

IBM DB2 Alphablox



入门

版本 8.4

IBM DB2 Alphablox



入门

版本 8.4

注意:

在使用本资料及其支持的产品之前，请阅读第 31 页的『声明』中的信息。

第二版 (2006 年 3 月)

本版本适用于 IBM DB2 Alphablox for Linux, UNIX and Windows (产品号 5724-L14) 的 V8R4 及所有后续发行版和修订版，直到在新版本中另有声明为止。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何信息，有权利以任何它认为适当的方式使用或分发，而不必对您负任何责任。

Copyright © 1996 - 2006 Alphablox Corporation. All rights reserved.

© Copyright International Business Machines Corporation 1996, 2006. All rights reserved.

目录

第 1 章 教程: 构建您的第一个应用程序 . . . 1

定义应用程序	1
访问数据	2
创建应用程序主页	2
设置缺省主页	3
创建您的第一个分析视图	3
第一个分析视图页的结构	6
创建您的第二个分析视图	8
总结	9

第 2 章 教程: 构建您的第一个带有 Blox 组件的 portlet 11

安装样本 portlet	11
运行样本 portlet	12
检查带有 Blox 组件的 portlet JSP 页的结构	12
创建您自己的带有 Blox 组件的 portlet JSP 页	14
创建要使用 Blox 组件的 portlet 项目	15
使用 Rational Application Developer 配置 portlet 项目	16
后续步骤	17
Portlet 开发技巧	17

第 3 章 教程: 使用 Rational Developer 工具构建应用程序 19

准备开发环境	19
------------------	----

安装 DB2 Alphablox 工具箱	19
在 WebSphere 集成测试环境中安装 DB2 Alphablox	20
创建 WebSphere 服务器实例	21
创建 WebSphere 5.1 服务器替换变量	22
配置 WebSphere 5.1 服务器实例	22
将来宾用户添加至 DB2 Alphablox 管理员组	23
创建 DB2 Alphablox 应用程序	23
创建带有 DB2 Alphablox 内容的 JSP 文件	24
访问 DB2 OLAP Server 和 Essbase 数据源	25

第 4 章 教程: 使用 DB2 Cube Views 构建 DB2 Alphablox 立方体 27

定义 DB2 关系数据源	27
定义 Alphablox Cube Server Adaptor 数据源	28
定义 DB2 Alphablox 立方体	29
启动 DB2 Alphablox 立方体	29

声明	31
商标	32

索引	35
--------------	----

第 1 章 教程: 构建您的第一个应用程序

本教程将教您一些有关 Blox 组件的比较重要的基础知识, 同时还提供了快速构建 DB2® Alphablox 应用程序的步骤。

要构建 DB2 Alphablox 应用程序, 需要用到 JavaServer Pages (JSP) 技术, 但是本教程不涉及 JSP 知识。本教程将说明构建分析应用程序需要了解的 JSP 知识。要了解有关 JSP 技术的更多信息, 有很多好书可供参考。

先决条件

本教程中的步骤假定您正在使用的是在 WebSphere® Application Server 上运行的 DB2 Alphablox。如果您将 BEA WebLogic 或 Apache Tomcat 用作应用程序服务器, 则可能需要更改下面的某些步骤以适应您的服务器的特殊需求。

另外, 在开始本教程之前, 您需要安装 DB2 Alphablox 立方体 (qcc_2003) 及其底层的关系数据源 (qcc2003-rdb)。在 DB2 Alphablox 安装 CD 的 `sampledata/qcc/acs` 目录中, 可以找到有关配置本教程所需的 DB2 Alphablox 立方体和关系数据库的样本数据库文件以及指示信息 (在 `readme.txt` 文件中)。

定义应用程序

要使用 J2EE 开发方式创建应用程序, 需要创建带有 WEB-INF 目录的目录结构, 此目录包含应用程序描述符文件 (`web.xml`)。在 DB2 Alphablox 中创建此结构的最简单的方法是使用“DB2 Alphablox 管理页面”中的“应用程序”页面来创建新的应用程序。

要创建应用程序和文件夹:

1. 通过遵循《管理员指南》『应用程序定义』一节中所描述的步骤, 在 DB2 Alphablox 中创建名为 MyApp 的新应用程序。
 - 在名称字段中输入 MyApp。
 - 在显示名称字段中输入 My App (带有空格), 它定义“应用程序”页面的列表中所显示的该应用程序的标签。
2. 单击页面左上部的应用程序选项卡。会打开可用应用程序的列表。
3. 单击应用程序列表中新创建的应用程序的名称 (My App)。由于您尚未创建任何文件, 所以此文件目录是空的。
4. 浏览至应用程序服务器上的新应用程序文件夹 (对于 WebSphere, 此文件夹位于 `WebSphere installedApps` 目录中)。注意, 此应用程序包括带有 `web.xml` 文件和 `tlds` 子目录的 WEB-INF 目录。`web.xml` 文件定义应用程序信息。`tlds` 目录包含 Blox 标记库描述符文件 (`blox.tld`), 它定义将用来在本教程中创建分析视图的 Blox 标记。此目录还包含本教程没有用到的一些 Blox 标记库的标记库描述符 (TLD) 文件, 如 `bloxform.tld`、`bloxlogic.tld`、`bloxreport.tld` 和 `bloxui.tld`。

您已创建了 DB2 Alphablox 应用程序框架, 现在可以向其中添加分析视图。

访问数据

对于本教程，您将会使用到开始本教程之前所创建的 DB2 Alphablox 立方体数据源。您的关系数据库和 Alphablox Cube Server 必须正在运行。同时，要使用的 DB2 Alphablox 立方体必须正在运行。

要验证所需的数据源是否可用：

1. 打开浏览器并浏览至“DB2 Alphablox 管理页面”。
2. 单击**管理**选项卡。
3. 单击**数据源**。
4. 在“数据源”页面左边的菜单中查找 qcc2003-rdb 数据源。如果未找到此数据源，则需要复查 readme.txt 文件中所描述的步骤，此文件位于 DB2 Alphablox 安装 CD 的 sampledata/qcc/acs 目录中。
5. 单击可用数据源列表中的 qcc2003-rdb，然后单击**测试所选数据源**按钮。如果数据库正在运行并且数据源定义配置正确，则会出现成功消息。如果出现错误消息，则需要检查数据源定义（选择此数据源并单击**编辑**按钮），或检查是否对此数据库正确配置了用户访问权。
6. 从数据源列表中选择 qcc2003-acs 数据源，然后单击**测试所选数据源**按钮。如果正确配置了 Alphablox Cube Server 数据源，则会出现成功消息。如果出现错误消息，则检查数据源定义。

在成功测试了数据源之后，可开始学习如何构建一组简单的 DB2 Alphablox 分析视图。

创建应用程序主页

对于此任务，您可使用基本 Web 技术来创建简单的主页。

对于本教程，您只需要创建极简单的 Web 应用程序结构。在主页上，需要创建两个链接，用于访问您在本教程中创建的两个分析视图。

要创建简单的主页：

1. 使用文本编辑器或您常用的开发工具，将其命名为 index.html。对于本教程，您可使其更简单些。
2. 编辑此文件以指定应用程序标题（例如，“My DB2 Alphablox Application”）和两个链接，它们与您将要创建的两个分析视图连接。

将下列代码复制并粘贴（或输入）至 index.html 文件。

```
<html>
<head>
<title>My DB2 Alphablox Application</title>
</head>
<body>
<h2>My DB2 Alphablox Application</h2>
<p>
<a href="PresentBloxView.jsp">Simple PresentBlox View</a>
</p>
<p>
<a href="ChartView.jsp">Customized Chart View</a>
</p>
</body>
</html>
```


3. 将此文件保存至定义此应用程序时所创建的 MyApp 目录。
4. 打开 Web 浏览器并浏览至 DB2 Alphablox 主页。缺省情况下，浏览器会显示应用程序选项卡。
5. 单击 **My App** 链接以验证此文件是否包括在目录列表中。单击 **index.html** 链接以查看您的主页。如果 DB2 Alphablox 安装在 IBM WebSphere Application Server 或 BEA WebLogic 服务器上，则必须先启用目录列表，然后才能执行此步骤。对于安装在 Apache Tomcat 上的 DB2 Alphablox，应不需要其他步骤。如果出现 HTTP 403 错误（缺省情况下不允许使用目录列表），可以修改服务器以允许使用目录列表，或者仅将文件名 `myapp.html` 追加至 URL 的末尾即可（例如，`http://localhost:9080/MyApp/myapp.html`）。在使用标准链路地址时，应显示此新页。

您现在已创建了一个主页，该主页上带有两个指向下一步将创建的两个分析视图的链接。

设置缺省主页

在前面的步骤中，您通过单击 MyApp 应用程序目录中的文件名链接或通过 URL 中指定主页文件 `index.html` 直接访问了主页。在此步骤中，您可指定此应用程序的缺省主页。

