

Pentaho Design Studio 用户向导

原文版本	1.1.5
原文链接	
翻译	土豆 (QQ: 445817848)
校对	覃辉
Pentaho 中文讨论组 QQ 群: 12635055 论坛: http://www.bipub.org/bipub/index.asp	

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

介绍

本文档描述了如何使用Pentaho Design Studio建立面向实际问题的解决方案。本文档假定读者熟悉Pentaho BI平台以及Action Sequence的文档。我们建议在阅读本文前阅读《Creating Pentaho Solutions》。要尝试实例，读者需要能够访问一个Pentaho BI服务器。最简单的办法是本地计算机上安装设置好的Pentaho（Pre-Configured Install PCI）。请参考《Getting Started Guide》以获得使用PCI的信息。

Action Sequences 简介

Pentaho BI平台执行的操作取决于Action Sequence文件（XML格式）。Action Sequence定义以下的Action：数据库查询，报表生成，邮件发送以及各动作的顺序。数据可以在一个Action Sequences内部的Action之间传递，也可以在不同的Action Sequence之间传递。他们定义操作执行前需要有什么样的数据输入，以及将要依次执行什么样的操作，还要定义完成后产生怎样的输出给其他Action Sequences。《Creating Pentaho Solutions》深入阐述了Pentaho BI服务器的体系结构以及Action Sequence文件在其中的角色；它也提供了XML的格式参考以及对体系内其他模块的接口。

Design Studio

Pentaho Design Studio 是一个图形化的编写和测试 Action Sequence 文件的开发环境。它提供了组织解决方案的方法，并利用模版和向导协助自动完成一些常见的操作。尽管你仍然需要对底层的 XML 有足够的了解才能编写较复杂的解决方案，Design Studio 消除了你直接编写 XML 文件的困难，诸如保证每个标记正确的匹配，整理标记的格式，以及保证没有敲错 XML。当然，如果你喜欢直接编写 XML 文件，Design Studio 同样允许你显示和编辑原始的 Action Sequence 文件。

Design Studio 细节

Pentaho Design Studio 是整合在一个应用程序里的多个编辑器，试图和管理模块。其底层的应用框架式 Eclipse——www.eclipse.org。

Eclipse 是一个开放源代码的软件开发平台，目标是为开发集成工具提供一个健壮，特性完备，具商业产品质量的平台。它提供了建立和运行集成的软件开发环境所必需的模块和基础架构。Eclipse 平台的特性让开发者可以独立编写各种工具，不同人开发的工具之间可以结合的非常紧密以至于使用者无法区分各种工具。利用 Eclipse 给我们提供了一系列的优势：一个现有的广泛接受和良好定义的框架；在整合不同工具的同时保持一致的界面风格；重用现有的组件，以及大大家少了开发时间。

插件系统

Eclipse 有一个基于插件的体系架构，其几乎所有的功能，都来自于相互独立的模块。只要根据需求选择安装的插件，Eclipse 平台可以具有很多或很少的功能。Pentaho BI Design Studio 是作为独立的插件发布的。他们可以安装到用户已经在使用的 Eclipse 中。对于没有使用 Eclipse 的用户，或者那些需要独立的解决 BI 问题的“应用程序”的用户，解决方案是使用 Pentaho BI Design Studio。

Pentaho BI Design Studio 实际上是一个 Eclipse，安装了运行 Pentaho BI 需要的各种插件。它的界面有少许不同，为了有一个更简单的界面，删除了工具栏和菜单中的一些缺省项。它已经安装并配置了最新的 Pentaho 插件。

最新的插件呢？——OK，现在其实还只有一个插件，就是 Action Sequence 的编辑器，但是其他插件也快来了，我保证。

安装

有两种方法开始，取决于你是什么类型的用户。如果你现在在使用 Eclipse，安装了 3.1 版本或者更新的版本，并且希望继续在这个环境下工作，你应该下载 Pentaho 插件然后安装到 Eclipse 中。如果你是 Windows 用户，且没有使用 Eclipse，或者就是不像在你现在的开发环境中混合使用 Pentaho，你应选择 **Design Studio**。

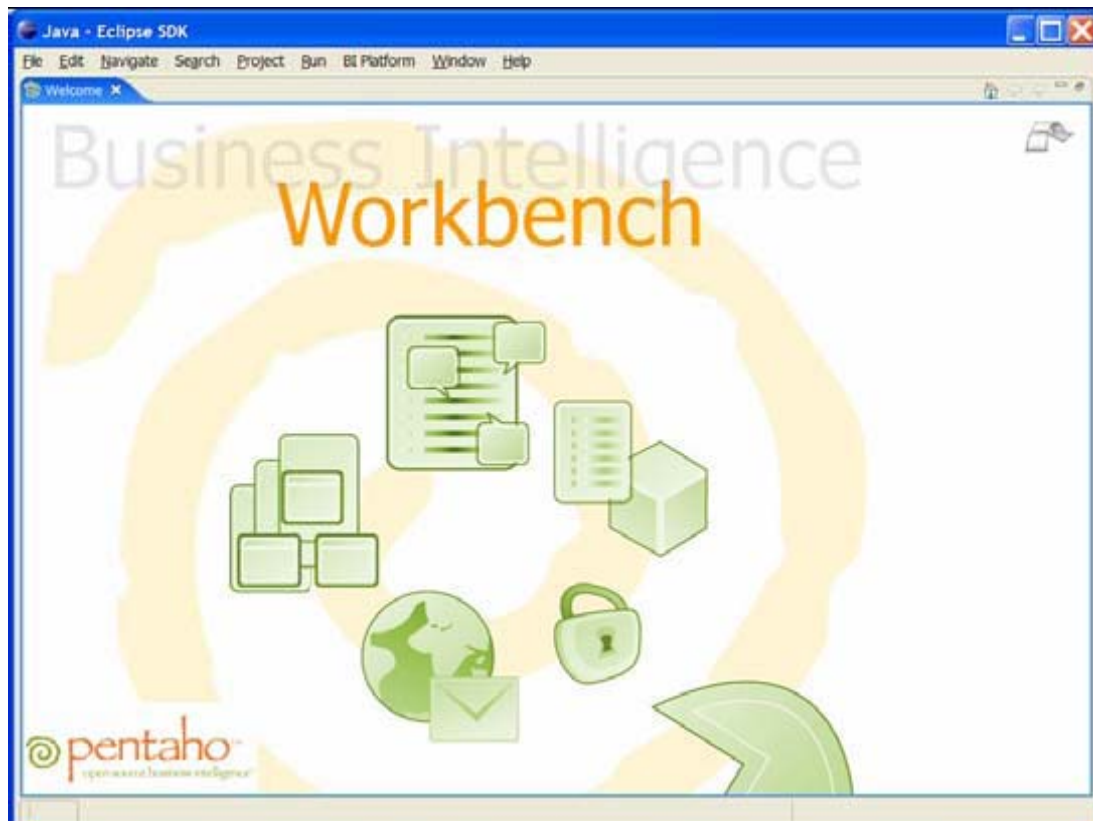
安装完整的 Design Studio

以下是一个包含 Pentaho Design Studio 的 Windows 版本的 zip 文件（Linux 的 RPM 安装文件马上会发布）

<http://www.pentaho.org/downloads/clients>

下载pentaho-design-studio_X.X.X.X.zip文件

解开此文件到你选择的路径，他会建立一个“PentahoDesign Studio”的目录并把文件解压到里面。不需要其他的安装步骤。要运行平台，执行PentahoDesign Studio.exe，你会看到欢迎页面：




