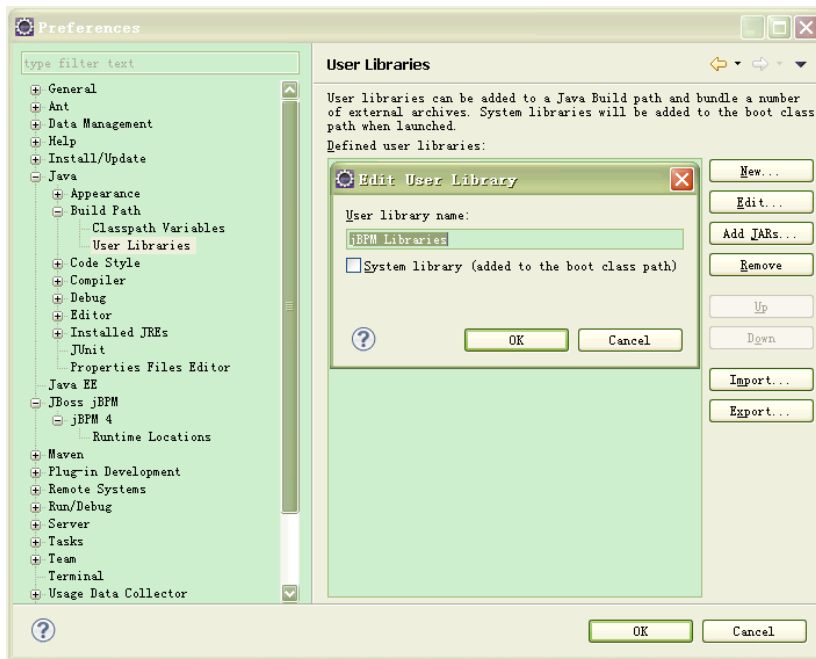


图 2-4 为工作空间定义 jBPM 用户库

## 2.9.4 添加 jPDL4 Schema 校验

正如之前我们所了解的，jPDL 是 jBPM 独有的流程定义语言，它以 XML 文件的形式描述业务流程。

在 jBPM3 即 jPDL3 中，要描述一个业务流程，系统需要产生两个文件：

- `gpd.xml`——这个文件记录业务流程的图形化表述，即一个单纯的图形坐标指示文件。
- `processdefinition.xml`——这才是真正的 jPDL 文件，描述业务流程的定义。

现在，在 jBPM4 即 jPDL4 体系设计中，已经将上述两个文件合二为一，即在 jPDL 中支持图形坐标位置的描述，开发者再也不用同时维护两个流程定义描述文件了，这是一个有效的简化。

由于 jBPM 官方提供的图形化流程设计器功能并不是特别全面，很多设计并不能全在图形界面下完成。因此，在很多情况下，我们需要直接编辑 jPDL 的 XML 源代码，所以，最好为 jPDL XML 指定 Schema，这样在编辑流程定义源代码的时候，可以通过编辑器快捷键“Alt+/'”快速呼出语法提示，同时这个 Schema 关联还可以帮助您校验出 jPDL 的语法错误，帮助您更为高效地编写流程定义 XML 源代码。

在 Eclipse 中配置此 Schema 的过程如下：

- 1) 选择 Window→Preferences 命令。
- 2) 选择 XML→XML Catalog 选项。
- 3) 单击 Add 按钮。
- 4) 将添加 XML Catalog Entry 的窗口打开。
- 5) 单击 File System 按钮。
- 6) 在打开的对话框中，选择 jBPM4 安装目录下 src 文件夹中 `jpdl.xsd` 文件。
- 7) 单击打开按钮。
- 8) 关闭所有的对话框，配置完成。