要设置缺省应用程序主页：

1. 打开浏览器并浏览至 DB2 Alphablox 主页。
2. 单击**管理**选项卡。
3. 单击**应用程序**链接。
4. 在应用程序列表中选择 **MyApp** 并单击列表下面的**编辑**按钮。
5. 在 MyApp 的“编辑应用程序”页面上，在**主页 URL** 字段中输入 `index.html`。
6. 单击**保存**按钮以保存所做的更改。
7. 单击主“应用程序”选项卡（左上部的文件夹选项卡）以返回至“应用程序”页面。
8. 单击 **My App** 应用程序名称。此应用程序现在直接打开到已定义的主页。

定义主页之后，即使用户没有在 URL 中指定主页，服务器也知道从何处启动该应用程序，并会在打开应用程序时定位到此页面。这样，如果用户在浏览器地址栏中输入 `http://yourServerName/MyApp/`，应用程序将自动打开到已定义的主页 `index.html`。

在后面的主题中，将使用由 DB2 Alphablox Query Builder 应用程序生成的简单 Blox 标记来创建 DB2 Alphablox 分析视图。

创建您的第一个分析视图

在此任务中，您将创建 DB2 Alphablox 分析视图，它用于显示组合到 PresentBlox 组件中的网格和图表。

先决条件： 请阅读《开发者指南》的介绍性部分，以了解 DB2 Alphablox 及其组件。这将使您初步了解 DB2 Alphablox 功能以及如何通过 JSP 技术使用 Blox 定制标记库来构建分析应用程序。

对于要创建的第一个分析视图，将使用 DB2 Alphablox Query Builder 应用程序来生成 PresentBlox 组件定义的 JSP 定制标记，此组件是用于显示数据的最常用 Blox 组件。PresentBlox 组件可包括网格和图表视图、数据布局面板、工具栏及其他嵌套组件。网格和图表视图显示相同数据的同步备用视图。

要使用 Query Builder 创建您的第一个分析视图：

1. 打开浏览器并浏览至“DB2 Alphablox 管理页面”。缺省情况下，可看到“应用程序”选项卡。
2. 单击 DB2 Alphablox Query Builder 以打开此应用程序。
3. 单击**连接设置**按钮。会出现“数据库连接”窗口。
4. 选择 qcc2003-acs 数据源并按**连接**按钮。此窗口关闭，此时应看到“已连接”字样出现在 Query Builder 页面上的“数据库状态”部分中。

注：您不需要填写**模式、目录、用户名和密码**字段的值，因为会从所选数据源的数据源定义中自动检索它们。另外，不要选择**执行缺省查询**复选框。

5. 在此页上找到“查询”部分，它位于“状态”部分的下方。请注意以下事项：
 - 对于本教程，您使用的是 DB2 Alphablox 立方体 (qcc2003-acs)。您将在“查询”部分标题的正下方看到可用（当前正在运行）的立方体列表。应该列出了 qcc2003-acs 立方体。
 - 与此数据源关联的缺省查询字符串会出现在文本窗口中。由于此数据源是 DB2 Alphablox 立方体，所以按字母顺序列出的第一个立方体会添加至查询语句中（例如，“select from [datasourceName]”）。

注：如果启用了 DB2 Alphablox Cube Server 中的连接池功能，则不能使用**获取缺省查询**按钮来获取 DB2 Alphablox 立方体的缺省查询。

- - 当数据出现在页面底部的 PresentBlox 中时，可使用标准用户界面来交换轴、向上钻取或向下钻取、在轴之间移动维或执行其他操作。
 - 要查看查询结果，单击“执行查询”按钮。结果集会出现在页面底部的 PresentBlox 中。
6. 在缺省查询的方括号之间输入 qcc2003-acs（例如，select from [qcc2003-acs]），然后按**执行查询**按钮。

PresentBlox 部分将根据查询来显示缺省视图。

7. 使用 PresentBlox 用户界面来创建新视图。

例如，您可将 Products 维拖至左边数据布局面板中的 Row 轴。然后，可将 Time (Calendar) 维拖至 Column 轴。接着，您可将 Measures 维拖至 Page 轴，这会使 Measures 菜单出现在页面面板中，该面板在显示区域上方显示页面过滤器。选择两个其他维并将它们添加至 Page 轴，以便这三个页面过滤器都出现在此显示内容的上方。如果您第一次使用 DB2 Alphablox 用户界面，则可从 PresentBlox 工具栏中选择**帮助**，以了解有关使用此界面的更多信息。

8. 在创建新视图之后，请注意出现在“查询”部分的查询语句。如果没有动态生成查询（在未选中“自动更新查询”选项的情况下），则可单击**获取当前查询**按钮以获取当前查询。文本框显示开发当前数据视图所需的查询语句。您可从“查询”字段复制并粘贴开发 DB2 Alphablox 应用程序时要用到的查询语句。

- 单击**生成 Blox 标记**按钮。会打开一个对话框，显示用于复制 PresentBlox 布局和结果集的标记（和嵌套标记）。
- 在文本编辑器中打开一个新文件并将下列文本添加至此文件，然后将此文件保存为 PresentBloxView.jsp。下一个主题将说明此页面上代码的许多重要元素。

```
<%@ taglib uri="bloxtld" prefix="blox" %>
//Copy and paste the tag information from the generated Blox tag
//immediately below this line, removing any line breaks from
//query statement. You can also remove this comment after adding
//the code from DB2 Alphablox Query Builder.
<html>
<head>
<blox:header/>
</head>
<body>
<h2>Simple PresentBlox View</h2>
<p>
<blox:display bloxRef="MyPresentBloxView"/>
</p>
</body>
</html>
```

- 将已保存的文件（PresentBloxView.jsp）复制到服务器上的 MyApp 目录中，并通过从 MyApp 主页单击到此页面的链接来测试新分析视图 - 简单的 PresentBlox 视图。您也可直接在浏览器地址栏中输入 URL（例如，<http://localhost:9080/MyApp/PresentBloxView.jsp>）。您看到的视图应该与您在 DB2 Alphablox Query Builder 中处理的 PresentBlox 类似。

完整版本的“简单的 PresentBlox 视图”页（PresentBloxView.jsp）的代码示例如下所示：

```
<%@ taglib uri="bloxtld" prefix="blox" %>
<blox:present
id="queryBuilder4_present"
height="500"
visible="false"
width="100%">
<blox:grid/>
<blox:chart/>
<blox:page/>
<blox:data
dataSourceName="qcc2003-acsc"
onErrorClearResultSet="true"
query=" SELECT DISTINCT( {[qcc_2003].[Time (Calendar)].[All Time (Calendar)], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2000], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2001], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2002], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2003]} )
ON AXIS(0), DISTINCT( {[qcc_2003].[Products].[All Products], [qcc_2003].[Products].[All Products].[100 Truffles], [qcc_2003].[Products].[All Products].[200 Chocolate Blocks], [qcc_2003].[Products].[All Products].[300 Chocolate Nuts], [qcc_2003].[Products].[All Products].[400 Specialties]} )
ON AXIS(1) FROM [qcc_2003] WHERE ([qcc_2003].[Measures].[Sales], [qcc_2003].[Time (Fiscal)].[All Time (Fiscal)], [qcc_2003].[Date Opened].[All Date Opened], [qcc_2003].[Has Nuts].[All Has Nuts], [qcc_2003].[Chocolate Type].[All Chocolate Type], [qcc_2003].[Ounces Per Package].[All Ounces Per Package], [qcc_2003].[Pieces Per Package].[All Pieces Per Package], [qcc_2003].[Date Introduced].[All Date Introduced], [qcc_2003].[Seasonal].[All Seasonal], [qcc_2003].[Scenario].[Actual], [qcc_2003].[Locations].[All Locations])"
selectableSlicerDimensions="[qcc_2003].[Measures], [qcc_2003].[Locations],[qcc_2003].[Chocolate Type]"
```

```

useAliases="true"/>
<blox:toolbar/>
<blox:dataLayout/>
<bloxui:calculationEditor />
</blox:present>
<html>
<head>
<blox:header/>
</head>
<body>
<h2>Simple PresentBlox View</h2>
<blox:display bloxRef="queryBuilder4_present"/>
</p>
</body>
</html>

```

在上述代码示例中，为方便阅读，查询语句中带有换行符。要正常运行此页，需要除去属性值中的换行符（如上述 `query` 和 `selectableSlicerDimensions` 属性）。在 DB2 Alphablox Query Builder 生成的代码中，包含许多不必要的标记。此处可以看到所有的标记，能让您了解如何将 Blox 标记用作父标记和嵌套标记。还可看到标记属性的使用，它们是您在 Query Builder 的 PresentBlox 中进行操作时所添加的属性。要了解有关使用 Blox 标记、标记属性以及如何开发 DB2 Alphablox 应用程序的更多详细信息，请访问 DB2 Alphablox 信息中心。

第一个分析视图页的结构

本主题将概括介绍前一个主题中所创建的第一个分析视图（`PresentBloxView.jsp`）的结构。

下面是对前一个主题中所创建页的某些重要内容的简短概述。有关本代码的更完整描述，请参阅 DB2 Alphablox 信息中心中所包含的相关主题。

`PresentBloxView.jsp` 页以下列行开始：

```
<%@ taglib uri="bloxtld" prefix="blox" %>
```

