

IBM DB2 Alphablox



Erste Schritte

Version 84

IBM DB2 Alphablox



Erste Schritte

Version 84

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter „Bemerkungen“ auf Seite 35 gelesen werden.

Zweite Ausgabe (März 2006)

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 Alphablox Getting Started,
IBM Form GC18-9607-01,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2006
© Copyright IBM Deutschland GmbH 2006

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
März 2006

Inhaltsverzeichnis

Kapitel 1. Lerntext: Erzeugen Ihrer ersten Anwendung 1

Definieren Ihrer Anwendung	1
Zugreifen auf Daten	2
Erstellen Ihrer Anwendungshomepage.	2
Festlegen einer Standardhomepage	3
Erstellen Ihrer ersten analytischen Sicht	4
Struktur der Seite Ihrer ersten analytischen Sicht	7
Erstellen Ihrer zweiten analytischen Sicht.	9
Zusammenfassung	10

Kapitel 2. Lerntext: Erzeugen Ihres ersten Portlets mit Blox-Komponenten . . . 13

Installieren der Musterportlets	13
Ausführen der Musterportlets	14
Untersuchen der Struktur einer Portlet-JSP-Seite mit einer Blox-Komponente	15
Erstellen Ihrer eigenen Portlet-JSP-Seite mit Blox-Komponenten	17
Erstellen eines Portletprojekts zum Verwenden von Blox-Komponenten	18
Konfigurieren eines Portletprojekts mit Rational Application Developer.	19
Weitere Schritte	20
Tipps zur Portletentwicklung	21

Kapitel 3. Lerntext: Erzeugen von Anwendungen mit den Rational Developer-Tools. 23

Vorbereiten Ihrer Entwicklungsumgebung	23
Installieren des DB2 Alphablox-Toolkits	23
Installieren von DB2 Alphablox in integrierten WebSphere-Testumgebungen	24
Erstellen einer WebSphere-Serverinstanz.	25
Erstellen der Serversubstitutionsvariablen von WebSphere 5.1	26
Konfigurieren der Serverinstanzen von WebSphere 5.1	26
Hinzufügen des Gastbenutzers zur Gruppe der Administratoren von DB2 Alphablox	27
Erstellen von DB2 Alphablox-Anwendungen	27
Erstellen von JSP-Dateien mit DB2 Alphablox-Inhalt	28
Zugreifen auf DB2 OLAP Server und Essbase-Datenquellen	28

Kapitel 4. Lerntext: Erzeugen eines DB2 Alphablox-Würfels mit DB2 Cube Views 31

Definieren einer relationalen DB2-Datenquelle.	31
Definieren einer Alphablox Cube Server-Adapterdatenquelle	32
Definieren eines DB2 Alphablox-Würfels	33
Starten Ihres DB2 Alphablox-Würfels	34

Bemerkungen. 35

Marken.	37
-----------------	----

Index 39

Kapitel 1. Lerntext: Erzeugen Ihrer ersten Anwendung

Im vorliegenden Lerntext erhalten Sie wichtige grundlegende Informationen zu Blox-Komponenten. Mit Hilfe der hier aufgeführten Schritte können Sie schnell eine DB2 Alphablox-Anwendung erzeugen.

Zum Erzeugen von DB2 Alphablox-Anwendungen werden Sie JSP-Technologie (JSP - JavaServer Pages) verwenden. Zum Ausführen der in diesem Lerntext beschriebenen Aktionen sind jedoch keine JSP-Kenntnisse erforderlich. In diesem Lerntext werden die Aspekte der JSP-Technologie erläutert, mit denen Sie vertraut sein müssen, um mit dem Erzeugen von analytischen Anwendungen beginnen zu können. Es gibt viele hervorragende Bücher, in denen Sie mehr über die JSP-Technologie erfahren.

Voraussetzungen

In den Schritten dieses Lerntexts wird davon ausgegangen, dass DB2 Alphablox unter WebSphere Application Server ausgeführt wird. Wenn Sie BEA WebLogic oder Apache Tomcat als Anwendungsserver verwenden, müssen Sie unter Umständen einige der folgenden Schritte im Hinblick auf die jeweiligen abweichenden Merkmale Ihres Servers modifizieren.

Außerdem müssen Sie einen DB2 Alphablox-Würfel (qcc_2003) und die zugehörige zu Grunde liegende relationale Datenquelle (qcc2003-rdb) installieren, bevor Sie diesen Lerntext durcharbeiten. Die Musterdatenbankdateien und Anweisungen (in den Dateien mit dem Namen **readme.txt**) zum Konfigurieren des DB2 Alphablox-Würfels und der relationalen Datenbank, die für diesen Lerntext erforderlich sind, finden Sie auf der DB2 Alphablox-Installations-CD im Verzeichnis **sampledata/qcc/acs**.

Definieren Ihrer Anwendung

Zum Erstellen einer Anwendung mit der J2EE-Entwicklungsmethode müssen Sie eine Verzeichnisstruktur mit einem Verzeichnis WEB-INF erstellen, in dem eine Anwendungsdeskriptordatei (web.xml) enthalten ist. Sie erstellen diese Struktur in DB2 Alphablox am einfachsten, indem Sie mit Hilfe der Seite **Applications** in den DB2 Alphablox-Verwaltungsseiten eine neue Anwendung erstellen.

Gehen Sie wie folgt vor, um Ihre Anwendung und Ihren Ordner zu erstellen:

1. Erstellen Sie eine neue Anwendung namens **MeineAnw** in DB2 Alphablox, indem Sie die Schritte ausführen, die im Abschnitt über Anwendungsdefinitionen des Handbuchs *Administrator's Guide* beschrieben werden.
 - Geben Sie **MeineAnw** in das Feld **Name** ein.
 - Geben Sie **Meine Anw** (mit einem Leerzeichen) in das Feld **Display Name** ein. Der Anzeigename legt die Bezeichnung der Anwendung fest, die in der Liste auf der Seite **Applications** angezeigt wird.
2. Klicken Sie die Indexzunge **Applications** ganz oben links auf der Seite an. Eine Liste der verfügbaren Anwendungen wird angezeigt.
3. Klicken Sie den Namen Ihrer neu erstellten Anwendung (**Meine Anw**) in der Liste der Anwendungen an. Da Sie noch keine Dateien erstellt haben, ist das Dateienverzeichnis leer.

4. Navigieren Sie zum neuen Anwendungsordner Ihres Anwendungsservers (bei WebSphere befindet sich dieser Ordner im WebSphere-Verzeichnis `installedApps`). Beachten Sie, dass die Anwendung ein Verzeichnis `WEB-INF` mit einer Datei `web.xml` und dem Verzeichnis `tlds` enthält. Die Datei `web.xml` definiert die Anwendungsinformationen. Das Verzeichnis `tlds` enthält die Deskriptordatei der Blox-Tagbibliothek (`blox.tld`), die die Blox-Tags definiert, mit denen Sie Ihre analytischen Sichten in diesem Lerntext erstellen werden. Das Verzeichnis enthält weitere TLD-Dateien für andere Blox-Tagbibliotheken, die nicht in diesem Lerntext verwendet werden. Dazu gehören die Dateien `bloxform.tld`, `bloxlogic.tld`, `bloxreport.tld` und `bloxui.tld`.

Sie haben ein DB2 Alphablox-Anwendungsframework erstellt, dem jetzt analytische Sichten hinzugefügt werden können.

Zugreifen auf Daten

In diesem Lerntext werden Sie die Datenquelle des DB2 Alphablox-Würfels verwenden, die Sie erstellt haben, bevor Sie mit dem Durcharbeiten dieses Lerntexts begonnen haben. Ihre relationale Datenbank und Alphablox Cube Server müssen aktiv sein. Der verwendete DB2 Alphablox-Würfel muss ebenfalls aktiv sein.

Gehen Sie wie folgt vor, um zu überprüfen, ob Ihre erforderlichen Datenquellen verfügbar und betriebsbereit sind:

1. Öffnen Sie die DB2 Alphablox-Verwaltungsseiten in Ihrem Browser.
2. Klicken Sie die Indexzunge **Administration** an.
3. Klicken Sie **Data Sources** an.
4. Suchen Sie auf der Seite **Data Sources** nach der Datenquelle **qcc2003-rdb** in dem Menü, das sich links auf der Seite befindet. Wenn Sie sie dort nicht finden, müssen Sie die Schritte in der Datei `readme.txt` im Verzeichnis `sampledata/qcc/acs` Ihrer DB2 Alphablox-Installations-CD überprüfen.
5. Klicken Sie **qcc2003-rdb** in der Liste der verfügbaren Datenquellen an, und klicken Sie dann den Knopf **Test Selected Data Source** an. Wenn die Datenbank aktiv ist und Ihre Datenquellendefinition ordnungsgemäß konfiguriert ist, wird eine Nachricht über die erfolgreiche Ausführung angezeigt. Wenn eine Fehlermeldung angezeigt wird, müssen Sie die Datenquellendefinition überprüfen, indem Sie die Datenquelle auswählen und den Knopf **Edit** anklicken, oder Sie müssen überprüfen, ob die Datenbank ordnungsgemäß mit einem Benutzer konfiguriert ist, der über Zugriffsberechtigungen für die Datenbank verfügt.
6. Wählen Sie die Datenquelle **qcc2003-acs** aus der Liste der Datenquellen aus, und klicken Sie dann den Knopf **Test Selected Data Source** an. Wenn die Alphablox Cube Server-Datenquelle richtig konfiguriert ist, wird eine Nachricht über die erfolgreiche Ausführung angezeigt. Wenn eine Fehlermeldung angezeigt wird, überprüfen Sie die Datenquellendefinition.

Nachdem Sie die Datenquellen erfolgreich getestet haben, können Sie das Erzeugen von zwei einfachen analytischen DB2 Alphablox-Sichten erlernen.

Erstellen Ihrer Anwendungshomepage

Diese Task beinhaltet das Erstellen einer einfachen Homepage mit grundlegendem Web-Know-how.