安装插件

包含 Eclipse3.1 使用的 Pentaho Action Sequence 编辑器插件的 zip 包（适用各种操作系统）：

<http://www.pentaho.org/downloads/clients>

下载 [pentaho-action-editor-plugin_X.X.X.X.zip](#).

解压文件到你的 Eclipse 根目录，zip 包会把文件解压到正确的目录中。

重新启动 Eclipse，你会在 Eclipse 工具栏上看到 Action Sequence 的图标 

Design Studio 基础

如果你不熟悉 Eclipse，先读一下“Design Studio Users Guide”的“Getting Started”，在菜单里面选择“HELP/Help Contents”就可以看到。

用 Design Studio 创建和编辑内容需要编辑文件系统上的 XML 文档。Design studio 会分类组织这些内容，并提供懂得 XML 格式的编辑器，帮助你完成所有的步骤。在 design studio 里面，有多个层次的分类：workspace，perspectives，视图，编辑器，项目，文件夹和文件。

Workspace

Workspace 是存放你的工作内容的目录。它是项目，项目设置以及其他相关配置信息的容器。Design studio 开始的时候已经有一个建立好的 workspace——如果你需要，完全可以改变它和它存放的位置。Workspace 的位置确定后，会显示一个 Design studio 窗口。一个 workspace 窗口提供一个或多个 perspective。一个 perspective 包含编辑器，视图（如 Navigator 资源浏

览器)等。可以同时打开多个 design studio 窗口。Workspace 可以帮助你组织相关的项目。你会发现把所有有关报表和 BI 的项目组织进一个 BI 专用的 workspace 是很有用的。

Perspective

Perspective 是 Design studio 窗口内的一组视图和编辑器。一个 Design studio 窗口可以有一个或者多个 Perspective。每个 Perspective 包含一个或者多个视图和编辑器。在一个窗口内部，每个 Perspective 可以有不同的视图，但是所有的 Perspective 有相同的编辑器。你可以把 Perspective 理解在 workspace 里可视化地组织窗口的方法。

视图

视图是 Design studio 里面的可视化组件。视图通常用来浏览信息的目录(比如 Design studio 内的资源)，打开一个编辑器，或者显示一个活动编辑器的属性。视图的修改会马上被保存。在 Design studio 窗口里面，通常每个类型的视图只能有一个实例存在。“Navigator”是一个视图。它允许你访问在项目里面的文件夹和文件。

编辑器

编辑器是 Design studio 里面的可视化组件。编辑器通常用来编辑或者显示某种资源。编辑器里面的修改遵循“打开-保存-关闭”的生命周期。在 Design studio 窗口内部，一个编辑器类型可以有多个实例存在。Action Sequence 编辑器是一个典型的编辑器。

项目，文件夹，文件

在 Design studio 内部有三种类型的资源：项目，文件夹和文件。项目是 Design studio 用到的最大的结构单元。项目包含文件夹和文件，可以被打开，关闭或者创建 (build)。文件夹包含其它的文件或者文件。一个 Pentaho BI 解决方案通常是作为一个项目的形式存在。文件夹和目录是解决方案里面的具体内容。

刚开始的时候，第一个 Design studio 窗口打开，会显示 Pentaho perspective，里面只看得到一个 Welcome 视图。点击视图里面名为“Design studio”的箭头，会显示其它的视图。

(任何时候都可以选择 Help) Welcome 以显示 Welcome 视图)

一个快捷键工具栏会显示在窗口的右上角。它允许你打开心 perspective 或在已经打开的 perspective 之间切换。当前 perspective 的名字会显示在窗口的标题上，它所关联的工具会在工具栏里面高亮显示。

开始使用

首先。。。要声明一点。Pentaho Design Studio 实际上是一个特殊的修改了的 Eclipse Design studio，所以“Design studio”可能用来指任何一个。“Action Sequence Editor”指 Action Sequence 插件。不管你使用预先配置好的 Pentaho BI Design Studio 还是在你自己的 Eclipse 上安装 Action Sequence 插件，都是一样的。

pentaho 实例

本文档里面提到的例子需要你访问 Pentaho BI 服务器，并且 Pentaho 例子要么在本机上，要么在网络共享的文件夹中。如果没有 Pentaho BI 服务器，最简单的方法就是安装预先配置好的安装文件 (PCI)。PCI 是一个单独的 Pentaho BI 服务器。只要解开压缩包就可以运行。它已经包含了例子和用例子数据生成的配置好数据库。本文的余下部分会将设你已经成功地在浏览起里面运行了“getting-started”和“bursting”两组例子。PCI 可以在以下地址下载：