此行是 JSP taglib 伪指令，它通知服务器您要使用 Blox 标记库。uri 是指向标记库描述符文件的指针。定义为 blox 的 prefix 值告知服务器查找此页上任何以 blox 开头的标记，然后根据标记库描述符文件中的定义，使用 Blox 标记库来处理内容。

就在 taglib 伪指令的下面，使用下列标记及其标记属性来指定 PresentBlox 组件：

```

<blox:present
id="queryBuilder4_present"
height="500"
visible="false"
width="100%">
<blox:grid/>
<blox:chart/>
<blox:page/>
<blox:data
dataSourceName="qcc2003-acs"
onErrorClearResultSet="true"
query="SELECT DISTINCT({[qcc_2003].[Time (Calendar)].[All Time
(Calendar)], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2000],
[qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2001],
[qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2002],
[qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2003]})
ON AXIS(0), DISTINCT( {[qcc_2003].[Products].[All Products],
[qcc_2003].[Products].[All Products].[100 Truffles],

```

```

[qcc_2003].[Products].[All Products].[200 Chocolate Blocks],
[qcc_2003].[Products].[All Products].[300 Chocolate Nuts],
[qcc_2003].[Products].[All Products].[400 Specialties}} )
ON AXIS(1) FROM [qcc_2003] WHERE ([qcc_2003].[Measures].[Sales],
[qcc_2003].[Time (Fiscal)].[All Time (Fiscal)],
[qcc_2003].[Date Opened].[All Date Opened],
[qcc_2003].[Has Nuts].[All Has Nuts],
[qcc_2003].[Chocolate Type].[All Chocolate Type],
[qcc_2003].[Ounces Per Package].[All Ounces Per Package],
[qcc_2003].[Pieces Per Package].[All Pieces Per Package],
[qcc_2003].[Date Introduced].[All Date Introduced],
[qcc_2003].[Seasonal].[All Seasonal],
[qcc_2003].[Scenario].[Actual],
[qcc_2003].[Locations].[All Locations])"
selectableSlicerDimensions="[qcc_2003].[Measures],
[qcc_2003].[Locations],[qcc_2003].[Chocolate Type]"
useAliases="true"/>
<blox:toolbar/>
<blox:dataLayout/>
<bloxui:calculationEditor />
</blox:present>

```

<blox:present> 标记指定您想要在此处放置一个 PresentBlox，并且其 id 属性值为 queryBuilder4_present。id 属性使您能够识别此特殊 Blox 以用于脚本编制。注意，<blox:present> 标记包括设置为 false 的 visible 属性。设置为 false 时，将不显示 PresentBlox，直至在此页中编译器遇到 <blox:display> 标记。

使用标记定义的每个 Blox 有很多属性，但是除非您需要指定缺省值以外的属性值，否则不需要将它们包括在标记中。在此示例中，height 值为 500，width 值为 100%。高度和宽度可以使用像素或百分比来定义。

此处包含嵌套的 DataBlox，它将 dataSourceName 属性定义为 qc2003-acs。如果未指定数据源，那么在显示此页时，您将看到“没有数据可用”的消息。查询属性包括 DB2 Alphablox Query Builder 生成的 MDX 查询语句。如果您了解与 DB2 Alphablox Cube Server 配合使用的 MDX 查询语言，则可简化查询，使其更短。此处的查询语句是完整的，并将生成您在 Query Builder 应用程序中创建的视图。

此页的 <head> 部分包含一个特殊的 Blox 标记，它用于在显示此页前将重要代码添加至此页：

```
<blox:header/>
```

DB2 Alphablox 使用此标记以自动将所需的 HTML、JavaScript™ 和 CSS 代码添加至页的头部分。当服务器显示页时，此标记会将 CSS 链接添加至定义的 HTML 主题和用以避免高速缓存的 meta 标记。请记住将此标记输入到使用 Blox 组件的每个 JSP 页。如果不包括此标记，将无法正常显示页。

页的 <body> 部分是下列内容：

```
<blox:display bloxRef="MyPresentBloxView"/>
```

此 Blox 标记的作用是：显示在以上内容中指定的 PresentBlox。这使您能够在某个位置更改 Blox 组件或其嵌套组件，在此位置您还可有其他控制 Blox 组件行为的 Blox 标记和 JSP scriptlet。在复杂的 JSP 文件中，如果您熟悉 DB2 Alphablox，可以选择使 DataBlox 组件与 PresentBlox 标记分离，并在使用 Web 浏览器显示 Blox 组件之前，添加会修改 Blox 组件及其行为的 JSP scriptlet。

总的来说，此页中所使用的最重要的 blox 标记包括: <blox:header/>、<blox:present> 和 <blox:display>。这三个标记及其嵌套标记在页中指定了分析视图而无需任何 Java™ 代码。错综复杂的表示逻辑就是由这些 Blox 标记管理。您可以根据业务需求，通过添加嵌套 Blox 及修改属性值来定制视图。

要创建更复杂的视图和应用程序，可能需要其他的 JavaBean 组件和 Java 代码。

创建您的第二个分析视图

在此任务中，您将基于本教程第一个分析视图的代码创建一个分析视图，它仅显示带有页面过滤器的图表。

在许多应用程序中，仅使用网格或图表来显示数据。PresentBlox 组件可将 GridBlox 和 ChartBlox 作为嵌套组件组合在一起，并可在 PresentBlox 组件中一起显示，在这种情况下，您可禁用或隐藏不想看到的嵌套组件。使用 PresentBlox 组件创建带有页面过滤器的图表视图相对来说比较简单，通过修改某些嵌套 Blox 标记的标记属性可完成此任务。

在此任务中，将会重复使用第一个分析视图中的 PresentBlox 标记来创建第二个定制分析视图。

要创建用于显示带有页面过滤器的图表的分析视图:

1. 在编辑器中打开以前创建的 PresentBloxView.jsp 文件并将它另存为 CustomizedChartBlox.jsp。
2. 在下列 PresentBlox 标记的嵌套组件中，对每个标记都添加一个 visible 属性并将其值设置为 false: 添加此属性将导致这些嵌套组件不显示，而仅显示带有页面过滤器栏的图表视图。

```
<blox:grid visible="false"/>
<blox:chart visible="false"/>
<blox:toolbar visible="false"/>
<blox:dataLayout visible="false"/>
```

3. 修改 PageBlox 标记 (<blox:page>)，添加 labelPlacement 属性并将其值设置为 top。缺省情况下，页面过滤器的标签会显示在菜单的左边。添加此属性并将它设置为 top 会覆盖缺省行为并将标签置于页面过滤器的上方。

```
<blox:page labelPlacement="top" />
```

4. 将 <blox:present> 标记上的 id 属性名修改为值 chartview。在页面上显示的每个 Blox 组件都需要一个唯一标识以避免出错。
5. 保存所做的更改并将此文件添加到服务器上的 MyApp 目录中:
6. 打开 Web 浏览器，使用下列其中一种方式来测试新页: 单击主页中的此链接，或使用 URL (例如，http://localhost:9080/MyApp/CustomizedChartView.jsp) 直接访问它。

现在，您已使用 DB2 Alphablox Query Builder 应用程序帮助您创建了两个分析视图。另外，您也了解到通过修改或添加现有标记的属性，可以控制 Blox 及其嵌套组件的外观和行为。

总结

如果您已完成所有教程任务，那么就已了解到如何使用“DB2 Alphablox 管理页面”、DB2 Alphablox Query Builder 应用程序和 Blox 标记来构建基本的 DB2 Alphablox 应用程序。如果您阅读了 *Developer's Reference*，则会了解有许多属性和方法可用于在 JSP 页面上定义和处理 Blox。那么下一步该做什么呢？

如果想在 DB2 Alphablox 视图中看到您的企业数据，那么您现在需要知道如何从头开始创建应用程序并将分析视图添加至您的应用程序。如果您想立即看到某些数据，则可能要对 MyApp 应用程序做些简单修改。要从企业数据库而非 qcc2003-acs 数据源来显示数据，则可创建指向企业数据库的数据源，您可修改 dataSourceName 属性，使其指向新定义的数据源，然后添加合适的 query 属性。通过使用 DB2 Alphablox Query Builder，您可随时快速创建可用的查询语句和 Blox 标记。有关创建数据源的详细信息，请参阅《管理员指南》。要了解如何创建适当查询及获取有关构建 DB2 Alphablox 应用程序的更多信息，请参阅《开发者指南》中的『检索数据』一节以及 *Developer's Reference* 中的 DataBlox 一节。

第 2 章 教程: 构建您的第一个带有 Blox 组件的 portlet

在本教程中, 您将了解如何将 Blox 组件添加至 portlet。您将:

1. 安装预构建的样本 portlet。通过此步骤, 您可快速了解如何将 Blox 组件添加至 portlet JSP 页以及它在门户网站页上如何显示。
2. 编写您自己的带有 GridBlox 的 JSP 页。

本教程中的任务不提供有关普通 portlet 开发的详细信息。本教程主要介绍特定于 DB2 Alphablox 的任务, 它假定读者在一定程度上了解门户网站环境和 portlet 开发的基本概念。