Für die Zwecke dieses Lerntexts muss nur eine sehr einfache Webanwendungsstruktur erstellt werden. Sie müssen auf der Homepage zwei Links für den Zugriff auf die zwei analytischen Sichten angeben, die Sie in diesem Lerntext erstellen werden.

Gehen Sie wie folgt vor, um eine einfache Homepage zu erstellen:

1. Geben Sie der Homepage mit Hilfe eines Texteditors oder Ihres bevorzugten Entwicklungstools den Namen `index.html`. Für die Zwecke dieses Lerntexts reicht es aus, wenn Sie eine sehr einfache Homepage erstellen.
2. Bearbeiten Sie die Datei, um einen Anwendungstitel (z. B. "My DB2 Alphablox Application") und zwei Links zu den beiden analytischen Sichten anzugeben, die Sie erstellen werden.

Kopieren Sie den folgenden Code, und fügen Sie ihn in die Datei **index.html** ein, bzw. geben Sie ihn dort ein.

```
<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. Speichern Sie diese Datei im Verzeichnis `MeineAnw`, das Sie bei der Definition der Anwendung erstellt haben.
4. Öffnen Sie die DB2 Alphablox-Homepage in Ihrem Web-Browser. Der Browser zeigt standardmäßig die Indexzeile **Applications** an.
5. Klicken Sie den Link **Meine Anw** an, um zu prüfen, ob die Datei in der Verzeichnisliste enthalten ist. Klicken Sie den Link **index.html** an, um Ihre Homepage anzuzeigen. Wenn DB2 Alphablox auf einem System mit IBM WebSphere Application Server oder auf einem BEA WebLogic-Server installiert ist, müssen die Verzeichnislisten aktiviert werden, bevor Sie diesen Schritt ausführen können. Bei DB2 Alphablox-Installationen unter Apache Tomcat sollten keine weiteren Schritte erforderlich sein. Wenn der Fehler HTTP 403 (Verzeichnislisten sind standardmäßig nicht zulässig) angezeigt wird, können Sie entweder den Server modifizieren, um Verzeichnislisten zuzulassen, oder Sie hängen einfach den Dateinamen `meineanw.html` am Ende der URL-Adresse an (z. B. `http://localhost:9080/MeineAnw/meineanw.html`). Die neue Seite sollte angezeigt werden, wenn Sie die vollständig qualifizierte Verbindungsadresse verwenden.

Sie verfügen jetzt über eine Homepage mit Links zu den zwei analytischen Sichten, die Sie als Nächstes erstellen werden.

Festlegen einer Standardhomepage

In den vorangegangenen Schritten haben Sie direkt auf die Homepage zugegriffen, indem Sie den Dateinamenlink im Anwendungsverzeichnis `MeineAnw` angeklickt oder die Homepagedatei `index.html` in der URL-Adresse angegeben haben. Mit Hilfe der folgenden Schritte geben Sie die Standardhomepage für die Anwendung an.

Gehen Sie wie folgt vor, um die Standardhomepage für die Anwendung festzulegen:

1. Öffnen Sie die DB2 Alphablox-Homepage in Ihrem Browser.
2. Klicken Sie die Indexzunge **Administration** an.
3. Klicken Sie den Link **Applications** an.
4. Wählen Sie **MeineAnw** in der Anwendungsliste aus, und klicken Sie den Knopf **Edit** unterhalb der Liste an.
5. Geben Sie auf der Seite **Edit Application** für MeineAnw die Datei `index.html` in das Feld **Home URL** ein.
6. Klicken Sie den Knopf **Save** an, um die vorgenommene Änderung zu speichern.
7. Klicken Sie die Hauptindexzunge **Applications** (die obere linke Ordnerindexzunge) an, um zur Seite **Applications** zurückzukehren.
8. Klicken Sie den Anwendungsnamen **Meine Anw** an. Die Anwendung wird nun direkt auf der definierten Homepage geöffnet.

Durch die Definition der Homepage weiß der Server jetzt, wo er die Anwendung starten muss. Der Server wird die Anwendung so steuern, dass sie auf dieser Seite geöffnet wird, auch wenn der Benutzer die Homepage nicht in der URL-Adresse angegeben hat. Wenn ein Benutzer also `http://ihr_servername/MeineAnw/` in die Browseradressleiste eingibt, wird die Anwendung automatisch auf der definierten Homepage `index.html` geöffnet.

In den folgenden Themen beginnen Sie mit der Erstellung analytischer DB2 Alphablox-Sichten unter Verwendung von einfachen Blox-Tags, die mit der Anwendung DB2 Alphablox Query Builder generiert werden.

Erstellen Ihrer ersten analytischen Sicht

Diese Task beinhaltet das Erstellen einer analytischen DB2 Alphablox-Sicht, die ein Raster und ein Diagramm als Kombination in der PresentBlox-Komponente anzeigt.

Voraussetzung: Lesen Sie die Abschnitte zur Einführung im Handbuch *Developer's Guide*, um mehr über DB2 Alphablox und die zugehörigen Komponenten zu erfahren. Dadurch erhalten Sie ein Grundverständnis von der DB2 Alphablox-Funktionalität und erfahren, wie die Bibliotheken mit angepassten Blox-Tags beim Erzeugen von analytischen Anwendungen mit JSP-Technologie verwendet werden können.

Für die erste analytische Sicht, die Sie erstellen, werden Sie die Anwendung DB2 Alphablox Query Builder verwenden, um einen angepassten JSP-Tag zu generieren, der für eine PresentBlox-Komponente definiert ist. Dies ist die am häufigsten verwendete Blox-Komponente zum Darstellen von Daten. Eine PresentBlox-Komponente kann Raster- und Diagrammsichten, eine Datenlayoutanzeige, Funktionsleisten und andere verschachtelte Komponenten enthalten. Die Raster- und Diagrammsichten sind synchronisierte alternative Sichten derselben Daten.

Gehen Sie wie folgt vor, um Ihre erste analytische Sicht mit Query Builder zu erstellen:

1. Öffnen Sie die DB2 Alphablox-Verwaltungsseiten in einem Browser. Standardmäßig wird die Indexzunge **Applications** angezeigt.
2. Klicken Sie **DB2 Alphablox Query Builder** zum Öffnen der Anwendung an.

3. Klicken Sie den Knopf **Verbindungseinstellungen** an. Das Fenster **Datenbankverbindung** wird angezeigt.
4. Wählen Sie die Datenquelle **qcc2003-acs** aus, und drücken Sie dann den Knopf **Verbindung herstellen**. Das Fenster wird geschlossen und im Abschnitt **Datenbankstatus** der Seite **Query Builder** sollte "Verbunden" angezeigt werden.

Anmerkung: Sie müssen keine Werte in die Felder **Schema**, **Katalog**, **Benutzername** und **Kennwort** eingeben, da diese automatisch aus der Datenquellendefinition für die von Ihnen ausgewählte Datenquelle abgerufen werden. Wählen Sie außerdem das Markierungsfeld **Standardabfrage ausführen** nicht aus.

5. Suchen Sie den Abschnitt **Abfrage** auf der Seite unterhalb des Abschnitts **Status**. Beachten Sie die folgenden Aspekte:
 - Sie verwenden für diesen Lerntext einen DB2 Alphablox-Würfel (qcc2003-acs). Direkt unterhalb der Abschnittsüberschrift **Abfrage** sollte eine Liste mit Würfeln angezeigt werden, die verfügbar (momentan aktiv) sind. Der Würfel **qcc2003-acs** sollte aufgelistet sein.
 - Im Textfenster wird eine Standardabfragezeichenfolge angezeigt, die dieser Datenquelle zugeordnet ist. Da die Datenquelle ein DB2 Alphablox-Würfel ist, wird der erste Würfel in der alphabetischen Auflistung der Abfrageanweisung hinzugefügt. Beispiel: "select from [datenquellename]".

Anmerkung: Wenn die Verbindungspoolingfunktion in DB2 Alphablox Cube Server aktiviert ist, können Sie den Knopf **Standardabfrage abrufen** nicht verwenden, um eine Standardabfrage für einen DB2 Alphablox-Würfel abzurufen.

- Wenn im PresentBlox unten auf der Seite Daten angezeigt werden, verwenden Sie die Standardbenutzerschnittstelle, um die Achsen auszutauschen, eine Gesamtabfrage oder eine Detailabfrage durchzuführen, Dimensionen von einer Achse auf eine andere Achse zu versetzen usw.
 - Klicken Sie den Knopf **Abfrage ausführen** an, um die Abfrageergebnisse anzuzeigen. Die Ergebnismenge wird im PresentBlox unten auf der Seite angezeigt.
6. Geben Sie qcc2003-acs in die eckigen Klammern der Standardabfrage ein. Beispiel: select from [qcc2003-acs]. Drücken Sie dann den Knopf **Abfrage ausführen**.

Der Abschnitt **PresentBlox** sollte eine Standardsicht auf der Grundlage der Abfrage anzeigen.

7. Erstellen Sie mit der PresentBlox-Benutzerschnittstelle eine neue Sicht. Sie können z. B. die Dimension **Products** auf die Zeilenachse in der Datenlayoutanzeige auf der linken Seite ziehen. Danach können Sie die Dimension **Time (Calendar)** auf die Spaltenachse ziehen. Anschließend könnten Sie die Dimension **Measures** auf die Seitenachse ziehen, woraufhin das Menü **Measures** in der Seitenanzeige, die Seitenfilter anzeigt, über dem Anzeigebereich geöffnet wird. Wählen Sie zwei weitere Dimensionen aus, und fügen Sie diese der Seitenachse hinzu, so dass über der Anzeige drei Seitenfilter angezeigt werden. Wenn Sie mit der Verwendung der DB2 Alphablox-Benutzerschnittstelle noch nicht vertraut sind, wählen Sie **Hilfe** in der PresentBlox-Funktionsleiste aus, um mehr über die Verwendung der Schnittstelle zu erfahren.
8. Nachdem Sie Ihre neue Sicht erstellt haben, beachten Sie die Abfrageanweisung, die im Abschnitt **Abfrage** angezeigt wird. Wenn die Abfrage nicht

dynamisch generiert wird (wenn die Option **Abfragen automatisch aktualisieren** nicht ausgewählt ist), können Sie den Knopf **Aktuelle Abfrage abrufen** anklicken, um die aktuelle Abfrage abzurufen. Das Textfenster zeigt die Abfrageanweisung an, die für das Entwickeln der aktuellen Datensicht erforderlich ist. Sie können die Abfrageanweisungen aus dem Feld **Abfrage** kopieren und einfügen, die bei der Entwicklung von DB2 Alphablox-Anwendungen verwendet werden sollen.

9. Klicken Sie den Knopf **Blox-Tag generieren** an. Ein Dialog wird geöffnet, der den Tag (und die verschachtelten Tags) für die Reproduktion des PresentBlox-Layouts und der PresentBlox-Ergebnismenge angezeigt.
10. Öffnen Sie in Ihrem Texteditor eine neue Datei, und fügen Sie ihr den folgenden Text hinzu. Speichern Sie die Datei dann als PresentBloxView.jsp. Im nächsten Thema werden viele wichtige Elemente des Codes auf dieser Seite erläutert.