<http://pentaho.org/downloads/platform>

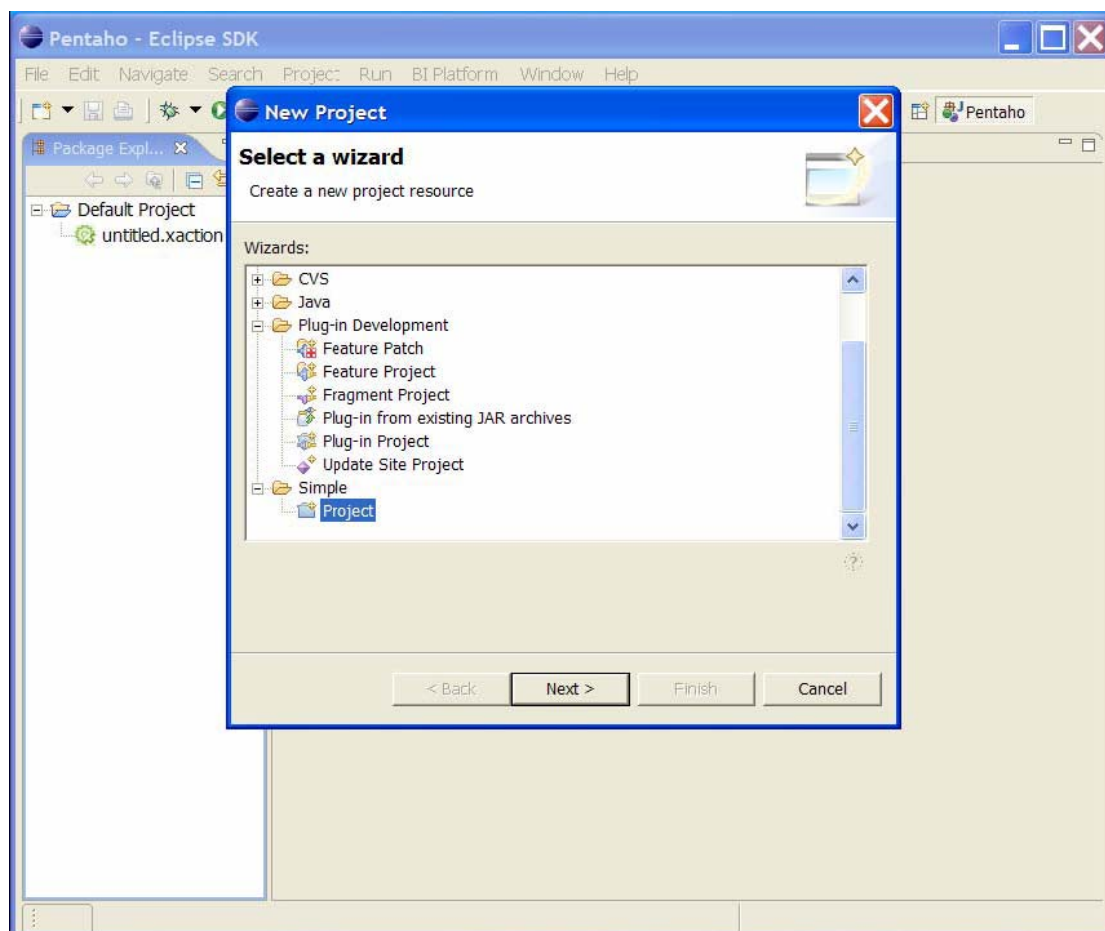
找名为pentaho_demo-x.x.x.zip的文件

运行 Design Studio

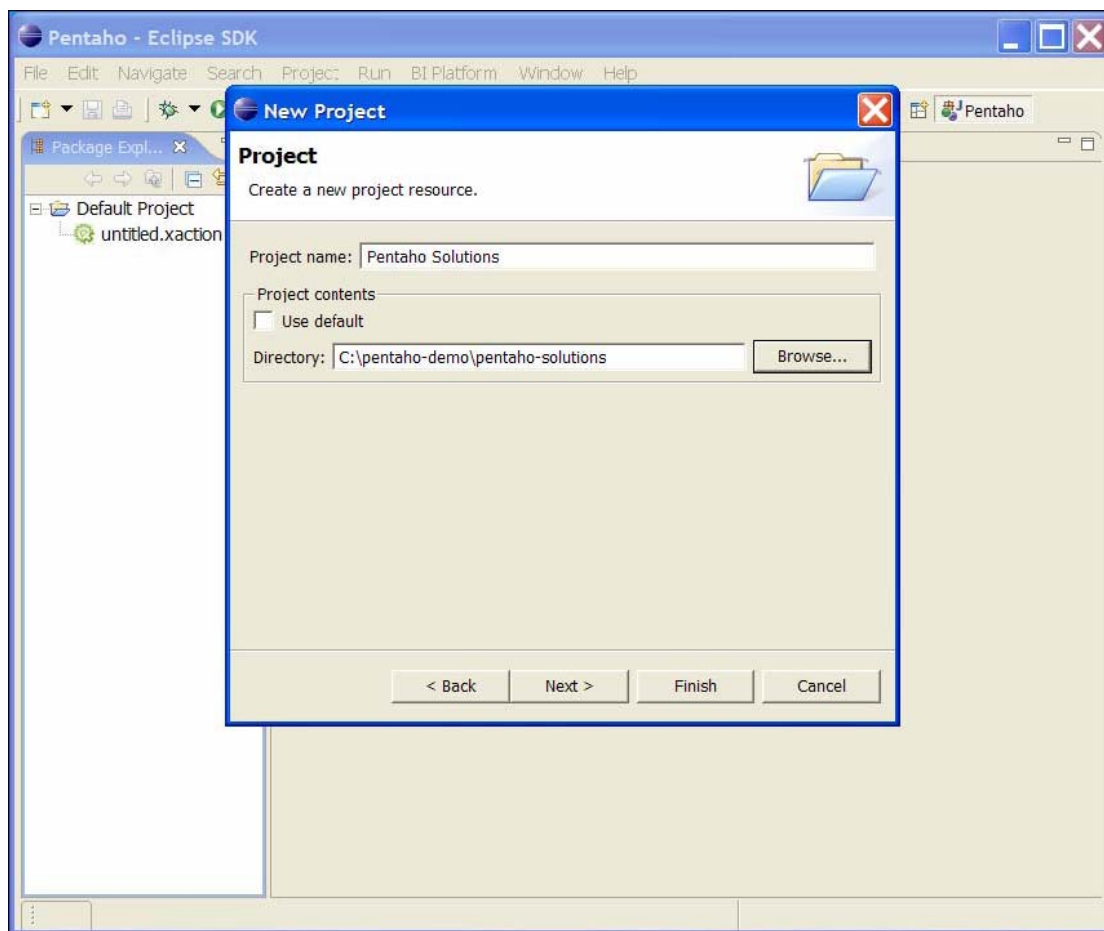
现在，你应该已经安装了一个单独的 Design Studio，或者在你的 Eclipse 里面安装了 Action Sequence 插件。你有可以运行的 Pentaho 例子。这些例子在你的浏览器里面测试成功。你的 Pentaho BI 服务器已经开始运行并等待请求。

如果你还没有完成这些，请如安装部分描述的那样，启动 Design Studio，然后单击 Welcome 旁的 X，关闭 Welcome 窗口，开始下一步的工作。

我们首先要做的事情是在 Design Studio 里面打开例子。选择 File->New->Project。然后选择在新项目向导里面选择 Simple，然后单击 Next 按钮。



输入“Pentaho Solutions”作为项目的名称。当然什么名称都是可以的，但是本文档会使用这个名称，取消掉 default 选择框，然后定位到 pentaho-solutions 的目录。如果你使用 PCI，这些应该在/pentaho-demo/pentaho-solutions。选择 finish 然后就完成了。



你现在应该可以看到新的 Pentaho Solution 项目显示在 Design Studio 的左边的树里。如果你展开目录，你会看到很多文件。在本教程里，我们主要关注扩展名为“.xaction”的文件。这些就是 Design Studio 所管理的 action sequence 文件。我们可以先看看一个，感觉以下 Design Studio 到底可以为我们做些什么。在左边的树里面，打开 Pentaho Solutions/samples/getting-started 文件夹。双击 HelloWorld.xaction 文件然后 Action Sequence 编辑器会打开。