先决条件

- 必须在 WebSphere Portal V5.1 服务器上安装 DB2 Alphablox。请参阅《安装指南》以了解有关安装的详细信息。
- 必须启动 WebSphere Portal Server。
- 必须对 WebSphere Portal Server 具有管理访问权。
- 必须熟悉 WebSphere Portal 中的管理功能和用户界面。
- 应该具有 Java 和 JSP 的基本知识。
- 应该安装了 JSP 编辑器。

对于本教程, 您可使用任何 JSP 编辑器, 甚至可使用文本编辑器, 但是当您开发自己的 portlet 时, 请使用 WebSphere Portal 建议使用的开发工具, 例如, Rational® Application Developer。

本教程使用与 DB2 Alphablox 一起安装的预定义数据源。您可使用此数据源快速开发基本应用程序。不必考虑为本教程配置定制数据源。

安装样本 portlet

要了解如何将 Blox 添加至 portlet, 最佳方法是安装 DB2 Alphablox 中提供的样本 portlet 并将它们装入到门户网站页。这样您就可以检查 JSP 代码的基本结构并使它与门户网站上的输出匹配。

要安装 DB2 Alphablox 中提供的样本 portlet:

1. 打开浏览器并作为管理用户登录至您的门户网站 (URL 的格式为 `http://<yourPortalServer>:<port>/wps/portal`)。
2. 单击**管理按钮**。
3. 在“Portlet 管理”部分, 单击 **Web 模块**。“管理 Web 模块”页面出现在右边。
4. 单击**安装**。会提示您输入要安装的 Web 模块。
5. 单击**浏览按钮**并浏览至 DB2 Alphablox 安装目录下的 `installableApps` 目录。
6. 选择 `AlphabloxSamplePortlets.war` 并单击**下一步**。在 Portlet 应用程序表中会显示 DB2 Alphablox Sample Portlets 应用程序, 它带有 DB2 Alphablox JSP Page Sample Portlet。

7. 单击完成。

现在已安装了 DB2 Alphablox Sample Portlets 应用程序及其包含的 portlet。请检查 WebSphere Portal 安装目录下的 installedApps 目录。新创建的目录名称以 DB2 Alpha 开头，并以动态生成的 portlet 标识（格式为 _PA_x_x_xx.ear）结尾。

运行样本 portlet

要在门户网站页上运行样本 portlet:

1. 转至门户网站页。
2. 创建或编辑现有页。您可创建新页以测试此样本 portlet，或可单击现有页以编辑。
3. 在门户网站布局页上单击其中一个添加 **Portlet** 按钮。
4. 在“搜索”字段中输入 DB2 并按搜索。会显示 **DB2 Alphablox JSP Page Sample Portlet** 复选框。
5. 选择此复选框并单击**确定**。
6. 单击**完成**。

刷新此门户网站页后，会在该页显示 PresentBlox。此 PresentBlox 具有下列项:

- 位于顶部的菜单栏
- 位于菜单栏下方的两个工具栏面板
- 位于左边的数据布局面板，它使您能够围绕不同的轴来移动维
- 以表格形式显示数据的网格
- 位于右边的三维条形图

此 present.jsp 文件是缺省页，根据样本中包括的 BloxJSPPagePortlet servlet 中的指定内容装入它。可在 WEB-INF/src/ 目录中获取此样本 servlet 的源代码。

您已成功安装了样本 portlet 并已将带有 Blox 的 portlet 添加至门户网站页。下一个任务是检查此 JSP 页中的代码结构。

检查带有 Blox 组件的 portlet JSP 页的结构

在此任务中，您要检查包含 Blox 的 JSP 页的代码结构。所有 JSP 页都必须有相同的键元素。

要打开 JSP 文件:

1. 浏览至 WebSphere Portal 安装目录下的 installedApps/ 目录，找到新创建的以 DB2 Alpha 开头的应用程序文件夹。
2. 浏览至 PA_x_x_xx.war/jsp/html/ 目录。
3. 在 JSP 或 Java 编辑器中打开 present.jsp。
4. 检查下列代码并注意键元素:

```
<%@ page contentType="text/html"%>
<%@ taglib uri="bloxtld" prefix="blox"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<portletAPI:init/>
<%
String bloxName = portletResponse.encodeNamespace("presentBlox");
%>
<head>
<blox:header />
```

```

</head>
<blox:present id="presentBlox" bloxName="<%= bloxName %>" width="800">
<blox:data dataSourceName="canned" />
</blox:present>

```

此代码块包含六个键元素:

1. 第一行告知浏览器输出的是 HTML:

```
<%@ page contentType="text/html"%>
```

2. 下面的一组代码指定此页中使用的两个 JSP 标记库:

```

<%@ taglib uri="bloxtld" prefix="blox"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>

```

uri 是指向标记库描述符文件所在目录位置的指针。定义为 blox 和 portletAPI 的 prefix 值告知服务器:

- 查找此页上任何以 blox 开头的标记, 然后根据标记库描述符文件中的定义, 使用 Blox 标记库来处理内容。
- 查找此页上任何以 portletAPI 开头的标记, 然后根据标记库描述符文件中的定义, 使用 Portlet 标记库来处理内容。

3. 然后添加 portlet 初始化标记:

```
<portletAPI:init/>
```

此标记提供对 PortletRequest、PortletResponse 和 PortletConfig 对象的访问权。使用 PortletResponse, 您可调用 encodeNamespace() 方法, 以确保 Blox 名称不与同一页上运行的其他 portlet 上的对象冲突。

4. 下一个标记对要添加至此页的 Blox 的名称空间进行编码:

```

<%
String bloxName = portletResponse.encodeNamespace("presentBlox"); %>

```

这使您稍后能创建 Blox 并对它指定此唯一名称。

5. 下一个代码块添加 Blox 显示和服务器 / 客户机通信所需的 Blox 头标记:

```

<head>
<blox:header />
</head>

```

DB2 Alphablox 使用此标记以自动将所需的 HTML、JavaScript 和 CSS 代码添加至页的头部分。当服务器显示页时, 此标记会使定义的 HTML 主题和 meta 标记包含 CSS 链接以避免高速缓存。必须将此标记添加至每个包含 Blox 组件的 JSP 页, 否则, 这些组件将不会正常显示。

6. 使用 Blox 标记库中提供的标记来添加 PresentBlox:

```

<blox:present id="presentBlox" bloxName="<%= bloxName %>" width="800">
<blox:data dataSourceName="canned" />
</blox:present>

```

- 此段代码添加 id 为 presentBlox 且 bloxName 为 xx_x_x_xxx_presentBlox (对名称空间进行编码后的值) 的 PresentBlox。
- 此 Blox 的宽度为 800 像素, 高度为 400 像素 (缺省值)。
- 包括嵌套的 DataBlox, dataSourceName 标记属性设置为 canned。

PresentBlox 的 id 标记属性是必需的。它指定要在 JSP 页上使用的 Java 脚本编制名称。bloxName 属性指定服务器上的对象名。编码的 bloxName 确保实例在服务器上是一致的。

注：如果未指定数据源，通常会在网格中看到**没有数据可用**的消息。canned 数据源是在安装期间预定义的，它包含少量的样本数据。它不需要安装和配置实际存在的外部数据库，可使用它来进行故障诊断和学习。由于我们不需要指定查询，所以代码中不存在 query 属性。

此页没有最外面的 <html> 标记，因此 JSP 页显示在带有其他 portlet 的门户网站页上。不需要其他的 <html> 或 <body> 标记。

既然您已检查了 JSP 结构并了解了 portlet JSP 页中包含的主要代码，那么在下一个任务中，您将创建带有不同 Blox 的新 JSP 页，并指定某些公共 Blox 属性以使此页可在门户网站页中更好地显示。

创建您自己的带有 Blox 组件的 portlet JSP 页

在此任务中，您将创建带有 GridBlox 的新 JSP 页并设置它的某些属性。此任务的目的是在创建适合典型门户网站页的 GridBlox 的同时，使您逐步了解一般 Blox 标记构造。

GridBlox 的缺省大小是 400x400 像素。它还带有菜单栏和工具栏。您将设置某些常用的 GridBlox 属性，在同时关闭其菜单栏和工具栏的情况下，此网格的高度仅有 100 像素。通过设置下列 GridBlox 属性可达到此效果：

- height: 设置为 100 像素
- menubarVisible: 设置为 false
- toolbarVisible: 设置为 false

确保您满足了第 11 页的第 2 章，『教程：构建您的第一个带有 Blox 组件的 portlet』中指定的要求，并且按照第 11 页的『安装样本 portlet』中描述的内容安装了样本 portlet 应用程序。

请执行下列步骤：

1. 在显示 PresentBlox 的浏览器窗口中，单击 portlet 的编辑按钮（此按钮带有一个铅笔图标）。会出现一个选择下拉列表。
2. 在此下拉列表中，选择“Grid Blox”并单击**确定**。