```
<%@ taglib uri="bloxtld" prefix="blox" %>
//Kopieren Sie die Taginformationen aus dem generierten Blox-Tag, und
//fügen Sie sie direkt unter dieser Zeile ein, wobei Sie alle Zeilenumbrüche
//aus der Abfrageanweisung entfernen. Sie können diesen Kommentar löschen,
//nachdem Sie den Code aus DB2 Alphablox Query Builder hinzugefügt haben.
<html>
<head>
<blox:header/>
</head>
<body>
<h2>Simple PresentBlox View</h2>
<p>
<blox:display bloxRef="MyPresentBloxView"/>
</p>
</body>
</html>
```

11. Kopieren Sie die gespeicherte Datei (PresentBloxView.jsp) in das Verzeichnis **MeineAnw** auf Ihrem Server, und testen Sie Ihre neue analytische Sicht, indem Sie den Link zu dieser Seite (**Simple PresentBlox View**) von Ihrer Homepage **MeineAnw** aus anklicken. Alternativ hierzu können Sie die URL-Adresse direkt in ein Browseradressfeld eingeben (z. B. <http://localhost:9080/MeineAnw/PresentBloxView.jsp>). Die angezeigte Sicht sollte dem PresentBlox ähneln, das Sie in DB2 Alphablox Query Builder bearbeitet haben.

Im Folgenden wird ein Codebeispiel einer vollständigen Version der Seite **Simple PresentBlox View** (PresentBloxView.jsp) aufgeführt:

```
<%@ 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-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>
<html>
<head>
<blox:header/>
</head>
<body>
<h2>Simple PresentBlox View</h2>
<blox:display bloxRef="queryBuilder4_present"/>
</p>
</body>
</html>

```

Im oben aufgeführten Codebeispiel wurden der Abfrageanweisung Zeilenumbrüche zur besseren Lesbarkeit hinzugefügt. Damit diese Seite ordnungsgemäß ausgeführt werden kann, müssen Sie die Zeilenumbrüche in den Attributwerten (wie den oben aufgeführten Attributen `query` und `selectableSlicerDimensions`) entfernen. Im von DB2 Alphablox Query Builder generierten Code sind viele Tags enthalten, die nicht notwendig sind. Durch das Auflisten aller Tags erhalten Sie jedoch eine Vorstellung davon, wie Blox-Tags als übergeordnete und verschachtelte Tags funktionieren. Des Weiteren können Sie die Verwendung der Tagattribute verstehen, die als Ergebnis der von Ihnen im PresentBlox von Query Builder vorgenommenen Bearbeitung hinzugefügt wurden. Weitere Details zur Verwendung von Blox-Tags und Tagattributen sowie zur Entwicklung von DB2 Alphablox-Anwendungen finden Sie in der DB2 Alphablox-Informationszentrale.

Struktur der Seite Ihrer ersten analytischen Sicht

In diesem Thema wird die Struktur der ersten analytischen Sicht (PresentBloxView.jsp) zusammengefasst, die Sie im vorherigen Thema erstellt haben.

Weiter unten finden Sie eine kurze Übersicht einiger wichtiger Punkte der Seite, die Sie im vorherigen Thema erstellt haben. Damit Sie den Code noch besser verstehen, sollten Sie die entsprechenden relevanten Themen in der DB2 Alphablox-Informationszentrale lesen.

Die Seite **PresentBloxView.jsp** beginnt mit der folgenden Zeile:

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

Bei dieser Zeile handelt es sich um eine JSP-Anweisung `taglib`, die dem Server mitteilt, dass Sie die Blox-Tagbibliothek verwenden wollen. `uri` ist ein Verweis auf die Deskriptordatei der Tagbibliothek. Der Wert `prefix`, für den `blox` festgelegt ist, veranlasst den Server, nach allen Tags auf dieser Seite zu suchen, die mit `blox`

anfangen, und dann den Inhalt mit Hilfe der Blox-Tagbibliothek entsprechend der Definition in der Deskriptordatei der Tagbibliothek zu verarbeiten.

Direkt unter der Anweisung `taglib` wird die `PresentBlox`-Komponente durch die folgenden Tags und deren Tagattribute angegeben:

```
<blox:present
id="queryBuilder4_present"
height="500"
visible="false"
width="100%">
<blox:grid/>
<blox:chart/>
<blox:page/>
<blox:data
dataSourceName="qcc2003-ac"
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>
```

Der Tag `<blox:present>` gibt an, dass hier ein `PresentBlox` mit dem `id`-Attributwert `queryBuilder4_present` angezeigt werden soll. Das Attribut `id` ermöglicht es Ihnen, dieses besondere Blox für Scripterstellungszwecke anzugeben. Beachten Sie, dass der Tag `<blox:present>` ein Attribut `visible` enthält, das auf `false` gesetzt ist. Bei der Einstellung `false` wird das `PresentBlox` erst wiedergegeben, wenn der Compiler den Tag `<blox:display>` auf der Seite erkannt hat.

Für jedes mit Hilfe von Tags definierte Blox sind zahlreiche Attribute verfügbar. Die Angabe der Attribute im Tag ist jedoch nur dann notwendig, wenn sie sich von den Standardwerten unterscheiden. In diesem Beispiel wird für `height` der Wert 500 und für `width` der Wert 100% angegeben. Höhe und Breite können in Pixeln oder in Prozentsätzen angegeben werden.

Ferner ist ein verschachteltes `DataBlox` enthalten, das `qc2003-ac` für das Attribut `dataSourceName` festlegt. Wenn keine Datenquelle angegeben ist, wird beim Anzeigen der Seite eine Nachricht angezeigt, die besagt, dass keine Daten verfügbar sind. Das Abfrageattribut enthält die MDX-Abfrageanweisung, die von DB2 Alpha-

blox Query Builder generiert wurde. Wenn Sie mit der MDX-Abfragesprache vertraut sind, die mit DB2 Alphablox Cube Server verwendet wird, können Sie die Abfrage vereinfachen und dadurch verkürzen. Die hier dargestellte Abfrageanweisung ist jedoch vollständig und generiert die Sicht, die Sie in der Query Builder-Anwendung erstellt haben.

Der Abschnitt `<head>` der Seite enthält einen besonderen Blox-Tag, durch den der Seite wichtiger Code hinzugefügt wird, bevor die Seite wiedergegeben wird:

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

Dieser Tag wird von DB2 Alphablox verwendet, um dem Kopfsatzabschnitt der Seite automatisch erforderlichen HTML-, JavaScript- und CSS-Code hinzuzufügen. Wenn eine Seite vom Server wiedergegeben wird, fügt dieser Tag definierten HTML-Themen CSS-Links und einen Metatag zum Verhindern des Zwischenspeicherns hinzu. Denken Sie daran, diesen Tag auf jeder JSP-Seite einzugeben, die Blox-Komponenten verwendet. Wenn dieser Tag nicht vorhanden ist, wird die Seite nicht ordnungsgemäß wiedergegeben.

Im Abschnitt `<body>` der Seite befindet sich die folgende Zeile:

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

Dieser Blox-Tag bewirkt, dass das oben angegebene PresentBlox hier wiedergegeben wird. Dadurch können Sie Änderungen an einer Blox-Komponente bzw. an den zugehörigen verschachtelten Komponenten an der Stelle vornehmen, an der Sie möglicherweise noch weitere Blox-Tags und JSP-Scriptlets angegeben haben, die das Verhalten der Blox-Komponenten steuern. Wenn Sie sich mit DB2 Alphablox vertraut gemacht haben, können Sie in komplexen JSP-Dateien eine DataBlox-Komponente auch getrennt von den PresentBlox-Tags aufführen und JSP-Scriptlets hinzufügen, die die Blox-Komponenten und deren Verhalten modifizieren, bevor die Blox-Komponente im Web-Browser wiedergegeben wird.

Zusammenfassend kann man sagen, dass `<blox:header/>`, `<blox:present>` und `<blox:display>` zu den wichtigsten Blox-Tags gehören, die auf dieser Seite verwendet werden. Diese drei Tags und die zugehörigen verschachtelten Tags geben eine analytische Sicht an, ohne dass Java-Code auf der Seite erforderlich ist. Die Komplexität der Darstellungslogik wird von diesen drei Blox-Tags bewältigt. Durch Hinzufügen eines verschachtelten Blox und das Ändern der Attributwerte können Sie Sichten entsprechend Ihren Geschäftsanforderungen anpassen.

Für komplexere Sichten und Anwendungen können weitere JavaBeans-Komponenten und zusätzlicher Java-Code erforderlich sein.

Erstellen Ihrer zweiten analytischen Sicht

In dieser Task werden Sie eine analytische Sicht, die nur ein Diagramm mit Seitenfiltern anzeigt, basierend auf dem Code der ersten analytischen Sicht in diesem Lerntext erstellen.

In vielen Anwendungen werden Daten lediglich in einem Raster oder einem Diagramm dargestellt. Während die PresentBlox-Komponente GridBlox und ChartBlox als verschachtelte Komponenten kombiniert, die innerhalb der PresentBlox-Komponente gemeinsam angezeigt werden, können Sie die verschachtelten Komponenten, die nicht angezeigt werden sollen, inaktivieren oder ausblenden. Die Verwendung einer PresentBlox-Komponente zum Erstellen einer Diagrammsicht mit Seitenfiltern ist relativ einfach und wird erreicht, indem Sie die Tagattribute in einigen verschachtelten Blox-Tags modifizieren.

In dieser Task werden Sie den PresentBlox-Tag der ersten analytischen Sicht wiederverwenden, um eine zweite angepasste analytische Sicht zu erstellen.

Gehen Sie wie folgt vor, um eine analytische Sicht zu erstellen, die ein Diagramm mit Seitenfiltern anzeigt:

1. Öffnen Sie die zuvor erstellte Datei **PresentBloxView.jsp** in Ihrem Editor, und speichern Sie sie als **CustomizedChartBlox.jsp**.
2. Fügen Sie in den unten aufgelisteten verschachtelten Komponenten des PresentBlox-Tags jedem Tag ein Attribut `visible` hinzu, und setzen Sie den Wert auf `false`: Das Hinzufügen dieses Attributs bewirkt, dass diese verschachtelten Komponenten nicht angezeigt werden, wodurch eine Diagrammsicht mit einem Seitenfilterbalken erzeugt wird.

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

3. Modifizieren Sie den PageBlox-Tag (`<blox:page>`), indem Sie das Attribut `labelPlacement` hinzufügen und den Wert auf `top` setzen. Die Bezeichnung für Seitenfilter wird standardmäßig links von den Menüs angezeigt. Das Hinzufügen dieses Attributs und das Setzen seines Werts auf `top` überschreibt dieses Standardverhalten und platziert die Bezeichnungen oberhalb der Seitenfilter.