快速体验

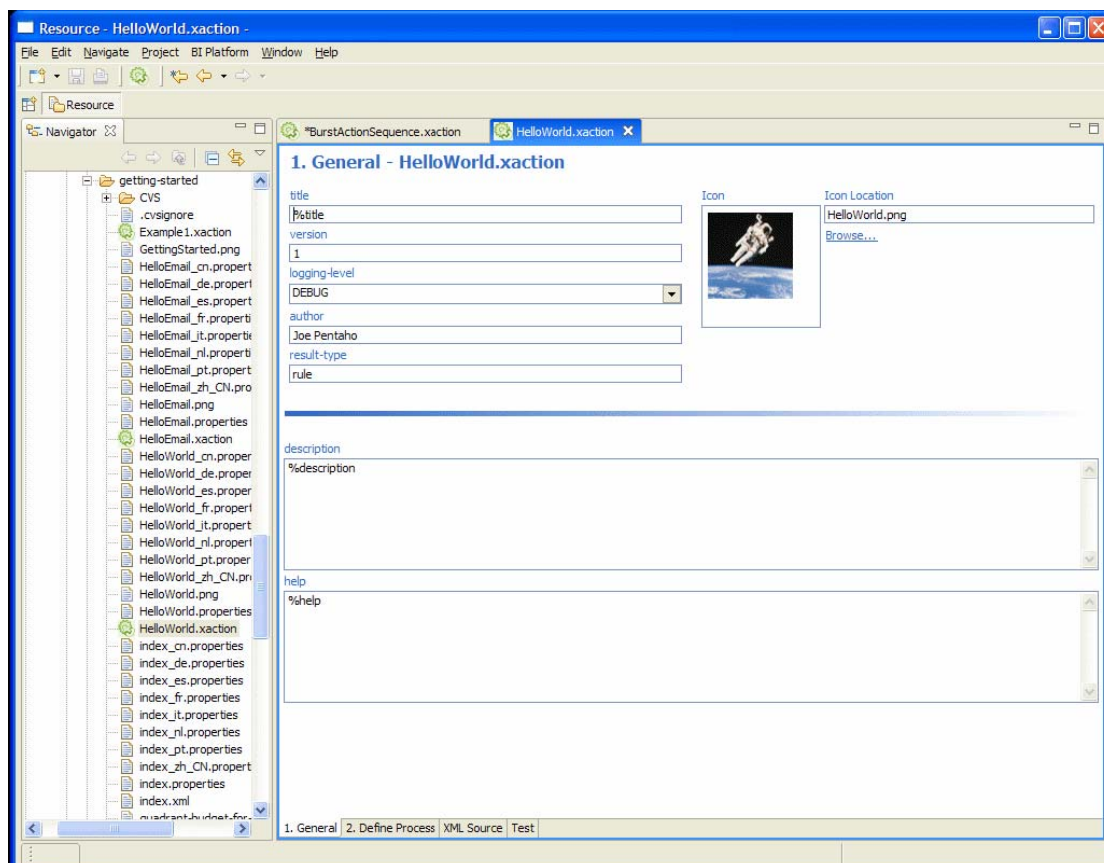
Action Sequence 编辑器的底部有四个 Tab 页：General，Define Process，XML Source 和 Test。每个 Tab 页的功能后面会详细的讨论，他们的基本功能是

1. General-基本的属性如标题，帮助等
2. Define Process-定义 Action Sequence 需要的输入，输出和资源，并且让你可以用编程控制 Action Sequence 和 Pentaho Component 之间的交互。
3. XML Source-编辑器产生的原始 XML
4. Test-提供一个界面，可在 Pentaho BI 服务器上执行 Action Sequence

依次点击每个Tab页以熟悉这些编辑器。看看XML source可以感觉一下编辑器为了节省了什么工作。最后，点击Test，这里我们可以测试配置。在test页的顶部，有一个输入框名为“Pentaho Server URL”。输入你的Pentaho BI服务器的URL，如果你使用的是PCI，这个地址应该是<http://localhost:8080/pentaho>。点击“Test Server”按钮，确保你在测试页上看到例子。点击“Run”执行HelloWorld Action Sequence。你应该看到熟悉的“Hello World. Greetings from the

Pentaho BI Platform.”信息。如果确实看到了，那么恭喜，你已经正确的配置了，可以继续了。

万一你确实不能看到HelloWorld信息，请确保服务器是运行的，而且你输入的URL是正确的。确保你在浏览器里面运行的确实是实例。如果还是失败，请到www.pentaho.org的Design Studio论坛提问



让我们对 Action Sequence 做一些修改,看看会发生什么。切换到第二个 Tab “Define Process” 点击 “Process Actions” 里面的 “HelloWorldComponent”。这个操作会在右边选择 HelloWorld Action 编辑器。在右边的 “Configuration” 部分，你会看到 “%quote” 修改 “%quote” 成其他的形式如 “Hello from the BI Design Studio”。

切换到 Test Tab 测试一下刚才的改动。选择 “File-> Save”（或者 Ctrl-S）保存改动。点击 Run 按钮新的信息会显示出来。Wow!!! Design Studio 的威力显示出来了。