在刷新此页之后，您将看到 400x400 的 GridBlox。缺省情况下，此 GridBlox 显示菜单栏和工具栏。在菜单栏和工具栏同时关闭的情况下，其大小将更改为 400x100。

3. 浏览至以前安装的 DB2 Alphablox Sample Portlets 应用程序目录下的 PA_x_x_xx.war/jsp/html/ 目录。
4. 在 JSP 编辑器中打开 grid.jsp。此页看上去与 present.jsp 非常相似，只有下列内容不同：

- <blox:present> 标记现在是 <blox:grid>，并且 id 的值也有所不同：

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400">
<blox:data dataSourceName="canned" /> </blox:grid>
```

- bloxName 的值不同：

```
<%
String bloxName = portletResponse.encodeNamespace("gridBlox"); %>
```

5. 通过添加 height 属性并将其值设置为 100 来将此 GridBlox 的高度设置为 100 像素：

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400" height="100">
<blox:data dataSourceName="canned" /> </blox:grid>
```

6. 通过将 `menubarVisible` 和 `toolbarVisible` 属性设置为 `false` 来关闭顶部的菜单栏和工具栏:

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400" height="100"
menubarVisible="false" toolbarVisible="false">
<blox:data dataSourceName="canned" />
</blox:grid>
```

确保输入的属性名正确，包括大小写（两个属性名中的大写“V”）。还要确保在右尖括号（“>”）之前添加属性。

7. 将名称空间更改为 `myFirstGrid`:

```
<%
String bloxName = portletResponse.encodeNamespace("myFirstGrid");
%>
```

更改此 `GridBlox` 的名称空间是为了确保在您将此 `JSP` 装入门户网站之后，刚才所做的更改能够得到反映。由于您以前已装入了此页，所以对于此会话，在服务器上此 `GridBlox` 已有一个正在运行的实例。除非更改名称空间，否则，即使刷新此页，刚才所做的更改也不会得到反映。更改名称空间是在开发环境中测试更改的快速方法。另外，您也可打开新的浏览器窗口，在服务器上为新会话创建新对象。

8. 保存此文件。

您现在可以在 `WebSphere Portal` 上测试此文件了。

要测试您所做的更改:

1. 返回到门户网站页。
2. 单击浏览器的“刷新”按钮以重新装入此页。

您将看到一个没有菜单栏和工具栏的 `400x100` 的 `GridBlox`。

注: 此 `JSP` 选择下拉列表不是 `WebSphere Portal` 的内置功能，而是由此样本 `portlet` 创建的。检查 `edit.jsp` 文件和 `Java` 源文件是否在 `WEB-INF/src/` 目录中。

创建要使用 `Blox` 组件的 `portlet` 项目

在使用开发工具创建 `portlet` 项目时，请确保将 `Blox` 组件所需的 `AlphabloxServer` `servlet` 映射和标记库引用添加至项目的 `web.xml` 文件中，并将用于 `DB2 Alphablox` 标记库的 `.tld` 文件复制到项目中。

1. 修改项目的 `web.xml` 文件，使其包括下列行:

- 对于 `servlet` 定义和 `servlet` 映射:

```
<servlet>
<servlet-name>AlphabloxServer</servlet-name>
<servlet-class>com.alphablox.server.webapps.server.AlphabloxServer
</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AlphabloxServer</servlet-name>
<url-pattern>/abx/*</url-pattern>
</servlet-mapping>
```

- 对于标记库引用:

```

<taglib>
<taglib-uri>bloxtld</taglib-uri>
<taglib-location>/WEB-INF/tlds/blox.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxformtld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxform.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxlogictld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxlogic.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxreporttld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxreport.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxportlettld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxportlet.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxuitld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxui.tld</taglib-location>
</taglib>

```

2. 将用于 DB2 Alphablox 标记库的 .tld 文件复制到项目的 WEB-INF/tlds/ 目录中。这些文件位于

```
<db2alphablox_dir>/bin/
```

其中，<db2alphablox_dir> 是您的 DB2 Alphablox 安装目录。

使用 Rational Application Developer 配置 portlet 项目

WebSphere Portal 建议您使用 Rational Application Developer 进行 portlet 开发。尤其是 WebSphere Portal V5.1 需要 Rational Application Developer V6.0。Rational Application Developer 提供了向导，可引导您设置和创建 portlet 项目。它根据您的选择设置相应的结构和部署描述符文件，并自动创建用作控制器的 Java 类和用于 portlet 视图的 JSP 页。当您使用“新建 Portlet 项目”向导时，请确保正确配置了下列内容：

- 为目标服务器选择“WebSphere Portal V5.1 存根”。在向导的第一页上，通过单击**显示高级>>**按钮可指定此项。
- 将用于 DB2 Alphablox 的 servlet 映射添加至 WebContent/WEB-INF/ 目录下创建的 web.xml 文件。通过在项目的 web.xml 中输入下列 servlet 定义和映射代码可完成此操作：