```
<blox:page labelPlacement="top" />
```
4. Ändern Sie den Namen des ID-Attributs im Tag `<blox:present>` in den folgenden Wert: `chartview`. Jede Blox-Komponente, die auf einer Seite angezeigt oder wiedergegeben wird, benötigt eine eindeutige ID, um Fehler zu vermeiden.
5. Speichern Sie Ihre Änderungen, und fügen Sie die Datei Ihrem Verzeichnis **MeineAnw** auf dem Server hinzu:
6. Öffnen Sie Ihren Web-Browser, und testen Sie Ihre neue Seite. Klicken Sie hierzu entweder den Link auf Ihrer Homepage an, oder greifen Sie direkt unter Verwendung der URL-Adresse auf sie zu (z. B. <http://localhost:9080/MeineAnw/CustomizedChartView.jsp>).

Sie haben nun zwei analytische Sichten mit der Anwendung DB2 Alphablox Query Builder erstellt, die Sie bei den ersten Schritten unterstützt hat. Sie haben ferner gelernt, dass Sie durch Modifizieren von Attributen bzw. durch Hinzufügen von Attributen zu vorhandenen Tags die Darstellung und das Verhalten des Blox und seiner verschachtelten Komponenten steuern können.

Zusammenfassung

Wenn Sie alle Lerntexttasks ausgeführt haben, haben Sie gelernt, wie eine DB2 Alphablox-Basisanwendung mit Hilfe der DB2 Alphablox-Verwaltungsseiten, der Anwendung DB2 Alphablox Query Builder und der Blox-Tags erzeugt wird. Wenn Sie das Handbuch *Developer's Reference* bereits durchgelesen haben, ist Ihnen bewusst, dass es viele Merkmale und Methoden gibt, die zum Definieren und Bearbeiten von Bloxes auf JSP-Seiten verfügbar sind. Wie geht es weiter?

Wenn Sie Ihre Unternehmensdaten in einer DB2 Alphablox-Sicht anzeigen wollen, wissen Sie jetzt, wie eine Anwendung von Anfang an erstellt wird und wie Ihrer Anwendung analytische Sichten hinzugefügt werden. Wenn Sie sofort etwas anzeigen wollen, können Sie an der Anwendung **MeineAnw** einige einfache Änderungen vornehmen. Damit Sie Daten aus Ihren unternehmensweiten Datenbanken an Stelle der Datenquelle **qcc2003-acs** anzeigen können, können Sie Datenquellen erstellen, die auf Ihre unternehmensweiten Datenbanken verweisen.

Außerdem müssen Sie die `dataSourceName`-Attribute so modifizieren, dass sie auf die neu definierte Datenquelle verweisen. Anschließend müssen Sie ein entsprechendes Attribut `query` hinzufügen. Sie können unter Verwendung von DB2 Alphablox Query Builder verwendbare Abfrageanweisungen und Blox-Tags oft schnell erstellen. Details zum Erstellen von Datenquellen finden Sie im Handbuch *Administrator's Guide*. Informationen zum Erstellen von geeigneten Abfragen und weitere Details zum Erzeugen von DB2 Alphablox-Anwendungen finden Sie im Abschnitt über das Abrufen von Daten im Handbuch *Developer's Guide* und im DataBlox-Abschnitt des Handbuchs *Developer's Reference*.

Kapitel 2. Lerntext: Erzeugen Ihres ersten Portlets mit Blox-Komponenten

Mit Hilfe dieses Lerntexts erfahren Sie, wie Sie Ihren Portlets Blox-Komponenten hinzufügen. Sie werden folgende Aktionen ausführen:

1. Installieren eines vordefinierten Musterportlets. Durch das Ausführen dieses Schritts lernen Sie schnell, wie einer Portlet-JSP-Seite eine Blox-Komponente hinzugefügt wird, und wie sie auf einer Portalseite angezeigt wird.
2. Schreiben Ihrer eigenen JSP-Seite mit einem GridBlox.

Die Tasks in diesem Lerntext enthalten keine Details zur allgemeinen Portletentwicklung. In diesem Lerntext liegt der Schwerpunkt auf den DB2 Alphablox-spezifischen Tasks. Außerdem wird eine gewisse Vertrautheit mit den allgemeinen Konzepten der Portalumgebung und der Portletentwicklung vorausgesetzt.

Voraussetzungen

- DB2 Alphablox muss auf einem Server mit WebSphere Portal Version 5.1 installiert sein. Details zur Installation finden Sie im *Installationshandbuch*.
- Ihr WebSphere Portal-Server muss gestartet sein.
- Sie müssen über den Verwaltungszugriff auf Ihren WebSphere Portal-Server verfügen.
- Sie müssen mit den Verwaltungsfunktionen und der Benutzerschnittstelle in WebSphere Portal vertraut sein.
- Sie sollten über Grundkenntnisse in Java und JSP verfügen.
- Sie sollten einen JSP-Editor installiert haben.

Sie können einen beliebigen JSP-Editor oder sogar einen Texteditor zum Ausführen der Schritte in diesem Lerntext verwenden. Wenn Sie jedoch Ihre eigenen Portlets entwickeln, sollten Sie ein Entwicklungstool verwenden, das von WebSphere Portal empfohlen wird, wie z. B. Rational Application Developer.

In diesem Lerntext wird eine vordefinierte Datenquelle verwendet, die mit DB2 Alphablox installiert wurde. Sie können diese Datenquelle verwenden, um schnell eine Basisanwendung zu entwickeln. Zum Ausführen der Schritte in diesem Lerntext müssen Sie keine angepasste Datenquelle konfigurieren.

Installieren der Musterportlets

Die beste Möglichkeit, das Hinzufügen eines Blox zu Ihrem Portlet zu erlernen, besteht darin, die Musterportlets zu installieren, die in DB2 Alphablox bereitgestellt werden, und sie auf eine Portalseite zu laden. Dadurch können Sie die Basisstruktur des JSP-Codes untersuchen und diesen mit der Ausgabe in Ihrem Portal in Übereinstimmung bringen.

Installieren Sie die Musterportlets, die in DB2 Alphablox zur Verfügung gestellt werden:

1. Öffnen Sie Ihren Browser, und melden Sie sich an Ihrem Portal als Benutzer mit Verwaltungsaufgaben an. (Die URL-Adresse hat das Format `http://<ihr_portalserver>:<port>/wps/portal.`)

2. Klicken Sie den Knopf **Administration** an.
3. Klicken Sie **Web Modules** im Bereich **Portlet Management** an. Die Seite **Manage Web Modules** wird rechts angezeigt.
4. Klicken Sie **Install** an. Sie werden aufgefordert, das Webmodul zu installieren.
5. Klicken Sie den Knopf **Browse** an, und navigieren Sie zum Verzeichnis `installableApps`, das sich in Ihrem DB2 Alphablox-Installationsverzeichnis befindet.
6. Wählen Sie `AlphabloxSamplePortlets.war` aus, und klicken Sie **Next** an. Die DB2 Alphablox-Musterportletanwendung mit einem Portal namens **DB2 Alphablox JSP Page Sample Portlet** wird in der Portletanwendungstabelle angezeigt.
7. Klicken Sie **Finish** an.

Die DB2 Alphablox-Musterportletanwendung und die enthaltenen Portlets sind jetzt installiert. Überprüfen Sie das Verzeichnis `installedApps` unter Ihrer WebSphere Portal-Installation. Der Name für das neu erstellte Verzeichnis beginnt mit `DB2 Alpha` und endet mit einer dynamisch generierten Portlet-ID nach dem folgenden Muster: `_PA_x_x_xx.ear`.

Ausführen der Musterportlets

Gehen Sie wie folgt vor, um die Musterportlets auf Ihrer Portalseite auszuführen:

1. Wechseln Sie auf Ihre Portalseite.
2. Erstellen Sie eine Seite, oder bearbeiten Sie eine vorhandene Seite. Sie können eine neue Seite für das Testen dieses Musterportlets erstellen oder eine vorhandene Seite anklicken, um die Seite zu bearbeiten.
3. Klicken Sie einen der **Add Portlets**-Knöpfe auf der Portallayoutseite an.
4. Geben Sie `DB2` in das Suchfeld ein, und drücken Sie **Search**. Das Markierungsfeld **DB2 Alphablox JSP Page Sample Portlet** wird angezeigt.
5. Wählen Sie das Markierungsfeld aus, und klicken Sie **OK** an.
6. Klicken Sie **Done** an.

Nach dem Aktualisieren der Portalseite wird ein `PresentBlox` auf Ihrer Portalseite angezeigt. Dieses `PresentBlox` verfügt über Folgendes:

- Eine Menüleiste oben
- Zwei Funktionsleistenanzeigen unterhalb der Menüleiste
- Eine Datenlayoutanzeige links, mit der Sie Dimensionen auf verschiedenen Achsen versetzen können
- Ein Raster, das die Daten im Tabellenformat anzeigt
- Ein 3D-Balkendiagramm rechts

Die Datei `present.jsp` ist die Standardseite, die wie im Servlet `BloxJSPPagePortlet` angegeben geladen wird, das in diesem Muster enthalten ist. Der Quellcode für dieses Musterservlet ist im Verzeichnis `WEB-INF/src/` verfügbar.

Sie haben die Musterportlets erfolgreich installiert und Ihrer Portalseite ein Portlet mit `Blox` hinzugefügt. Die nächste Task besteht darin, die Codestruktur auf dieser JSP-Seite zu untersuchen.

Untersuchen der Struktur einer Portlet-JSP-Seite mit einer Blox-Komponente

Diese Task beinhaltet das Überprüfen der Codestruktur einer JSP-Seite, die ein Blox enthält. Alle JSP-Seiten müssen über dieselben Schlüsselemente verfügen.

Gehen Sie wie folgt vor, um eine JSP-Datei zu öffnen:

1. Navigieren Sie zum Verzeichnis `installedApps/` unter Ihrer WebSphere Portal-Installation, und suchen Sie den neu erstellten Anwendungsordner, der mit DB2 Alpha beginnt.
2. Navigieren Sie zum Verzeichnis `PA_x_x_xx.war/jsp/html/`.
3. Öffnen Sie `present.jsp` in Ihrem JSP- oder Java-Editor.
4. Untersuchen Sie den folgenden Code, und beachten Sie die Schlüsselemente:

```
<%@ 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>
```

Dieser Codeblock enthält sechs Schlüsselemente:

1. Die erste Zeile teilt dem Browser mit, dass die Ausgabe HTML ist:

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

2. Die nächste Codegruppe gibt die zwei JSP-Tagbibliotheken an, die auf dieser Seite verwendet werden:

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

`uri` ist ein Verweis auf die Verzeichnisposition, an der sich die Deskriptordatei der Tagbibliothek befindet. Die `prefix`-Werte, die als `blox` und `portletAPI` definiert sind, teilen dem Server mit, folgende Aktionen auszuführen:

- Suchen nach allen Tags auf dieser Seite, die mit `blox` anfangen, und Verarbeiten des Inhalts mit Hilfe der Blox-Tagbibliothek entsprechend der Definition in der Deskriptordatei der Tagbibliothek.
 - Suchen nach allen Tags auf dieser Seite, die mit `portletAPI` anfangen, und Verarbeiten des Inhalts mit Hilfe der Portlet-Tagbibliothek entsprechend der Definition in der Deskriptordatei der Tagbibliothek.
3. Als Nächstes wird ein Initialisierungstag hinzugefügt:

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

Dieser Tag bietet Zugriff auf die Objekte `PortletRequest`, `PortletResponse` und `PortletConfig`. Mit `PortletResponse` können Sie die Methode `encodeNamespace()` aufrufen, um sicherzustellen, dass der Name Ihres Blox nicht mit anderen Objekten in anderen Portlets in Konflikt steht, die auf derselben Seite ausgeführt werden.

4. Der nächste Tag codiert den Namensbereich für das Blox, das der Seite hinzugefügt werden soll:

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

Dadurch können Sie später ein Blox erstellen und ihm diesen eindeutigen Name zuordnen.

- Der nächste Codeblock fügt den Blox-Header-Tag hinzu, der für die Blox-Wiedergabe und die Client/Server-Kommunikation erforderlich ist.

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

Dieser Tag wird von DB2 Alphablox verwendet, um dem Kopfsatzabschnitt der Seite automatisch erforderlichen HTML-, JavaScript- und CSS-Code hinzuzufügen. Wenn eine Seite vom Server wiedergegeben wird, führt dieser Tag zur Aufnahme von CSS-Links in definierten HTML-Themen und eines Metatags zum Verhindern des Zwischenspeicherns. Dieser Tag muss jeder JSP-Seite hinzugefügt werden, die Blox-Komponenten enthält, sonst werden die Komponenten nicht ordnungsgemäß wiedergegeben.

- Fügen Sie ein PresentBlox unter Verwendung von Tags hinzu, die in der Blox-Tagbibliothek zur Verfügung gestellt werden:

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

- Dieser Code fügt ein PresentBlox hinzu, bei dem id mit presentBlox und bloxName mit xx_x_x_xxx_presentBlox angegeben ist. Dies ist das Ergebnis der Namensbereichscodierung.
- Dieses Blox hat eine Breite von 800 Pixeln und eine Höhe von 400 Pixeln (Standardwert).
- Außerdem ist ein verschachteltes DataBlox enthalten, dessen Tagattribut dataSourceName auf canned gesetzt ist.

Das Tagattribut id des PresentBlox ist erforderlich. Es gibt den Java-Scripterstellungsnamen an, der auf der JSP-Seite verwendet werden soll. Das Attribut bloxName gibt den Objektnamen auf dem Server an. Durch das codierte Attribut bloxName wird sichergestellt, dass die Instanz auf dem Server eindeutig ist.

Anmerkung: Wenn keine Datenquelle angegeben wurde, wird standardmäßig die Nachricht **Keine Daten verfügbar** im Raster angezeigt. Die feststehende Datenquelle wird während der Installation vordefiniert und enthält eine kleine Menge an Musterdaten. Sie erfordert keine Installation und Konfiguration einer realen externen Datenbank und kann zur Fehlerbehebung und zu Lernzwecken verwendet werden. Da keine Abfrage angegeben werden muss, ist kein Attribut query vorhanden.

Diese Seite verfügt über keinen äußersten <html>-Tag, da diese JSP-Seite auf einer Portalseite mit anderen Portlets angezeigt wird. Es ist kein zusätzlicher <html>- oder <body>-Tag erforderlich.

Jetzt haben Sie die JSP-Struktur untersucht und den notwendigen Code kennen gelernt, der auf Ihren Portlet-JSP-Seiten verwendet werden soll. In der nächsten Task werden Sie eine neue JSP-Seite mit einem anderen Blox erstellen und einige allgemeine Blox-Attribute angeben, damit sich die Seite besser in Ihre Portalseite einfügt.

Erstellen Ihrer eigenen Portlet-JSP-Seite mit Blox-Komponenten

Diese Task beinhaltet das Erstellen einer neuen JSP-Seite mit einem GridBlox und das Festlegen einiger zugehöriger Merkmale. Ziel ist es, Sie mit dem allgemeinen Aufbau eines Blox-Tags vertraut zu machen, während Sie ein GridBlox erstellen, das sich gut in eine typische Portalseite einfügt.

Ein GridBlox verfügt über die Standardgröße von 400x400 Pixel. Es verfügt außerdem über eine Menüleiste und eine Funktionsleiste. Sie werden einige der allgemein verwendeten GridBlox-Merkmale festlegen, so dass das Raster nur 100 Pixel hoch ist, wenn die Menüleiste und die Funktionsleiste inaktiviert sind. Dies wird durch das Festlegen der folgenden GridBlox-Merkmale erreicht:

- `height` : auf 100 Pixel setzen
- `menubarVisible`: auf false setzen
- `toolbarVisible`: auf false setzen

Stellen Sie sicher, dass Sie die in Kapitel 2, „Lerntext: Erzeugen Ihres ersten Portlets mit Blox-Komponenten“, auf Seite 13 angegebenen Anforderungen erfüllt und die im Abschnitt „Installieren der Musterportlets“ auf Seite 13 beschriebene Musterportletanwendung installiert haben.

Führen Sie die folgenden Schritte aus:

1. Klicken Sie im Browserfenster, in dem das PresentBlox angezeigt wird, den Bearbeitungsknopf (der Knopf mit dem Stiftsymbol) des Portlets an. Daraufhin wird eine Auswahl-Dropdown-Liste angezeigt.
2. Wählen Sie in der Dropdown-Liste "Grid Blox" aus, und klicken Sie **OK** an. Nach dem Aktualisieren der Seite sollte ein GridBlox mit der Größe 400x400 angezeigt werden. Bei diesem GridBlox sind die Menüleiste und die Funktionsleiste standardmäßig aktiviert. Ändern Sie die Größe des GridBlox in 400x100, wobei sowohl die Menüleiste als auch die Funktionsleiste inaktiviert sind.
3. Navigieren Sie zum Verzeichnis `PA_x_x_xx.war/jsp/html/` unter der zuvor installierten DB2 Alphablox-Musterportletanwendung.
4. Öffnen Sie `grid.jsp` in Ihrem JSP-Editor. Diese Seite ist bis auf die folgenden Ausnahmen identisch mit `present.jsp`:
 - Der Tag `<blox:present>` lautet jetzt `<blox:grid>`, und der Wert für `id` ist anders:

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400">
<blox:data dataSourceName="canned" />
</blox:grid>
```
 - Der Wert für `bloxName` ist anders:

```
<%
String bloxName = portletResponse.encodeNamespace("gridBlox");
%>
```
5. Legen Sie für die Höhe dieses GridBlox 100 Pixel fest, indem Sie das Attribut `height` hinzufügen und seinen Wert auf 100 setzen:

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400" height="100">
<blox:data dataSourceName="canned" />
</blox:grid>
```
6. Inaktivieren Sie die Menüleiste und die Funktionsleiste oben, indem Sie die Attribute `menubarVisible` und `toolbarVisible` auf false setzen:

```

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

```

Stellen Sie sicher, dass Sie die Attributnamen korrekt eingeben, einschließlich der Großschreibung (beide Attributnamen werden mit einem großen "V" geschrieben). Stellen Sie außerdem sicher, dass die Attribute vor der spitzen Schlussklammer (">") hinzugefügt werden.

7. Ändern Sie den Namensbereich in myFirstGrid:

```

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

```

Durch das Ändern des Namensbereichs für dieses GridBlox kann sichergestellt werden, dass die gerade vorgenommenen Änderungen wiedergegeben werden, wenn Sie diese JSP in Ihr Portal laden. Da Sie diese Seite vor kurzem geladen haben, ist für diese Sitzung bereits eine Instanz dieses GridBlox auf dem Server aktiv. Wenn Sie den Namensbereich nicht ändern, werden die Änderungen, die Sie gerade vornehmen, nicht wiedergegeben, auch wenn Sie die Seite aktualisieren. Durch das Ändern des Namensbereichs können Sie Ihre Änderungen schnell in einer Entwicklungsumgebung testen. Alternativ können Sie auch ein neues Browserfenster öffnen, so dass für die neue Sitzung ein neues Objekt auf dem Server erstellt wird.

8. Speichern Sie die Datei.

Sie können diese Datei nun in WebSphere Portal testen.

Gehen Sie wie folgt vor, um die gerade vorgenommenen Änderungen zu testen:

1. Gehen Sie zurück auf die Portalseite.
2. Klicken Sie den Knopf des Browsers zum Aktualisieren an, um die Seite erneut zu laden.

Es sollte ein GridBlox der Größe 400x100 ohne Menü- und Funktionsleiste angezeigt werden.

Anmerkung: Diese JSP-Auswahl-Dropdown-Liste ist keine integrierte Funktion von WebSphere Portal. Sie wurde von diesem Musterportlet erstellt. Überprüfen Sie die Datei `edit.jsp` und die Java-Quellendateien im Verzeichnis `WEB-INF/src/`.

Erstellen eines Portletprojekts zum Verwenden von Blox-Komponenten

Wenn Sie Ihr Portletprojekt in Ihrem Entwicklungstool erstellen, stellen Sie sicher, dass die `AlphabloxServer`-Servletzuordnung und die Tagbibliotheksverweise, die für Blox-Komponenten benötigt werden, der Datei `web.xml` Ihres Projekts hinzugefügt werden und dass die TLD-Dateien für DB2 Alphablox-Tagbibliotheken in Ihr Projekt kopiert werden.

1. Modifizieren Sie die Datei `web.xml` Ihres Projekts, so dass die folgenden Zeilen enthalten sind:

- Für Servletdefinition und Servletzuordnung:

```

<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>

```

- Für Tagbibliotheksverweise:

```

<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. Kopieren Sie die TLD-Dateien für DB2 Alphablox-Tagbibliotheken in das Verzeichnis WEB-INF/tlds/ Ihres Projekts. Diese Dateien befinden sich im folgenden Verzeichnis:

```
<db2alphablox_verz>/bin/
```

Dabei ist <db2alphablox_verz> Ihr DB2 Alphablox-Installationsverzeichnis.

Konfigurieren eines Portletprojekts mit Rational Application Developer

WebSphere Portal empfiehlt die Verwendung von Rational Application Developer für die Portletentwicklung. Rational Application Developer Version 6.0 ist insbesondere für WebSphere Portal Version 5.1 erforderlich. Rational Application Developer stellt einen Assistenten zur Verfügung, der Sie durch das Setup und die Erstellung eines Portletprojekts führt. Ausgehend von Ihrer Auswahl konfiguriert er die entsprechende Struktur und die Implementierungsdeskriptordatei. Ferner erstellt er automatisch die notwendigen Java-Klassen für Ihren Controller und die JSP-Seiten für Ihre Portletsicht. Während Sie dem Assistenten **New Portlet Project** folgen, stellen Sie sicher, dass Sie Folgendes korrekt konfiguriert haben:

- Wählen Sie **WebSphere Portal v5.1 stub** als Ihren Zielservers aus. Dies wird in der ersten Anzeige des Assistenten angegeben, indem Sie den Knopf **Show Advanced >>** anklicken.
- Fügen Sie der Datei web.xml die Servletzuordnung für DB2 Alphablox hinzu, die unter WebContent/WEB-INF/ erstellt wurde. Dies erreichen Sie, indem Sie die folgende Servletdefinition und den folgenden Zuordnungscode in die Datei web.xml Ihres Projekts eingeben:

```

<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>

```

- Fügen Sie der Datei web.xml Ihres Projekts die Alphablox-Tagbibliotheken hinzu. Führen Sie dazu die folgenden Schritte aus:
 1. Klicken Sie bei geöffneter Datei web.xml die Indexzunge **Variables** an.
 2. Blättern Sie ans Ende zum Abschnitt für die Tagbibliotheksverweise.
 3. Klicken Sie **Add** an.
 4. Geben Sie die Werte für **URL** (URL-Adresse) und **Location** (Speicherposition) auf Grundlage der folgenden Tabelle ein:

URL-Adresse	Speicherposition
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

Für jedes Paar aus URL-Adresse und Speicherposition müssen Sie **Finish** anklicken, um den Tagbibliotheksverweis hinzuzufügen. Klicken Sie anschließend **Add** an, um das nächste Paar hinzuzufügen.

Anmerkung: Details zu den unterschiedlichen Tagbibliotheken und deren Verwendung finden Sie im Abschnitt über die Verwendung von Java-Server Pages und Blox.

Denken Sie daran, die TLD-Dateien für die DB2 Alphablox-Tagbibliotheken in das Verzeichnis WEB-INF/tlds/ Ihres Projekts zu kopieren. Diese Dateien befinden sich im folgenden Verzeichnis:

```
<db2alphablox_verz>/bin/
```

Dabei ist <db2alphablox_verz> Ihr DB2 Alphablox-Installationsverzeichnis.

Weitere Schritte

Sie haben gelernt, wie ein Portlet mit Blox-Komponenten und der notwendigen Codestruktur erstellt wird. Gehen Sie wie folgt vor, um Daten aus Ihren Datenbanken anzuzeigen:

1. Definieren Sie eine neue Datenquelle für DB2 Alphablox, die auf Ihre Datenquelle verweist.
2. Modifizieren Sie das Attribut dataSourceName des DataBlox so, dass es auf die definierte Datenquelle verweist.
3. Fügen Sie DataBlox ein Attribut query mit Ihrer Abfragezeichenfolge hinzu.

Details zum Erstellen von Datenquellen finden Sie im Handbuch *Administrator's Guide*. Informationen zum Erstellen von geeigneten Abfragen finden Sie im Abschnitt über das Abrufen von Daten im Handbuch *Developer's Guide* und im DataBlox-Abschnitt des Handbuchs *Developer's Reference*.

Sobald Ihre eigenen Daten in Blox-Sichten angezeigt werden, können Sie auch damit anfangen, die vielen Merkmale zu untersuchen, die mit Hilfe der Blox-Tags gesetzt werden können.

Tipps zur Portletentwicklung

Wenn Sie die in der DB2 Alphablox-Dokumentation bereitgestellten Informationen verwenden und mit Ihrer Portletentwicklung fortfahren, sollten Sie sich über einige portletspezifische Themen, Konzepte und allgemeine Richtlinien bewusst sein. Im Folgenden werden diese allgemeinen Entwicklungsrichtlinien und Verweise auf bestimmte Abschnitte in der Dokumentation aufgelistet, die für Sie interessant sind.

- Codieren Sie den Namen Ihres Blox grundsätzlich unter Verwendung des Namensbereichs des Portlets. Durch die Verwendung eines Namensbereichs wird sichergestellt, dass der Blox-Name für die aktuelle J2EE-Sitzung eindeutig ist.
- Legen Sie die Breite und Höhe eines Blox immer mit Pixeln fest. Außerhalb der Portalumgebung können Sie die Breite und Höhe auf einen Prozentwert setzen, wie z. B. "50%" oder "100%". In der Portalumgebung funktionieren Prozentwerte nicht, da auf der Seite mehrere Portlets vorhanden sind.
- Verwenden Sie keine relativen URL-Adressen, um Ressourcen innerhalb eines Portlets aufzurufen. URL-Adressen sollten mit der Methode `encodeURL()` der `PortletResponse`-Klasse codiert werden. An den Stellen, an denen in der Dokumentation die Verwendung relativer URL-Adressen beschrieben wird, sollten Sie die URL-Adressen immer codieren.
- Verwenden Sie das Dienstprogramm für Portal-Schemas (Portal Theme Utility), um sicherzustellen, dass Ihr Blox-Portlet ein Thema verwendet, das Ihrem Portaltema ähnelt. Dieses Dienstprogramm ist über die folgenden zwei Quellen verfügbar:
 - Die DB2 Alphablox-Homepage
Klicken Sie unter der Indexzeile **Administration** den Link **General** an. Das Dienstprogramm ist im Bereich **Portal** aufgelistet. Details finden Sie im Handbuch *Administrator's Guide* und in der Onlinehilfe.
 - Die Datei `AlphabloxAdminPortlets.war`
Dies ist eine Portletversion des Dienstprogramms für Portal-Schemas. Sie können dieses Portlet für die Ausführung in Ihrem Portal installieren, ohne sich separat an der DB2 Alphablox-Homepage anmelden zu müssen.

Dieses Dienstprogramm kombiniert die Darstellung Ihrer Portalumgebung und die Darstellung in DB2 Alphablox, so dass ein Blox in einer ähnlichen Farbe und mit ähnlicher Schriftart angezeigt wird wie andere Portlets auf der Seite.

- Lesen Sie den Abschnitt über die Blox-Portlettagbibliothek im Handbuch *Developer's Guide*, um mehr über die Themen zu erfahren, die Sie während der Planungsphase berücksichtigen müssen.
- DB2 Alphablox verfügt über ein leistungsfähiges Blox-UI-Modell. Das UI-Modell enthält ein `ClientLink`-Objekt, mit dem Sie eine angegebene URL-Adresse laden können, wenn eine Blox-Komponente auf der Seite angeklickt wird. Verwenden Sie bei der Portletentwicklung die Blox-Portlettagbibliothek, um das `ClientLink`-Objekt zu erstellen. Die Tags führen die URL-Erstellung und -Handhabung dynamisch aus, so dass die Links nach dem erneuten Anzeigen der Seite immer aktuell sind. Details hierzu finden Sie im Abschnitt über das Planen der Portletentwicklung im Handbuch *Developer's Guide*.