主要一下，不要忘了我们实际上是在显示和修改你的 PCI 里面的 Action Sequence。所以小心不要覆盖了你不应该覆盖的文件。

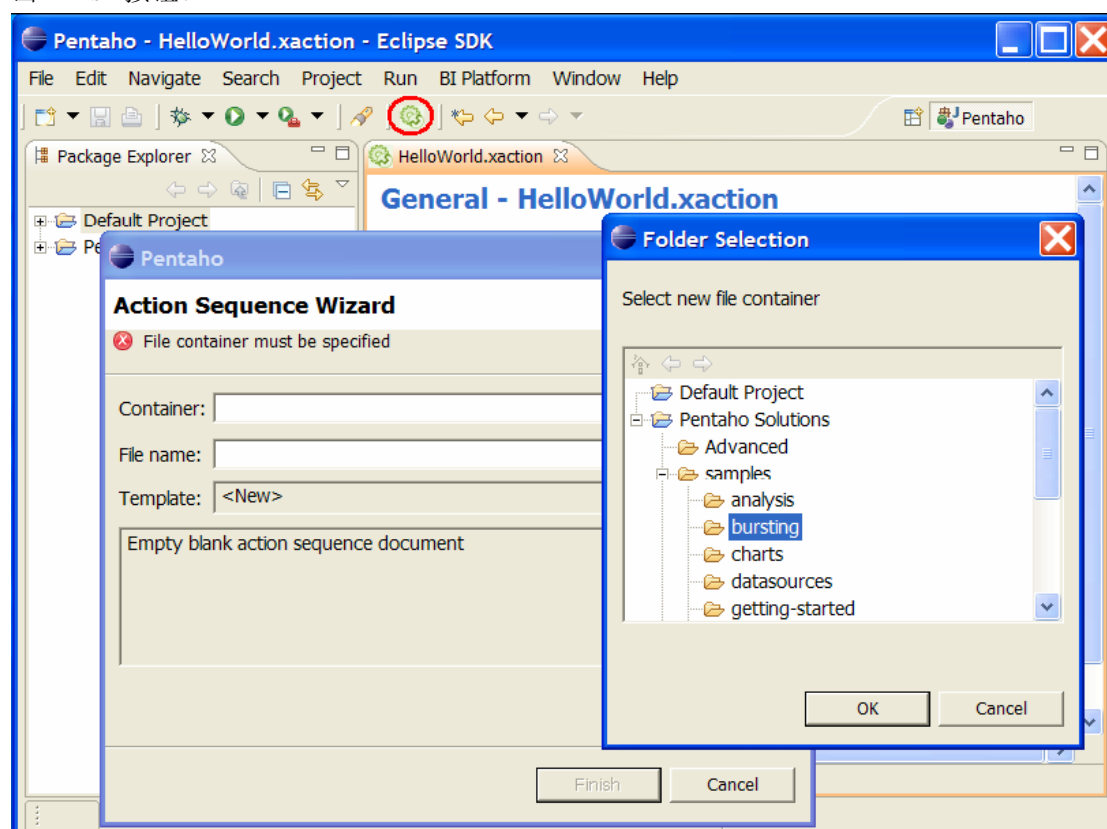
注意：

在最后一个例子里面，你可以输入 “Hey-what is that % stuff?” -%开头的字符串作为信息的关键字，根据当前使用的语言，会在合适的 properties 文件中查找本地化的字符串。这是解决方案国际化的方法。现在他们没有被 Design Studio 处理但是马上就会（在正式版发布之前）。现在仍然有一些地方涉及原始的 XML 文件，这些也会马上修正。

建立第一个 Action Sequence

如果你成功的完成了“开始使用”，你已经可以做更大和更好的事情了。那么让我们整理一下学习的知识，建立我们自己的 action sequence。

选择 Pentaho Action Sequence 图标（红圈里面的）。然后点 Container Browser 按钮，选择 PCI 里面的“getting-started”目录然后点 OK。现在给你的新 action sequence 一个名字。弄点有创意的如“Design StudioTutorial.xaction”。如果这个不爽的话换一个别的。一定要记得以“.xaction”结尾。最后，在“Template”下拉菜单里面选择“Sample Burst Action”然后点击 Finish 按钮。



Template 是快速完成 action sequence 的好方法。大量的信息已经为你设置好了。你所需要做的只是填上那些空白。你可以创建自己的 action sequence。完成以后，把他们保存到：

PentahoWorkbench\plugins\org.pentaho.workbench.editors.actionsequence_x.x.x.x/templates 目录

下一步次启动Action Sequence向导，你的action sequence会作为一个可用的模板。

回到现在正在进行的任务上，你可能要问“那么Sample Burst Action到底要做什么呢？”Burst 报表是一种可以为不同的人运行多次的报表。如果每个人有一个不同的参数来区别他所统计的数据，那么这个时候就是最合适的。在这个例子里面，我们执行一个查询以获得经理的列表，他们的邮件地址和每个人负责的领域。我们遍历结果集，产生一个用领域过滤的报表。报表和个人化的信息会被发给相关的经理。将会使用到以下的四个组件：

SQLLookupRule-生成领域和经理的列表

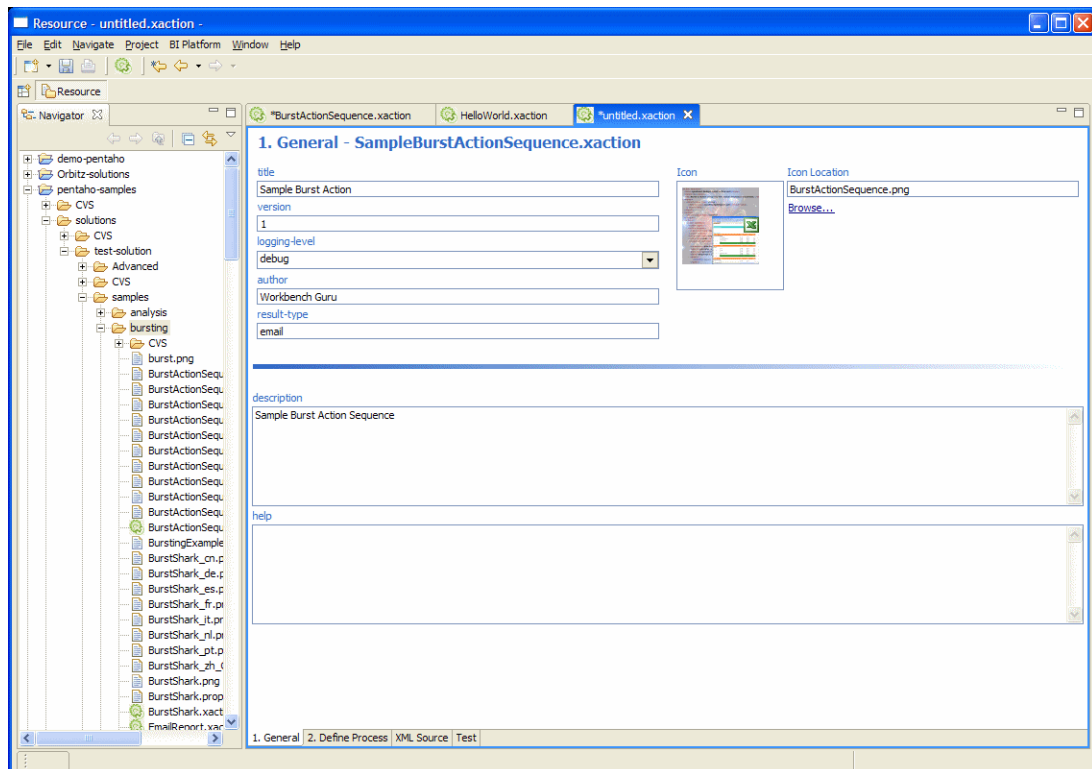
UtilityComponent-生成邮件信息，标题和附件的名称

JFreeReportComponent-生成报表

EmailComponent-发送报表

常用信息

顾名思义，General tab 页是管理 action sequence 的常用信息的地方。如浏览解决方案的时候可以看到的标题，描述和图标等。另外还可以指定要使用的日志的级别。日志信息在 pentaho-demo/jboss/server/default/log/server.log 文件里面。如果遇到不能成功运行你的 action sequence，日志文件记录了有用的信息帮助你定位错误。注意我们使用的 template 没有指定缺省的图标。我们可以点击“Browse...”按钮来指定一个图标，可以选择 samples/bursting/BurstActionSequence.png。为了试验一下，可以把描述信息也修改一下。选择 File-> Save 然后我们可以进行后面的操作。