```

<servlet>
<servlet-name>AlphabloxServer</servlet-name>
<servlet-class>com.alphablox.server.webapps.server.AlphabloxServer
</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>AlphabloxServer</servlet-name>
<url-pattern>/abx/*</url-pattern>
</servlet-mapping>

```

- 将 Alphablox 标记库添加至项目的 web.xml 中。请执行下列步骤：
 1. 打开 web.xml，单击**变量**选项卡。
 2. 滚动至底部的“标记库引用”部分。
 3. 单击**添加**。

4. 根据下表来输入 **URL** 和**位置**的值:

URL	位置
bloxtld	/WEB-INF/tlds/blox.tld
bloxformtld	/WEB-INF/tlds/bloxform.tld
bloxlogictld	/WEB-INF/tlds/bloxlogic.tld
bloxreporttld	/WEB-INF/tlds/bloxreport.tld
bloxportlettld	/WEB-INF/tlds/bloxportlet.tld
bloxuitld	/WEB-INF/tlds/bloxui.tld

对于每对 URL 和“位置”，都需要单击**完成**来添加标记库引用，然后单击**添加**以添加下一对。

注：有关不同标记库及其用法的详细信息，请参阅“使用 JavaServer 页和 Blox”。

最后，请记住将用于 DB2 Alphablox 标记库的 .tld 文件复制到项目的 WEB-INF/tlds/ 目录中。这些文件位于

<db2alphablox_dir>/bin/

其中，<db2alphablox_dir> 是您的 DB2 Alphablox 安装目录。

后续步骤

您已了解如何创建带有 Blox 组件的 portlet 及其主要代码结构。要显示来自数据库的数据，则需要：

1. 对指向您的数据源的 DB2 Alphablox 定义新数据源。
2. 修改 DataBlox 的 dataSourceName 属性以指向定义的数据源。
3. 将带有查询字符串的 query 属性添加至 DataBlox。

有关创建数据源的详细信息，请参阅《管理员指南》。要了解如何创建适当的查询，请参阅《开发者指南》中的『检索数据』一节以及 *Developer's Reference* 中的 DataBlox 一节。

如果您自己的数据出现在 Blox 视图中，则还可以探究许多可使用 Blox 标记设置的属性。

Portlet 开发技巧

由于您使用 DB2 Alphablox 文档中提供的信息并会继续 portlet 开发，所以需要了解某些特定于 portlet 的主题、概念和一般规则。下表总结了这些一般开发规则，并对文档中您会关注的具体部分提供了指示信息。

- 始终使用 portlet 的名称空间对 Blox 的名称进行编码。名称空间确保当前 J2EE 会话的 Blox 名称是唯一的。
- 始终使用像素来设置 Blox 的宽度和高度。在门户网站环境之外，可将宽度和高度设置为百分比值，例如“50%”或“100%”。百分比值在门户网站环境中不起作用，因为有多个 portlet 共存于此页上。

- 不要使用相对 URL 来调用 portlet 中的资源。应使用 PortletResponse 类的 `encodeURL()` 方法来对 URL 进行编码。在文档中凡是出现要使用相对 URL 的地方，都应对 URL 进行编码。
- 要确保 Blox portlet 使用的主题与门户网站主题相似，请使用 Portal 主题实用程序。此实用程序的来源有两个：
 - DB2 Alphablox 主页。

在**管理**选项卡下，单击**常规**链接。在“门户网站”部分列出了此实用程序。请参阅《**管理员指南**》和联机帮助以获取详细信息。

- AlphabloxAdminPortlets.war 文件。

这是 portlet 版本的 Portal 主题实用程序。您可安装此 portlet，以便通过门户网站运行它而无须专门登录至 DB2 Alphablox 主页。

此实用程序将门户网站环境的样式和 DB2 Alphablox 中的样式组合为一种样式，这就能够使用与此页上其他 portlet 类似的颜色和字体来显示 Blox。

- 检查《**开发者指南**》中的『**Blox Portlet 标记库**』主题，以了解规划阶段的注意事项。
- DB2 Alphablox 具有功能强大的 Blox UI 模型。此 UI 模型包括 ClientLink 对象，它使您在单击页面上的 Blox 组件时装入指定的 URL。对于 portlet 开发，请使用 Blox Portlet 标记库来创建 ClientLink。这些标记将动态处理 URL 信息，以便在刷新页后链接不会失效。请参阅《**开发者指南**》中的『**portlet 开发计划**』主题，以获取详细信息。
- 对于 portlet 和操作 URI，请使用 Blox Portlet 标记库来创建 portlet 链接或操作链接。然后可使用门户网站的 Portlet API 来处理操作或 portlet 链接。请参阅《**开发者指南**》中的『**Blox Portlet 标记库**』主题，以获取详细信息。

第 3 章 教程: 使用 Rational Developer 工具构建应用程序

您可将 Rational Developer 工具与 DB2 Alphablox 工具箱配合使用, 以通过 DB2 Alphablox 组件和 DB2 Alphablox Java API 快速开发应用程序。

可将 Rational Developer 工具 (Rational Application Developer 或 Rational Web Developer) 与 DB2 Alphablox 工具箱 (一组基于 Eclipse 的插件) 配合使用, 以通过 DB2 Alphablox 技术开发应用程序, 并测试这些应用程序。本教程将引导您配置 Rational 集成开发环境, 以便启用方法和标记补全, 以及 DB2 Alphablox 工具箱提供的定制增强功能。

开始将 Rational Developer 工具和 DB2 Alphablox 工具箱配置在一起, 以开发和测试 DB2 Alphablox 应用程序。

准备开发环境

在安装和使用 DB2 Alphablox 工具箱之前, 先要准备好开发环境, 确保已经安装了所有必需的软件。

准备开发环境:

1. 在工作站上安装 Rational Developer 工具 (Rational Application Developer 或 Rational Web Developer)。
2. 安装所需的任何更新, 以确保至少有 Rational Application Developer 或 Rational Web Developer V6.0.0.1。
3. 如果要使用 WebSphere 5.1 集成测试环境, 则确保您已更新了 Rational Developer 版本, 以便包括此可选安装。

注: Rational Developer 工具的缺省安装不包括 WebSphere 5.1 集成测试环境。

您现在可执行本教程中的任务了。

安装 DB2 Alphablox 工具箱

将 DB2 Alphablox 工具箱与 Rational Developer 工具配合使用可使您更易于开发带有 DB2 Alphablox 内容的基于 Web 的应用程序。

DB2

Alphablox 工具箱需要使用 Rational

Application Developer 或 Rational

Web Developer V6.0.0.1。

将 DB2 Alphablox 工具箱安装至 Rational Developer 工具可以省去很多启用 Java 方法和 JSP 定制标记内容补全的手工操作。定制向导和备忘单也能引导您创建带有 DB2 Alphablox 内容的应用程序, 并引导您定义 WebSphere 服务器实例。

要将 DB2 Alphablox 工具箱安装至 Rational Application Developer 或 Rational Web Developer:

1. 在工作站上将 DB2 Alphablox 安装盘插入 CD 驱动器中。
2. 在plugin 目录中找到名为 UpdateSite 的子目录。
3. 将 UpdateSite 目录复制到您硬盘上的适当位置。 例如, 将此目录复制到 C:\DB2Alphablox\UpdateSite。
4. 启动 Rational Developer 工具。
5. 在菜单栏上选择帮助 > 软件更新 > 查找并安装。
6. 在打开的“安装”窗口中选择搜索要安装的新功能选项, 然后单击下一步。
7. 在“要访问的更新站点”窗口中, 单击新的本地站点按钮并浏览至 UpdateSite 目录。
8. 单击下一步并选择 **DB2 Alphablox** 工具箱功能部件, 然后再次单击下一步。
9. 在后续的窗口中, 接受许可协议并选择将用来安装 DB2 Alphablox 工具箱功能部件的目录。
10. 在完成安装之后, 重新启动 Rational Developer 工具

在重新启动 Rational

Developer 工具之后, 将可使用 DB2

Alphablox 工具箱功能部件。

在 WebSphere 集成测试环境中安装 DB2 Alphablox

在此任务中, 您将在 Rational Developer 工具的 WebSphere 集成测试环境中安装 DB2 Alphablox。

先决条件: 必须在 Rational

Developer 工具中安装受支持的 WebSphere

集成测试环境。您需要访问 DB2

Alphablox 安装盘。复查并执行《DB2 Alphablox 安装指南》『预安装任务』一节中所描述的任何所需预安装步骤。

DB2 Alphablox 可安装在单机版的 WebSphere 应用程序服务器上, 或安装在可从 Rational Developer 工具获取的 WebSphere 集成测试环境中。下面概括的几个步骤是安装过程中的差异, 在 WebSphere 运行时 (可在 Rational Developer 工具中获取) 上安装 DB2 Alphablox 时, 必须执行这些步骤。

在 WebSphere 集成测试环境中安装 DB2 Alphablox

安装 DB2 Alphablox 可遵循《DB2 Alphablox 安装指南》『安装』一节中的步骤, 但存在下列差异:

1. 在安装程序的“配置 WebSphere”窗口中, 在 **WebSphere 根目录** 字段中指定想要用作集成测试环境服务器的 WebSphere 运行时的位置。

例如，要在缺省安装时将 DB2 Alphablox 安装至 WebSphere 6 集成测试环境，则选择到 base_v6 目录的路径。对于典型的 Rational Application Developer 安装，其路径与下列路径类似：