- Verwenden Sie für Portlet- und Aktions-URIs die Blox-Portlettagbibliothek, um den Portlet-Link bzw. den Aktionslink zu erstellen. Anschließend können Sie die Portlet-API des Portals verwenden, um den Aktionslink bzw. Portlet-Link zu verarbeiten. Details hierzu finden Sie im Thema über die Blox-Portlettagbibliothek im Handbuch *Developer's Guide*.

Kapitel 3. Lerntext: Erzeugen von Anwendungen mit den Rational Developer-Tools

Sie können Rational Developer-Tools mit dem DB2 Alphablox-Toolkit verwenden, um Anwendungen mit DB2 Alphablox-Komponenten und den DB2 Alphablox-Java-APIs schneller zu entwickeln.

Die Rational Developer-Tools (Rational Application Developer oder Rational Web Developer) können mit dem DB2 Alphablox-Toolkit, einer Eclipse-basierten Gruppe von Plug-ins, verwendet werden, um Anwendungen zu entwickeln und zu testen, die mit der DB2 Alphablox-Technologie erzeugt wurden. In diesem Lerntext erhalten Sie eine Anleitung zur Konfiguration Ihrer integrierten Rational-Entwicklungsumgebung, um die Verwendung der Methoden- und Tagervollständigung sowie angepasste funktionale Erweiterungen zu aktivieren, die mit dem DB2 Alphablox-Toolkit hinzugefügt wurden.

Gehen Sie zum Konfigurieren Ihres Rational Developer-Tools mit dem DB2 Alphablox-Toolkit wie folgt vor, um die DB2 Alphablox-Anwendungen zu entwickeln und zu testen.

Vorbereiten Ihrer Entwicklungsumgebung

Bevor Sie das DB2 Alphablox-Toolkit installieren und verwenden können, bereiten Sie Ihre Entwicklungsumgebung vor, indem Sie sicherstellen, dass Sie über die gesamte erforderliche Software verfügen.

Gehen Sie wie folgt vor, um Ihre Entwicklungsumgebung vorzubereiten:

1. Installieren Sie ein Rational Developer-Tool (entweder Rational Application Developer oder Rational Web Developer) auf Ihrer Workstation.
2. Installieren Sie alle erforderlichen Aktualisierungen, um sicherzustellen, dass Sie mindestens über Rational Application Developer oder Rational Web Developer Version 6.0.0.1 verfügen.
3. Wenn Sie vorhaben, integrierte Testumgebungen von WebSphere 5.1 zu verwenden, stellen Sie sicher, dass Sie Ihre Rational Developer-Version aktualisiert haben, damit diese optionale Installation enthalten ist.

Anmerkung: Die Standardinstallationen der Rational Developer-Tools enthalten die integrierten Testumgebungen von WebSphere 5.1 nicht.

Jetzt können Sie die Tasks in diesem Lerntext ausführen.

Installieren des DB2 Alphablox-Toolkits

Die Verwendung des DB2 Alphablox-Toolkits mit dem Rational Developer-Tool erleichtert es Ihnen, mit der Entwicklung webbasierter Anwendungen mit DB2 Alphablox-Inhalt zu beginnen.

Das DB2 Alphablox-Toolkit erfordert die Verwendung von Rational Application Developer oder Rational Web Developer Version 6.0.0.1.

Bei der Installation des DB2 Alphablox-Toolkits in ein Rational Developer-Tool ist es nicht erforderlich, viele manuelle Schritte auszuführen, um die Verwendung der

Inhaltsvervollständigung für Java-Methoden und angepasste JSP-Tags zu aktivieren. Sie können auch angepasste Assistenten und Cheat-Sheets verwenden, um sich beim Erstellen von Anwendungen mit DB2 Alphablox-Inhalt und dem Definieren der WebSphere-Serverinstanzen anleiten zu lassen.

Gehen Sie wie folgt vor, um das DB2 Alphablox-Toolkit in Rational Application Developer oder Rational Web Developer zu installieren:

1. Legen Sie die DB2 Alphablox-Installations-CD in das CD-Laufwerk Ihrer Workstation ein.
2. Suchen Sie im Verzeichnis plugin das Unterverzeichnis namens UpdateSite.
3. Kopieren Sie das Verzeichnis UpdateSite an eine günstige Speicherposition auf Ihrer Festplatte. Kopieren Sie es z. B. in C:\DB2Alphablox\UpdateSite.
4. Starten Sie das Rational Developer-Tool.
5. Wählen Sie in der Menüleiste **Help > Software Updates > Find and Install** aus.
6. Wählen Sie im daraufhin angezeigten Fenster **Install** die Option **Search for new features to install** aus, und klicken Sie dann **Next** an.
7. Klicken Sie im Fenster **Update sites to visit** den Knopf **New Local Site** an, und blättern Sie zur Speicherposition des Verzeichnisses UpdateSite.
8. Klicken Sie **Next** an, wählen Sie die Funktion **DB2 Alphablox Toolkit** aus, und klicken Sie erneut **Next** an.
9. Akzeptieren Sie in den nachfolgend angezeigten Fenstern die Lizenzvereinbarung, und wählen Sie die Speicherposition aus, an der die DB2 Alphablox-Toolkit-Funktion installiert wird.
10. Nachdem die Installation beendet ist, starten Sie das Rational Developer-Tool erneut.

Nachdem Sie das Rational Developer-Tool erneut gestartet haben, sind die Funktionen des DB2 Alphablox-Toolkits verfügbar.

Installieren von DB2 Alphablox in integrierten WebSphere-Testumgebungen

Diese Task beinhaltet das Installieren von DB2 Alphablox in den integrierten WebSphere-Testumgebungen eines Rational Developer-Tools.

Voraussetzungen: Die unterstützte integrierte WebSphere-Testumgebung muss im Rational Developer-Tool installiert sein. Sie müssen Zugriff auf die DB2 Alphablox-Installations-CD haben. Prüfen Sie alle erforderlichen Vorinstallationsschritte, die im Abschnitt über die Vorinstallationstasks des *DB2 Alphablox Installationshandbuchs* beschrieben werden, und führen Sie diese aus.

DB2 Alphablox kann auf eigenständigen WebSphere-Anwendungsservern oder in den integrierten WebSphere-Testumgebungen, die innerhalb des Rational Developer-Tools verfügbar sind, installiert werden. Die Zusammenfassung der unten aufgeführten Schritte erläutert die Unterschiede bei der Installation, die beachtet werden müssen, wenn DB2 Alphablox in den WebSphere-Laufzeitumgebungen installiert wird, die innerhalb des Rational Developer-Tools verfügbar sind.

Gehen Sie wie folgt vor, um DB2 Alphablox in integrierten WebSphere-Testumgebungen zu installieren.

Installieren Sie DB2 Alphablox, indem Sie die Schritte aus dem Installationsabschnitt des *DB2 Alphablox Installationshandbuchs* mit den folgenden Ausnahmen ausführen:

1. Geben Sie im Fenster **WebSphere konfigurieren** des Installationsprogramms in das Feld **WebSphere-Stammverzeichnis** die Speicherposition der WebSphere-Laufzeitumgebung ein, die Sie als Ihren integrierten Testumgebungsserver verwenden wollen.

Wählen Sie z. B. zum Installieren von DB2 Alphablox in der integrierten Testumgebung von WebSphere 6 bei einer Standardinstallation den Pfad zum Verzeichnis `base_v6` aus. Bei einer typischen Rational Application Developer-Installation würde der Pfad dem folgenden ähneln:

```
C:\Programme\IBM\Rational\SDP\6.0\runtimes\base_v6
```

2. Geben Sie im Fenster mit den WebSphere-Einstellungen einen Namen und ein Kennwort für den WebSphere-Administrator an. Diese Eintragswerte werden nicht in der integrierten WebSphere-Testumgebung verwendet, sie sind jedoch für das DB2 Alphablox-Installationsprogramm erforderlich.

Wichtig: Führen Sie nicht die Installationsabschlusschritte aus, die im *DB2 Alphablox Installationshandbuch* beschrieben sind.

Nach der Installation von DB2 Alphablox in die WebSphere-Laufzeitumgebung, können Sie WebSphere-Serverinstanzen zum Testen von DB2 Alphablox-Anwendungen innerhalb des Rational Developer-Tools erstellen.

Erstellen einer WebSphere-Serverinstanz

Zum Ausführen von Anwendungen oder JSP-Dateien, die DB2 Alphablox-Inhalt enthalten, müssen Sie eine WebSphere-Serverinstanz erstellen, die auf die erforderlichen DB2 Alphablox-Services und Java-Klassen zugreifen kann.

Voraussetzungen: Konfigurieren Sie das Rational Developer-Tool. Installieren Sie das DB2 Alphablox-Toolkit. Installieren Sie DB2 Alphablox in den integrierten WebSphere-Testumgebungen.

Gehen Sie wie folgt vor, um eine WebSphere-Serverinstanz zum Testen von Anwendungen und JSP-Dateien mit DB2 Alphablox-Inhalt zu erstellen:

1. Öffnen Sie das Rational Developer-Tool.
2. Klicken Sie die Indexzunge **Servers** an, falls die Indexzunge nicht sichtbar ist, wählen Sie **Window > Show View > Servers** aus.
3. Klicken Sie das Sichtfenster mit der rechten Maustaste an, und wählen Sie **New > Server** aus. Das Fenster **New Server** wird geöffnet.
4. Geben Sie `localhost` als Hostname ein, und wählen Sie den Servertyp aus. Klicken Sie **Next** an.
5. Geben Sie die Portnummer des Servers ein, und klicken Sie **Next** an.
6. Fügen Sie die verfügbaren Projekte, die Sie auf dieser Serverinstanz ausführen wollen, der Liste der konfigurierten Projekte hinzu. Klicken Sie **Finish** an. Ihre neue Serverinstanz wird in der Sicht **Servers** angezeigt.