输入、输出和资源

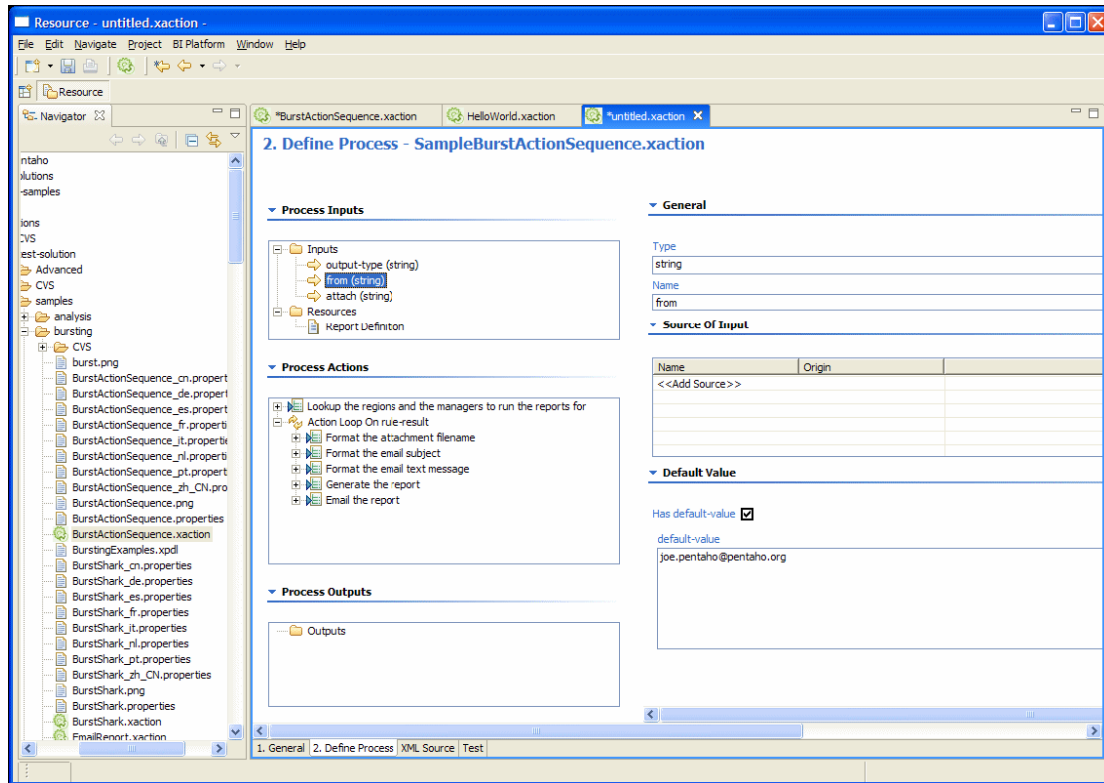
现在点击“Define Process”tab 页。这里有三个部分，分别是“Process Inputs”，“Process Actions”和“Process Outputs”。Process Inputs 管理输入和资源，Process Actions 管理组件，Process Outputs 管理输出。输入，输出和资源相当于 action sequence 和外部世界交互的约定。Process Action 会在后面进行讨论。

输入和资源是 action sequence 运行时需要从外部世界获得的信息。输入来自于请求（一般以 URL 的形式），session 或者运行环境。资源是 action sequence 完成工作所需要的文件。例如：如果 action sequence 要运行一个 JFree 报表（现在的例子就是这样），就肯定需要一个 JFree 报表文件的地址。

输出时 action sequence 完成时会留下的东西，其他的 action sequence 可以在此 action sequence 结束后利用这些东西。注意现在做的这个 action sequence 没有任何输出。

选择输入名称“from”。在右边，“form”的细节会显示出来。我们可以知道这是一个字符串，其缺省值看起来像是某人的电子邮件地址。现在在树里面选择“attach”，请注意这也是一个