```
C:\Program Files\IBM\Rational\SDP\6.0\runtimes\base_v6
```

2. 在“WebSphere 设置”窗口中，指定 WebSphere 管理员的名称和密码。在 WebSphere 集成测试环境中不使用这些输入值，但 DB2 Alphablox 安装程序需要它们。

重要：不要执行《DB2 Alphablox 安装指南》中所描述的安装后步骤。

将 DB2

Alphablox 安装到 WebSphere

运行时之后，可创建 WebSphere

服务器实例，以在 Rational

Developer 工具中测试 DB2

Alphablox 应用程序。

创建 WebSphere 服务器实例

要运行包含了 DB2 Alphablox 内容的应用程序或 JSP 文件，则需要创建可访问所需 DB2 Alphablox 服务和 Java 类的 WebSphere 服务器实例。

先决条件：配置 Rational

Developer 工具。安装 DB2

Alphablox 工具箱。将 DB2

Alphablox 安装至 WebSphere

集成测试环境中。

要创建 WebSphere

服务器实例以测试包含 DB2

Alphablox 内容的应用程序和 JSP 文件：

1. 打开 Rational Developer 工具。
2. 单击**服务器**选项卡（如果看不到此选项卡，则选择**窗口 > 显示视图 > 服务器**）。
3. 在视图窗口中右键单击并选择**新建 > 服务器**。“新建服务器”窗口打开。
4. 输入 localhost 作为主机名并选择服务器类型。单击**下一步**。
5. 输入服务器端口号并单击**下一步**。
6. 将您要在此服务器实例中运行的可用项目添加至已配置项目的列表中。单击**完成**。您的新服务器实例在**服务器**视图中打开。

如果已创建了 WebSphere

5.1 服务器实例，则需要执行下一个步骤“创建 WebSphere

5.1 服务器替换变量”。对于 WebSphere

6 服务器实例，请转至“将来宾添加至 DB2

Alphablox 管理员组”。

创建 WebSphere 5.1 服务器替换变量

在此任务中，您将修改用于 WebSphere 5.1 服务器实例的服务器替换变量，以便在 WebSphere 集成测试环境中正常运行 DB2 Alphablox。

1. 打开**服务器**视图并双击要修改的 WebSphere 5.1 服务器实例。
2. 为此服务器实例单击**配置**选项卡。
3. 选择**启用管理控制台**并取消选择**启用通用测试客户机**。
4. 为此服务器实例单击**变量**选项卡。**替换变量**窗口打开。
5. 使用**添加**按钮（位于**节点设置**列表旁边）添加两个新变量。
 - a. 添加名为 `WS_EAR_AlphabloxPlatform` 的变量并将值设置为 `$(APP_INSTALL_ROOT)/localhost/AlphabloxPlatform.ear`。
 - b. 添加名为 `WS_EAR_AlphabloxStudio` 的第二个变量并将值设置为 `$(APP_INSTALL_ROOT)/localhost/ApplicationStudio.ear`。
6. 保存更改。

对于 WebSphere

5.1 服务器实例，需要遵循“配置 WebSphere

5.1 服务器实例”中的步骤来完成此服务器实例的修改。

配置 WebSphere 5.1 服务器实例

需要配置 WebSphere 5.1 服务器实例以运行 DB2 Alphablox 应用程序和文件。

先决条件：创建所需的 WebSphere

5.1 服务器替换变量。

要配置 WebSphere

5.1 服务器实例：

1. 在 Rational Developer 工具中打开**服务器**视图，并启动要配置的服务器实例。
2. 右键单击此服务器实例并选择**运行管理控制台**。
3. 不要在**标识**字段中输入值。单击**确定**。
4. 单击配置表下的**确定**按钮。
5. 打开**企业应用程序**导航视图并单击 **ApplicationStudio** 应用程序名称。
6. 将**应用程序二进制**设置为 `$(WS_EAR_ApplicationStudio)`。
7. 选择**使用二进制的元数据**。
8. 单击配置表下的**确定**按钮。

9. 在 WebSphere 管理控制台上，单击**保存**按钮。
10. 在“保存至主配置”窗口中，单击**保存**。
11. 在 WebSphere 管理控制台中，通过选择**应用程序 > 企业应用程序**来启动 AlphabloxPlatform 和 ApplicationStudio 应用程序。

服务器实例已配置。下一步需要执行步骤“将来宾添加至 DB2

Alphablox 管理员组”。

将来宾用户添加至 DB2 Alphablox 管理员组

通过将来宾用户添加至 DB2 Alphablox 管理员组，可以使您在使用 Rational Developer 工具时访问“DB2 Alphablox 管理页面”。

先决条件: 创建 WebSphere

服务器实例。配置 WebSphere

服务器实例。

要将来宾用户添加至 DB2

Alphablox 管理员组:

重要: 不要在您的生产 WebSphere 服务器上对来宾用户授予管理员权限。

1. 通过使用下列 Telnet 命令登录至 DB2 Alphablox 控制台: `telnet localhost portNumber`。其中, `portNumber` 是安装 DB2 Alphablox 时指定的端口。
2. 在 Telnet 控制台提示符处输入下列 DB2 Alphablox 控制台命令: `set Administrators guest`, 然后按 Enter 键。
3. 在 Telnet 控制台中, 输入 `save`, 然后按 Enter 键。
4. 关闭 Telnet 会话。

现在可使用您的 WebSphere

服务器实例了。

创建 DB2 Alphablox 应用程序

当您在 Rational Developer 中创建新应用程序时，需要添加所需的 DB2 Alphablox 内容，以使应用程序能正常运行。

先决条件: 在 Rational

Developer 工具上安装 DB2

Alphablox 工具箱。

1. 在 Rational Developer 工具菜单栏上选择**文件 > 新建项目**。新建项目向导打开。
2. 展开 **Web** 选项，选择**动态 Web 项目**，然后单击**下一步**。
3. 输入项目名称并单击**显示高级按钮**。会出现其他选项。
4. 选择合适的 servlet 版本和目标服务器。

5. 单击下一步。“功能选项”窗口打开。
6. 选择 **DB2 Alphablox** 内容选项，然后单击完成。

您的新 Web 应用程序项目现在能够使用 DB2

Alphablox 了。 DB2

Alphablox 标记库和 Blox Java

API 现在可用，并且修改了应用程序描述符文件（web.xml），使其包括所需的 DB2

Alphablox 信息。您现在可将包含 DB2

Alphablox 内容的 JSP 文件添加至项目。

创建带有 DB2 Alphablox 内容的 JSP 文件

在此任务中，通过将 Rational Developer 工具与 DB2 Alphablox 工具箱配合使用，您将创建可访问 Blox 标记库的新 JSP 文件。

先决条件: 在 Rational

Developer 工具上安装 DB2

Alphablox 工具箱。

1. 在 Rational Developer 的项目资源管理器视图中，选择文件 > 新建 > **JSP** 文件。
2. 在文件名字段中输入文件名。
3. 单击配置高级选项并按下一步。
4. 添加将在 JSP 文件中使用的 DB2 Alphablox 标记库。
 - a. 单击添加按钮以打开“添加标记库”窗口。
 - b. 选择要在 JSP 文件中使用的 DB2 Alphablox 标记库。
 - c. 单击下一步。
5. 从编码列表中选择 **ISO 10646/Unicode(UTF-8)**。 DB2 Alphablox 应用程序需要 UTF-8 编码以正常运行。
6. 单击完成。新的 JSP 文件会出现在项目列表中。
7. 双击此文件名以在 JSP 编辑器窗口中打开此文件。
8. 将光标放置在 HTML <head> 标记内，但要放置在先前步骤中添加的 DB2 Alphablox JSP taglib 伪指令之后。
9. 在新行上输入下列 Blox 头标记: <blox:header/>。在运行 JSP 文件时，需要 Blox 头标记来添加所需的 DB2 Alphablox JavaScript 和 CSS 文件。

您的新 JSP 文件现在已经能够访问所选的 DB2

Alphablox 标记库了。通过使用 Rational

Developer 工具的内容辅助功能，您可插入所选标记库的 Blox 标记和标记属性。

访问 DB2 OLAP Server 和 Essbase 数据源

要在使用 WebSphere 5.1 服务器实例时，在 Rational Developer 工具中访问 DB2 OLAP Server™ 或 Essbase 数据源，则需要装入所需客户机库的启动批处理文件。

先决条件: 安装 DB2

Alphablox 工具箱。将 DB2

Alphablox 安装至 WebSphere

5.1 集成测试环境。配置 WebSphere

5.1 服务器实例。在开发机器上安装 IBM®DB2 OLAP Server

或 Hyperion Essbase。

由于 Rational Developer 工具的 WebSphere 5.1 集成测试环境中存在局限性，所以需要创建用于 Rational Developer 工具的启动批处理文件，以访问 DB2 OLAP Server 或 Hyperion Essbase 数据源。当您使用此批处理文件启动 Rational Developer 工具时，所需的 Essbase 客户机库被添加至 Java 库路径中。

要创建批处理文件以使用 WebSphere

5.1 服务器实例来访问 DB2 OLAP Server

或 Essbase:

1. 通过文本编辑器创建新的文本文档。
2. 添加一行代码以调用 DB2 Alphablox aassetup.bat 文件，它位于您的 DB2 Alphablox 安装目录中。例如，下列代码将运行位于指定 DB2 Alphablox 安装目录中的 aassetup.bat 文件:

```
call C:\alphablox\analytics\bin\aassetup.bat
```

3. 添加第二行代码以调用 RAD rationalsdp.exe 文件，它将启动 RAD。例如，下列代码将运行 Rational Developer 工具:

```
call C:\Program Files\IBM\Rational\SDP\6.0\rationalsdp.exe
```

4. 将此文件另存为 startRAD.bat 并放到您的工作站桌面上（或其他方便位置）。

当您双击 startRAD.bat 文件时，DB2

Alphablox aassetup.bat 会运行并设置所需路径和环境变量，然后启动 Rational

Application Developer。与 DB2

Alphablox 配合使用的 Essbase 客户机库版本需要与用作数据源的 DB2 OLAP Server（或 Hyperion Essbase）版本匹配。要修改与 DB2

Alphablox 配合使用的 Essbase 客户机库，运行 DB2 OLAP Server

/Essbase 客户机库实用程序（ChangeEssbase.bat），可在 db2_alphablox\analytics\bin 目录下找到它，其中，db2_alphablox 是 DB2

Alphablox 安装目录的根目录。此实用程序修改启动 RAD 时所运行的某个批处理文件，启动 RAD 时使用已创建的 startRAD 批处理文件。

下面是上述 startRAD.bat 文件的完整示例：