图 2-5 所示是主要操作步骤的示意图。

## 2.9.5 导入和使用范例

掌握一门新技术，最高效的手段就是在理解概念的基础上，自己亲自动手实验，

不是吗？

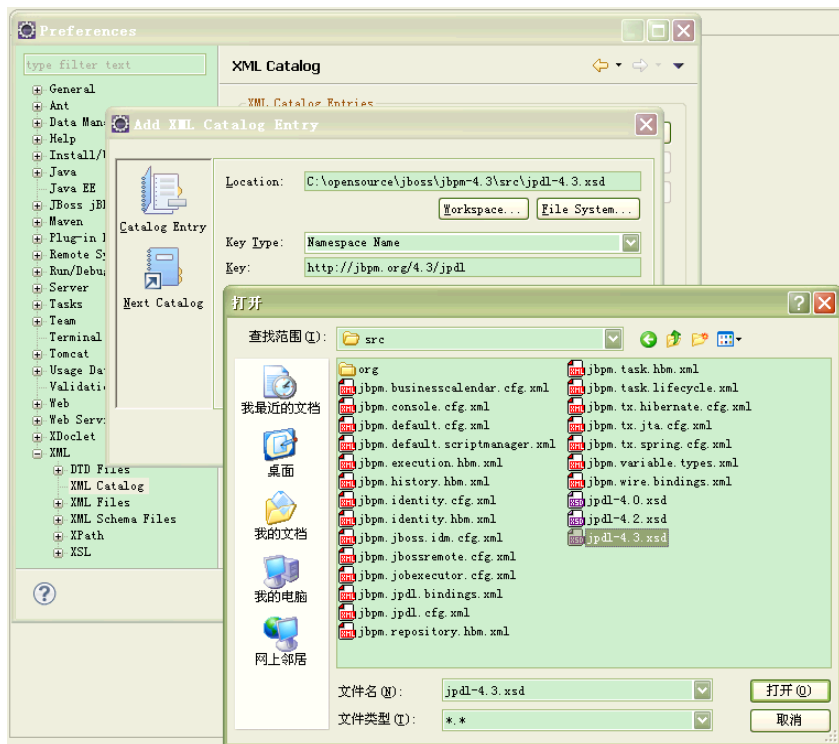


图 2-5 在 Eclipse IDE 中添加 jPDL4 Schema 校验

在 jBPM4 的软件包中，含有丰富的范例流程和测试代码，下面介绍如何将这些范例导入您的 Eclipse IDE，成为一个 examples 工程，供您学习和研究。

- 1) 选择 File→Import 命令。
- 2) 选择 General→Existing Projects into Workspace 选项。
- 3) 单击 Next 按钮。
- 4) 单击 Browse 按钮去选择 jBPM4 安装目录下的 examples 子目录。
- 5) 单击 OK 按钮。
- 6) 范例工程 examples 会被识别并准备导入。
- 7) 单击 Finish 按钮，完成。

在配置了 jBPM4 用户依赖库并且导入了范例工程后，范例中所有的单元测试类(继承自 org.jbpm.test.JbpmTestCase)都可以作为 JUnit Test 运行了，在测试类或方法上单击鼠标右键，选择 Run As→JUnit Test 命令即可。

设置完成了，还等什么呢？运行几个范例试试看吧！

提示：您可以使用 Eclipse + Ant 来处理范例流程的发布。首先，选择 Window → Show View → Other → Ant → Ant 命令，打开 Ant 视图；然后，将范例工程中的 Ant 构建文件 build.xml 从包视图拖曳到 Ant 视图，即可使用其中的 Ant 构建任务(target)来发布范例流程到目标服务器上。关于部署流程的细节，可参见第 4 章 把流程部署到服务器上去。

## 2.10 例程：jBPM HelloWorld

在本例程中，我们将设计一个最简单的流程定义——“HelloWorld”。

步骤如下：

- 1) 打开已经安装 GPD 的 Eclipse，新建一个“jBPM4 Process Definition”(jBPM4 流程定义文件)，命名为 process.jpdl.xml，如图 2-6 所示。