字符串，缺省值为“report-output”。最后选择“output-type”。这是一个字符串，缺省值为“xls”（即 Excel 电子表格文件的扩展名）。

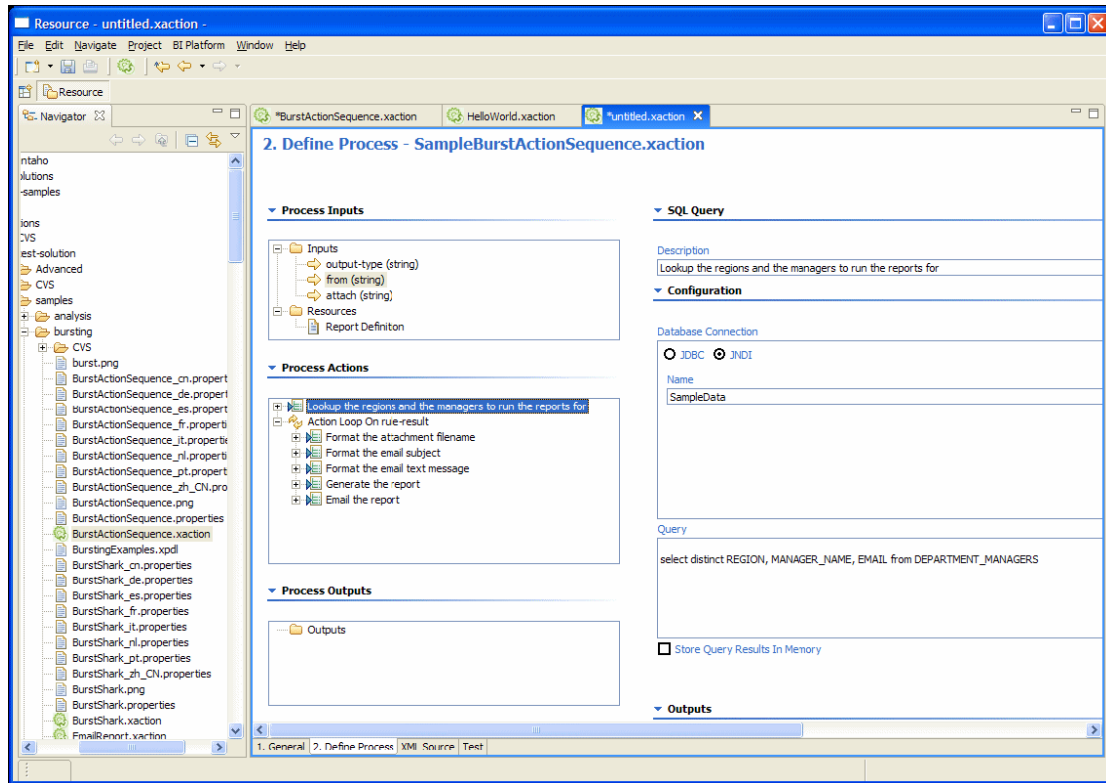


输入的源（Source）是需要你指定的输入的具体位置。在 action sequence 运行的时候有三个地方可以作为输入的来源。首先是请求（一般是 http 请求）。请求里面来的输入值是由触发 action sequence 的 URL 决定的。另外一个途径是 session（通常就是 action sequence 所在的 http session）。最后，输入的缺省来源是运行环境的上下文。这也是存放每个 action sequence 的输出，供其他 action sequence 使用的地方。在我们的例子里面，输入不会来源与以上任何一个途径。而是给每个输入指定了缺省值。可以给每个输入指定多个源（source），和一个缺省值。在这种情况下，查找输入的时候会按照源的顺序依次查找，最后才会使用缺省值。现在让我们看一下资源。在这个例子里面只有一个资源，叫做“Report Definition”。选择它看一下它的细节。看起来像是我们的解决方案里面的一个文件，但是似乎有人忘记了敲入集体的地址。实际上这是有意留成空白的。别忘了我们是从一个模板开始设计的。这个是我们必须填上的一个空白。既然是解决方案里面的文件，我们只需要指定该文件相对于 action sequence 的路径。在这个例子里面，输入“JFreeQuadrantForRegion.xml”作为文件的位置。现在我们比刚开始的时候对此 action sequence 了解更多了。如果我戴上我的福尔摩斯帽子（喔它听起来搞笑但是一定程度上帮助我更清晰的思考），我可能可以猜一下这个 action sequence 到底要做什么了。我们知道它需要一个报表定义文件，一些电子邮件地址，一个输出格式（Excel 电子表格），还有一些附件信息。OK，大胆的猜测：也许这个 action sequence 要执行一个报表，然后发送邮件给某人，把报表的结果作为 Excel 电子表格的形式附在里面。干嘛要猜呢？让我们看一下 Process Action 部分，看到底这个 action sequence 要做什么。

Action

现在越来越有趣了。你在 Process Action 里面看到是所有要执行的 action 的列表。请注意顺序是重要的。最顶部的 action 会最先执行，随后是下面的一个，依此类推。第二个 action，

就是以“Action Loop”开头的那个，也许最值得关注。这是一个循环的 action，会重复多次执行它里面所包含的 action，具体执行多少次要取决于配置。在这个例子里面这个循环看起来包含 5 个 action。让我们进一步探索一下。点击列表里面的第一个 action，就是以“Loop”开头的那个，看一下我们会见到什么。



在右边你可以看到 Lookup action 的细节。你会注意到它分为 3 个部分：SQL 查询，配置和输出。有时候你也会看到输出部分，但是这一个 action 没有任何输入，所以没有显示输入部分。SQL 查询部分有输入框让你输入一个简单的描述。输入描述没有必要，不过是一个好主意，因为这会让 action sequence 更容易理解。配置部分是输入 action sequence 的配置信息的地方；每个组件都有与自己相关的配置。在这个例子里面，可以输入数据连接和查询。输出部分是 action 声明其输出的地方，输出可能被同一个 action sequence 里面的 action 使用。类似于前面介绍的 Process Inputs 和 Process Outputs 一样，那些是整个 action sequence 声明其输出和输入的地方，action sequence 内部的每一个 action 也和其他的 action 达成了一个“协议”。所以如果一个 action 希望其运行的时候有什么样的信息，这个 action 必须在输入部分做出声明。如果它需要给其他 action 留下什么信息，也必须在输出部分声明。例如我们现在正在看的 action 产生了一个叫做 REGION 的输出。后面的其他 action 如果有需要可以以此作为输入。即一个 action 可以生成输出供后面的 action 作为输入。另外每个 action 还可以使用来自文档的输入。这些输入是来自（action sequence 的）外部。每个 action 都有自己可以处理得很好的东西。它读取输入，进行一些操作，然后生成输出供其他的 action 使用。你的工作是把这些孤立的 action 整合在一起来完成有意义的事情。

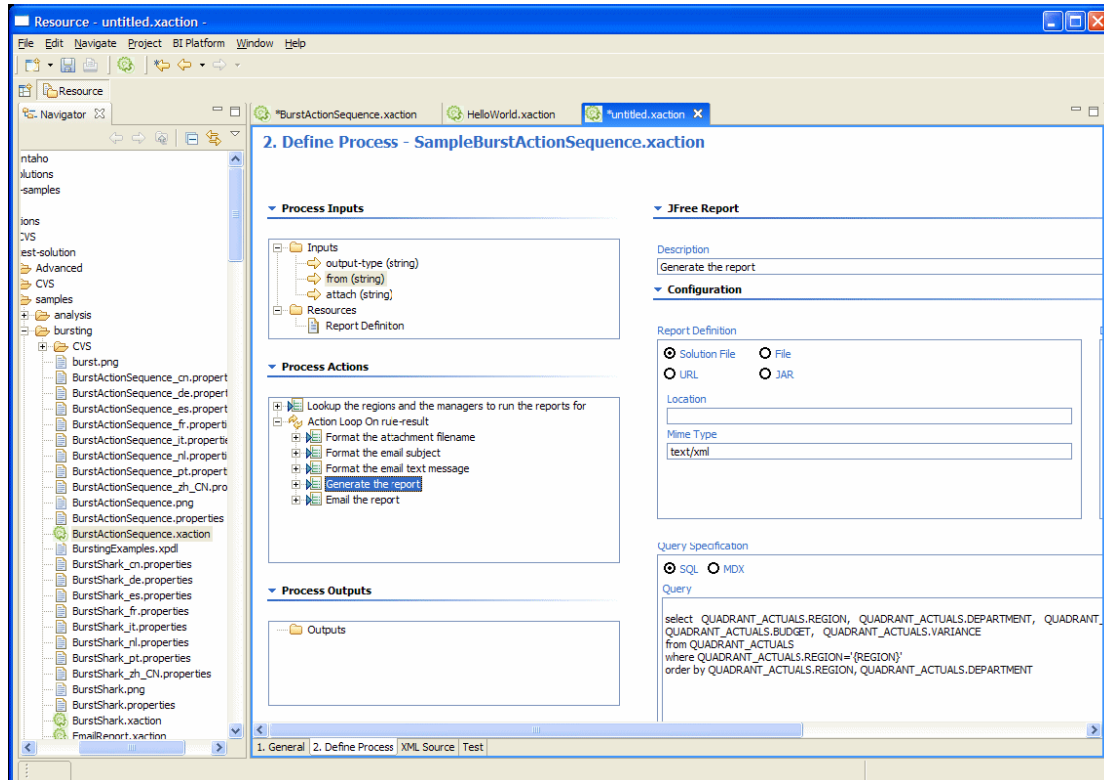
现在让我们快速的看一下每个 action 以及他们是怎么协作以完成有用的工作的。我们只是浏览一下过程，不会注重细节。现在是要体验一下这些 action 是如何协作的。后面我们会学习每个 action 的细节。