```
call C:\alphablox\analytics\bin\assetup.bat  
call C:\Program Files\IBM\Rational\SDP\6.0\rationalsdp.exe
```

无论您何时要访问 DB2 OLAP Server

或 Essbase 数据源，都必须使用 startupRAD.bat 文件来启动 Rational

Developer 工具。

第 4 章 教程: 使用 DB2 Cube Views 构建 DB2 Alphablox 立方体

本教程将引导您使用 DB2 Cube Views 样本数据库来创建 DB2 Alphablox 立方体。

本教程中的任务不提供有关构建定制 DB2 Alphablox 立方体的详细信息。其目的是使您了解如何快速创建 DB2 Alphablox 立方体，从而了解 DB2 Alphablox Cube Server 的功能。生成的数据源还可用于测试和构建 DB2 Alphablox 应用程序。

先决条件:

- 安装 DB2 Alphablox。请参阅《安装指南》以了解有关安装的详细信息。
- 在安装了 DB2 Cube Views CVSAMPLE 样本数据库的情况下，获取对所支持的 DB2 Cubes Views 的访问权。请参阅《安装指南》以了解有关所支持的 DB2 Cube Views 版本的详细信息。

在本教程中，您将了解如何基于 DB2 Cube Views CVSAMPLE 样本数据库来构建 DB2 Alphablox 立方体。在本教程中，您将完成下列任务:

定义 DB2 关系数据源

在此任务中，您将在 DB2 Alphablox 中把某个 DB2 数据库定义为数据源。

先决条件: 所需的 DB2

JDBC 驱动程序必须可供 DB2

Alphablox 访问。

DB2 Alphablox 立方体要求将底层关系数据源预定义为 DB2 Alphablox 数据源。使用关系数据库中可获取的元数据和数据生成 DB2 Alphablox 立方体。

要将某个 DB2 数据库定义为 DB2 Alphablox 数据源:

1. 作为 admin 用户（或者作为拥有管理员权限的用户）登录至“DB2 Alphablox 管理页面”。
2. 单击**管理**选项卡。
3. 单击**数据源**链接。
4. 单击**创建**按钮。
5. 从**适配器**菜单选择用于数据库服务器的合适的 IBM DB2 JDBC 驱动程序。

选择 **IBM DB2 JDBC Type 4 Driver** 或 **IBM DB2 UDB on iSeries Driver**。

6. 在**数据源名称**字段中输入 CVSAMPLE 作为您的数据源的名称。
7. 对**服务器名称**、**端口号**和**数据库名称**（其值应该是 CVSAMPLE）输入合适的值。

注: 如果您需要帮助确定这些字段的值是否正确，则与您的数据库管理员联系。

8. 输入缺省用户名和缺省秘密。

用户名和密码必须对该关系数据库有效。当 DB2 Alphablox 立方体访问关系数据库时，总是使用缺省用户名和密码。您使用的数据库用户应具有对此数据库的读访问权。

注：当使用指定的关系数据源来填充 DB2 Alphablox 立方体时，将忽略使用 **DB2 Alphablox 用户名和密码** 的值。访问控制表（ACL）可用于限制对 DB2 Alphablox 立方体的访问。有关 ACL 的信息，请参阅《管理员指南》。

9. 当使用此数据源来填充 DB2 Alphablox 立方体时，忽略**最大行数和最大列数**的值。您仍可输入这些值，并且其他应用程序使用此数据源时将使用它们。
10. 除非您想将 JDBC 日志记录信息写至 DB2 Alphablox 日志文件，否则，请将**启用 JDBC 跟踪**值设置为**否**。仅当您遇到问题并且需要调试问题原因时，才启用 JDBC 跟踪。
11. 单击**保存按钮**以保存数据源定义。

现在，您已将 DB2

Alphablox 数据源定义为 CVSAMPLE。您现在可创建 DB2

Alphablox 立方体定义，以访问此 DB2

数据源中的 Cube Views™

立方体元数据。

定义 Alphablox Cube Server Adaptor 数据源

在此任务中，您将定义使用 Alphablox Cube Server Adapter 的 DB2 Alphablox 数据源定义。

先决条件：将 DB2

CVSAMPLE 数据库作为 DB2 Alphablox 数据源。

要定义 DB2

Alphablox Cube Server Adapter 数据源：

1. 作为 admin 用户（或者作为拥有管理员权限的用户）登录至“DB2 Alphablox 管理页面”。
2. 单击**管理**选项卡。
3. 单击**数据源**链接。
4. 单击**创建按钮**。
5. 从**适配器**菜单中选择 **Alphablox Cube Server Adapter** 选项。
6. 在**数据源名称**字段中输入 DB2AlphabloxCubes 作为您的数据源的名称。
7. 单击**保存按钮**以保存数据源定义。

您已定义了可用来访问 DB2 Alphablox 立方体的 DB2 Alphablox 数据源。下一步需要定义您可访问的 DB2 Alphablox 立方体。

定义 DB2 Alphablox 立方体

在此任务中，您将基于 DB2 Cube Views CVSAMPLE 立方体中的元数据来定义 DB2 Alphablox 立方体。

先决条件： 定义 DB2 关系数据源。定义 DB2 Alphablox Cube Server Adapter 数据源。

要定义 DB2

Alphablox 立方体的一般属性：

1. 作为 admin 用户（或者作为拥有管理员权限的用户）登录至“DB2 Alphablox 管理页面”。
2. 单击**管理**选项卡。
3. 单击**立方体**链接。
4. 单击**创建**按钮。“立方体管理”窗口打开。
5. 定义新的立方体：
 - a. 在 **DB2 Alphablox 立方体名称** 字段中输入 CVSales。
 - b. 选择 **DB2 Alphablox 立方体名称** 字段旁边的**启用**选项。选择此选项会使您的立方体在服务器重新启动时自动启动。
 - c. 从**关系数据源**菜单中选择您为本教程创建的关系数据源 MyDB2。
 - d. 不要选择**安全角色**选项。此选项可用于限制用户访问某个特定的立方体。
6. 启用 DB2 Cube Views 设置并指定要使用的元数据：
 - a. 选择**启用 DB2 Cube Views**设置选项。
 - b. 从**立方体模型**菜单中选择 **CVSAMPLE.Sales**。
 - c. 从**立方体**菜单中选择**一般销售**立方体。
 - d. 选择**使用业务名**单选按钮以指定名称。选择此选项之后，通常会使用更有意义及更易读的成员名。
 - e. 单击**导入立方体定义**按钮。通过使用此选项，您可在 DB2 Alphablox 立方体中导入立方体定义并预填充量度和维。导入的立方体定义基于 DB2 Cube Views 元数据的支持，能够反映与 DB2 Alphablox 最匹配的 DB2 Cube Views 立方体。当您在使用立方体方面积累了更多经验时，可修改量度和维以满足您的需求。
 - f. 单击**显示导入日志**按钮，以查看与导入操作相关的日志指定信息和调试消息。对于本教程，执行此步骤仅仅是为了帮助您熟悉此功能。
 - g. 选择**在启动、重建和编辑时导入立方体定义**选项。此选项将使 DB2 Alphablox 立方体在每次启动、重建或打开以进行编辑时，都使用最新的 DB2 Cube Views 定义。在更了解 DB2 Alphablox 和 DB2 Cube Views 之后，您可导入立方体定义并定制它。
7. 单击**保存**按钮以保存 DB2 Alphablox 立方体定义。

您现在已创建了 DB2 Alphablox 立方体定义。您可启动新的 CVSales 立方体了。

启动 DB2 Alphablox 立方体

在此任务中，将使用“DB2 Alphablox 管理页面”启动 CVSales 立方体。

先决条件: 定义 DB2 关系数据源。定义 DB2 Alphablox Cube Server Adapter 数据源。
定义 DB2 Alphablox 立方体。

要启动 CVSales 立方体:

1. 作为 admin 用户（或者作为拥有管理员权限的用户）登录至“DB2 Alphablox 管理页面”。
2. 单击**管理**选项卡。
3. 在**运行时管理**部分下面，单击**立方体**链接。
4. 从 **DB2 Alphablox 立方体**列表中选择要启动的 DB2 Alphablox 立方体。
5. 单击**启动**按钮。 在启动了立方体之后，状态字段会显示**正在运行**。

您现在已有一个正在运行的样本 DB2 Alphablox 立方体。现在您可使用刚才创建的 DB2 Alphablox 立方体来构建应用程序。您可通过以下方法对新立方体执行快速检查: 使用 Query Builder 应用程序，选择您的立方体作为数据源并对此立方体运行 MDX 查询。

声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

*IBM Director of Licensing,
IBM Corporation,
North Castle Drive, Armonk, NY 10504-1785
U.S.A.*

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，或用书面方式将查询寄往：

*IBM World Trade Asia Corporation, Licensing,
2-31 Roppongi 3-chome, Minato-ku,
Tokyo 106-0032, Japan*

本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：International Business Machines Corporation “按现状” 提供本出版物，不附有任何种类的（无论是明示的还是暗含的）保证，包括但不限于暗含的有关非侵权、适销和适用于某种特定用途的保证。某些国家或地区在某些交易中不允许免除明示或暗含的保证。因此本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

本信息中对非 IBM Web 站点的任何引用都只是为了方便起见才提供的，不以任何方式充当对那些 Web 站点的保证。那些 Web 站点中的资料不是 IBM 产品资料的一部分，使用那些 Web 站点带来的风险将由您自行承担。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（i）允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation,
J46A/G4, 555 Bailey Avenue,
San Jose, CA 95141-1003
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本文中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际软件许可协议或任何同等协议中的条款提供。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可随时更改或收回，而不另行通知，它们仅仅表示了目标和意愿而已。

本信息包含在日常业务操作中使用的数据和报告的示例。为了尽可能完整地说明这些示例，示例中可能会包括个人、公司、品牌和产品的名称。所有这些名称都是虚构的，与实际商业企业所用的名称和地址的任何雷同纯属巧合。

本信息包括源语言形式的样本应用程序，这些样本说明不同操作平台上的编程方法。如果是为按照在编写样本程序的操作平台上的应用程序编程接口（API）进行应用程序的开发、使用、经销或分发为目的，您可以任何形式对这些样本程序进行复制、修改、分发，而无须向 IBM 付费。这些示例并未在所有条件下作全面测试。因此，IBM 不能担保或暗示这些程序的可靠性、可维护性或功能。用户如果是为了按照 IBM 应用程序编程接口开发、使用、经销或分发应用程序，则可以任何形式复制、修改和分发这些样本程序，而无须向 IBM 付费。

商标

下列各项是 International Business Machines Corporation 在美国和 / 或其他国家或地区的商标或注册商标:

DB2	DB2 OLAP Server	DB2 Universal Database
IBM	WebSphere	

Alphablox 和 Blox 是 Alphablox Corporation 在美国和 / 或其他国家或地区的商标或注册商标。

Java 和所有基于 Java 的商标是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

Linux[®] 是 Linus Torvalds 在美国和 / 或其他国家或地区的商标。

其他公司、产品或服务名称可能是其他公司的商标或服务标记。

索引

[C]

查询

 使用 Query Builder 生成 3

[J]

教程

 应用程序开发 1

P

portlet

 样本 11

 JSP 结构 12

Q

Query Builder

 使用 3

Query Builder, DHTML

 使用 3



程序号: 5724-L14

中国印刷

G151-0337-00