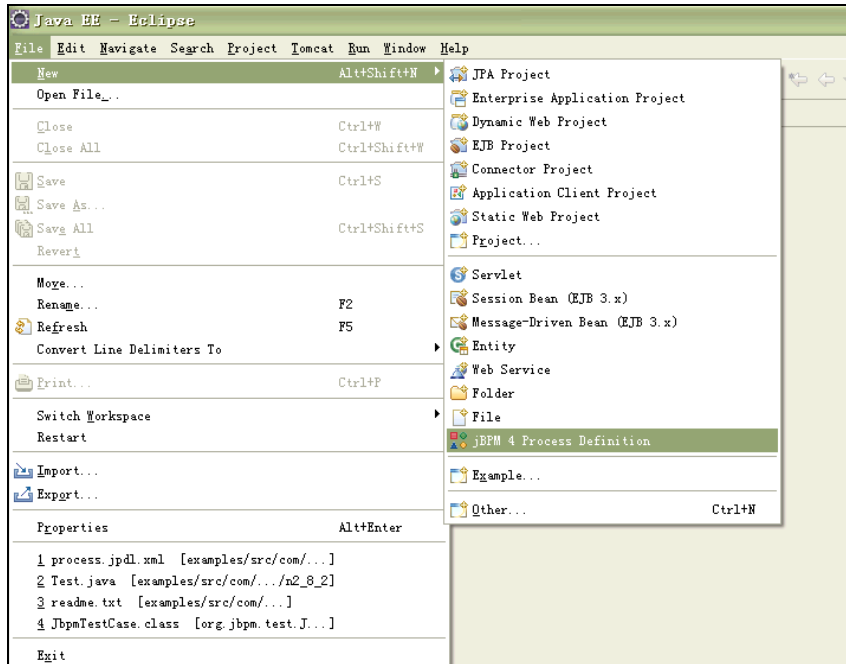


图 2-6 新建 jBPM4 流程定义文件

- 2) 进入流程定义设计界面，单击空白的流程图，在属性窗口中设置流程名称为“HelloWorld”。
- 3) 从左侧的组件工具栏（Components）中拖曳 start 活动、state 活动、end 活动至流程图。
- 4) 单击这 3 个活动，在其属性窗口中分别命名为“start”、“state”、“end”。
- 5) 从组件工具栏中选择 transition 转移线连接 start 活动至 state 活动、state 活动至 end 活动。完成设计后，如图 2-7 所示。

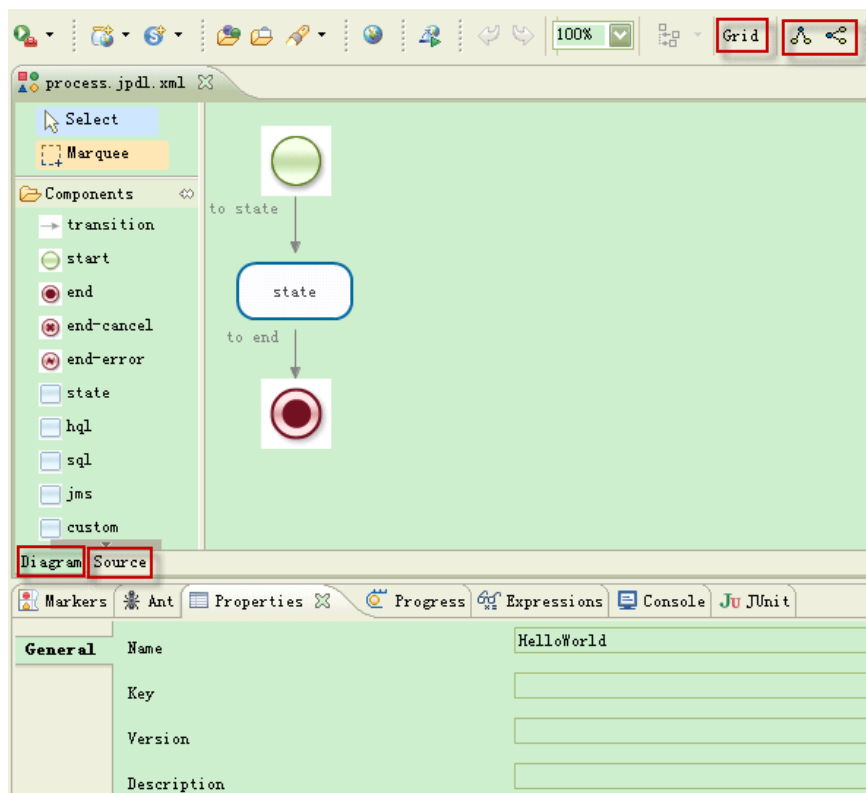


图 2-7 HelloWorld 流程定义设计示意图

- 6) 使用 GPD 顶部的 Grid 按钮为流程图显示网格线，用于对齐；还可以使用 、 两个按钮使流程图自动布局。

最终，单击 GPD 左下侧的“Diagram”页签，可以查看“HelloWorld”流程定义的图形化表达，如图 2-8 所示。

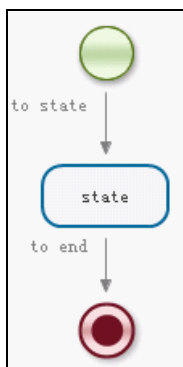


图 2-8 HelloWorld 流程定义图形

单击 GPD 左下侧的“Source”页签，可以查看流程图对应的 jPDL 源代码：

```
<process name="HelloWorld" xmlns="http://jbpm.org/4.3/jpdl">
  <start name="start">
    <transition name="to state" to="state"/>
  </start>
  <state name="state">
    <transition name="to end" to="end"/>
  </state>
  <end name="end"/>
</process>
```

## 2.11 小结

通过本章的内容，您学会了如何安装 jBPM4 的开发运行环境到目标操作系统，并且可以根据自己的安装需求做出一些个性化配置，最后通过一个简单的 Hello World 例程向世界宣布——jBPM，我来了！

下一章将介绍使用 jBPM 开发应用程序的第一个步骤——使用 jBPM 图形化流程设计器（GPD）来设计业务流程。