让我们从现在 action 树中选定的“Lookup...”开始。它执行一个 SQL 查询，从数据库里面获取 region, manager 和 email 的信息。注意它产生了一些输出。Rule-result 是存放 SQL 查询的输出的标准为之。其他三个输出告诉外部世界信息的字段名。做任何查询的时候都要记

得把字段名包括在输出中。这样其它 action 才知道 rule result 里面有什么信息。

下一个是 action 循环。如果你选择它，会发现没有多少东西。它只是前面的查询输出上进行一个循环。所以对于前面查询产生的记录，循环里面的 5 个 action 会依次运行一次。

后面的 3 个 action 是类似的，它们是格式化字符串的 action。它们读入输入字符串，放置到格式化的字符串中，然后把格式化的字符串作为输出。基本上它们整理输入供后面的 action 使用。



点击名为“Generate the report”的 action。现在我们看到了真正的工作了。这个 action 会产生一个 JFree 报表。在配置部分你可以看到数据库连接和会传递给 JFree 报表的查询。把注意力转移到输入部分你可以看到前面 Lookup 生成的 REGION 作为这个 action 的输入。事实上它是作为 JFree 报表所使用的查询的一个部分（看 where 语句的{REGION}）。另外这个 action 把 output-type 作为输入。output-type 的缺省值为“xls”，表示一个 Excel 电子表格。这个 action 以这种方式知道要输出的报表的格式。转到输出部分，你可以注意到报表保存到名为“report-output”的输出里面。这个是此类型的 action 的输出的标准名称。那么哪里是 JFree 报表的定义呢？它不在输入和输出里面。实际上如果你回头看输入和输出 Tab，你会发现这里有一个报表定义资源。这种类型的 action 知道要找这样的资源。这个资源指向什么地方，就是被 action 用来生成 JFree 报表。现在我们生成了报表而且它保存在一个叫做 report-content 的输出里面。现在要做什么？

选择最后一个 action。名字已经说明了一切。这个 action 会把报表用 email 发送给生成的报表所在的 region 的经历。看一下输入你就会明白这个 action 是怎么整合各种输入，把报表作为附件发送的。

似乎我们开始估计 action sequence 的目的是相当精确的（我会把帽子摘下来，它把我的耳朵弄出汗了）。这个 action sequence 会为每个区域经理生成 JFree 报表并给每个人发送与他相关的报表。真是一个很猛的东西!!!

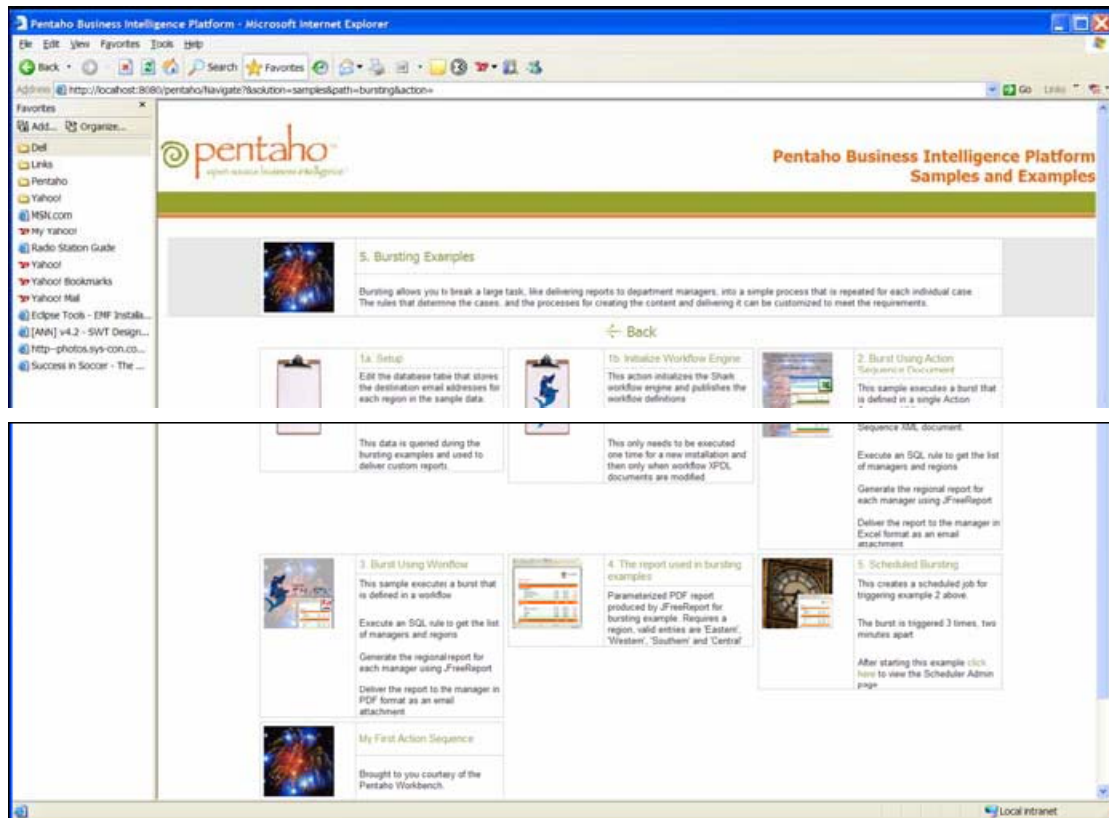
如果没有明白每个细节，不必担心。每个类型的 action 有自己的输入和输出。你熟悉了每个以后，你就可以很快的把它们组合起来得到有用的东西。

我们不能停在这里。现在我们会把这些东西放在一起看看是怎么运作的。

运行你的 Action Sequence

首先我们要发布做的东西到 PCI 里面去让其他人可以访问它。只要简单的把 action sequence 拷贝到 PCI 中正确的目录里面就可以了。回忆一下当我们开始教程的时候，我们以 bursting 目录作为放 action sequence 的地方。我们只要选择 File-> Save 发布过程就完成了。

你可以使用 Test tab 页里面的 RUN 按钮来运行你的 action sequence。发布以后，你也可以使用你自己的浏览器来运行。选择“My First Action Sequence”你就可以看到“Action Successful”的信息。你在 Quick Start Bursting Examples 里面配置的经理们就可以收到邮件了。



现在还可以做什么？

你已经完成了。创建了一个真实的 action sequence 并使能为世界所用，恭喜！Design Studio 进一步的发展，我们这个教程也会。有问题，需要请访问 www.pentaho.org 的 Design Studio 论坛。