Wenn Sie eine Serverinstanz von WebSphere 5.1 erstellt haben, müssen Sie den nächsten Schritt "Erstellen der Serversubstitutionsvariablen von WebSphere 5.1" ausführen. Fahren Sie für Serverinstanzen von WebSphere 6 mit der Anleitung "Hinzufügen des Gastbenutzers zur Gruppe der Administratoren von DB2 Alphablox" fort.

Erstellen der Serversubstitutionsvariablen von WebSphere 5.1

Diese Task beinhaltet das Modifizieren der Serversubstitutionsvariablen für Serverinstanzen von WebSphere 5.1, um DB2 Alphablox in integrierten WebSphere-Testumgebungen ordnungsgemäß auszuführen.

1. Öffnen Sie die Sicht **Servers**, und klicken Sie die zu modifizierende Serverinstanz von WebSphere 5.1 doppelt an.
2. Klicken Sie die Indexzunge **Configuration** für die Serverinstanz an.
3. Wählen Sie **Enable administration console** und **Enable universal test client** aus.
4. Klicken Sie die Indexzunge **Variables** für die Serverinstanz an. Daraufhin wird das Fenster **Substitution Variables** angezeigt.
5. Fügen Sie zwei neue Variablen mit dem Knopf **Add** hinzu. Dieser befindet sich neben der Liste **Node Settings**.
 - a. Fügen Sie eine Variable namens `WS_EAR_AlphabloxPlatform` hinzu, und setzen Sie den Wert auf `$(APP_INSTALL_ROOT)/localhost/AlphabloxPlatform.ear`.
 - b. Fügen Sie eine zweite Variable namens `WS_EAR_AlphabloxStudio` hinzu, und setzen Sie den Wert auf `$(APP_INSTALL_ROOT)/localhost/ApplicationStudio.ear`.
6. Speichern Sie Ihre Änderungen.

Bei Serverinstanzen von WebSphere 5.1 müssen Sie die Schritte in "Konfigurieren der Serverinstanzen von WebSphere 5.1" ausführen, um die Modifikation Ihrer Serverinstanz abzuschließen.

Konfigurieren der Serverinstanzen von WebSphere 5.1

Bei Serverinstanzen von WebSphere 5.1 müssen Sie die Serverinstanzen zum Ausführen von DB2 Alphablox-Anwendungen und -Dateien konfigurieren.

Voraussetzungen: Erstellen Sie die erforderlichen Serversubstitutionsvariablen von WebSphere 5.1.

Gehen Sie wie folgt vor, um Serverinstanzen von WebSphere 5.1 zu konfigurieren:

1. Öffnen Sie die Sicht **Servers** im Rational Developer-Tool, und starten Sie die Serverinstanz, die Sie konfigurieren wollen.
2. Klicken Sie die Serverinstanz mit der rechten Maustaste an, und wählen Sie **Run Administrative Console** aus.
3. Geben Sie in das Feld **ID** keinen Wert ein. Klicken Sie **OK** an.
4. Klicken Sie den Knopf **OK** unterhalb der Konfigurationstabelle an.
5. Öffnen Sie die Navigationssicht **Enterprise Applications**, und klicken Sie den Anwendungsnamen **ApplicationStudio** an.
6. Setzen Sie **Application Binaries** auf `$(WS_EAR_ApplicationStudio)`.
7. Wählen Sie **Use Metadata from Binaries** aus.
8. Klicken Sie den Knopf **OK** unterhalb der Konfigurationstabelle an.
9. Klicken Sie den Knopf **Save** in der WebSphere-Administrationskonsole an.
10. Klicken Sie im Fenster **Save to Master Configuration** die Option **Save** an.
11. Starten Sie die Anwendungen **AlphabloxPlatform** und **ApplicationStudio** in der WebSphere-Administrationskonsole unter **Applications > Enterprise Applications**.

Die Serverinstanz ist konfiguriert. Als Nächstes müssen Sie die Schritte unter "Hinzufügen des Gastbenutzers zur Gruppe der Administratoren von DB2 Alphablox" ausführen.

Hinzufügen des Gastbenutzers zur Gruppe der Administratoren von DB2 Alphablox

Das Hinzufügen des Gastbenutzers zur Gruppe der Administratoren von DB2 Alphablox ermöglicht es Ihnen, während der Verwendung des Rational Developer-Tools auf die DB2 Alphablox-Verwaltungsseiten zuzugreifen.

Voraussetzungen: Erstellen Sie die WebSphere-Serverinstanz. Konfigurieren Sie die WebSphere-Serverinstanz.

Gehen Sie wie folgt vor, um den Gastbenutzer zur Gruppe der Administratoren von DB2 Alphablox hinzuzufügen:

Wichtig: Erteilen Sie dem Gastbenutzer keine Administratorrechte für Ihre WebSphere-Produktionsserver.

1. Melden Sie sich an der DB2 Alphablox-Konsole mit dem folgenden Telnet-Befehl an: `telnet localhost portnummer`. Dabei ist *portnummer* der Port, den Sie während Ihrer DB2 Alphablox-Installation angegeben haben.
2. Geben Sie in die Eingabeaufforderung Ihrer Telnet-Konsole den DB2 Alphablox-Konsolbefehl `set Administrators guest` ein, und drücken Sie die Eingabetaste.
3. Geben Sie in die Telnet-Konsole `save` ein, und drücken Sie dann die Eingabetaste.
4. Schließen Sie die Telnet-Sitzung.

Ihre WebSphere-Serverinstanz ist jetzt einsatzbereit.

Erstellen von DB2 Alphablox-Anwendungen

Wenn Sie neue Anwendungen in Rational Developer erstellen, müssen Sie den erforderlichen DB2 Alphablox-Inhalt hinzufügen, damit die Anwendungen ordnungsgemäß ausgeführt werden.

Voraussetzungen: Installieren Sie das DB2 Alphablox-Toolkit in dem Rational Developer-Tool.

1. Wählen Sie in der Menüleiste des Rational Developer-Tools die Optionen **File > New Project** aus. Der Assistent **New Project** wird geöffnet.
2. Erweitern Sie die Option **Web**, wählen Sie **Dynamic Web Project** aus, und klicken Sie dann **Next** an.
3. Geben Sie einen Namen für Ihr Projekt ein, und klicken Sie den Knopf **Show Advanced** an. Zusätzliche Optionen werden angezeigt.
4. Wählen Sie die entsprechende Servletversion und den entsprechenden Zielsever aus.
5. Klicken Sie **Next** an. Das Fenster **Features options** wird geöffnet.
6. Wählen Sie die Option **DB2 Alphablox Content** aus, und klicken Sie dann **Finish** an.

Ihr neues Webanwendungsprojekt ist jetzt für DB2 Alphablox aktiviert. DB2 Alphablox-Tagbibliotheken und die Blox-Java-API sind verfügbar und die Anwendungsdeskriptordatei (`web.xml`) wurde modifiziert, damit Sie die erforderlichen DB2

Alphablox-Informationen enthält. Sie können jetzt anfangen, Ihrem Projekt die JSP-Dateien mit DB2 Alphablox-Inhalt hinzuzufügen.

Erstellen von JSP-Dateien mit DB2 Alphablox-Inhalt

Diese Task beinhaltet das Erstellen neuer JSP-Dateien mit Zugriff auf Blox-Tagbibliotheken unter Verwendung des Rational Developer-Tools mit dem DB2 Alphablox-Toolkit.

Voraussetzungen: Installieren Sie das DB2 Alphablox-Toolkit in dem Rational Developer-Tool.

1. Wählen Sie in der Sicht **Project Explorer** von Rational Developer die Optionen **File > New > JSP File** aus.
2. Geben Sie einen Dateinamen in das Feld **File Name** ein.
3. Klicken Sie **Configure advanced options** an, und drücken Sie **Next**.
4. Fügen Sie die DB2 Alphablox-Tagbibliotheken hinzu, die Sie in Ihrer JSP-Datei verwenden werden.
 - a. Klicken Sie den Knopf **Add** an, um das Fenster **Add Tag Libraries** zu öffnen.
 - b. Wählen Sie die DB2 Alphablox-Tagbibliotheken aus, die Sie in Ihrer JSP-Datei verwenden wollen.
 - c. Klicken Sie **Next** an.
5. Wählen Sie **ISO 10646/Unicode(UTF-8)** in der Liste **Encoding** aus. Die UTF-8-Codierung ist für DB2 Alphablox-Anwendungen erforderlich, damit sie ordnungsgemäß ausgeführt werden.
6. Klicken Sie **Finish** an. Ihre neue JSP-Datei wird in der Projektliste angezeigt.
7. Klicken Sie den Dateinamen doppelt an, um die Datei im JSP-Editorfenster zu öffnen.
8. Platzieren Sie Ihren Cursor innerhalb des HTML-Tags `<head>`, aber hinter den DB2 Alphablox-JSP-Anweisungen `<taglib>`, die Sie zuvor hinzugefügt haben.
9. Geben Sie den Blox-Header-Tag ein, indem Sie Folgendes in einer neuen Zeile eingeben: `<blox:header/>`. Der Blox-Header-Tag wird für das Hinzufügen der erforderlichen DB2 Alphablox-JavaScript- und CSS-Dateien benötigt, wenn Sie Ihre JSP-Datei ausführen.

Ihre neue JSP-Datei ist jetzt für den Zugriff auf die von Ihnen ausgewählten DB2 Alphablox-Tagbibliotheken aktiviert. Mit der Inhaltshilfefunktion des Rational Developer-Tools können Sie Blox-Tags und Tagattribute für die ausgewählten Tagbibliotheken einfügen.

Zugreifen auf DB2 OLAP Server und Essbase-Datenquellen

Zugreifen auf DB2 OLAP Server- und Essbase-Datenquellen

Sie müssen eine Stapeldatei zum Starten erstellen, die die erforderlichen Clientbibliotheken lädt, damit Sie auf DB2 OLAP Server- oder Essbase-Datenquellen im Rational Developer-Tool zugreifen können, wenn Sie Serverinstanzen von WebSphere 5.1 verwenden.

Voraussetzungen: Installieren Sie das DB2 Alphablox-Toolkit. Installieren Sie DB2 Alphablox in den integrierten Testumgebungen von WebSphere 5.1. Konfigurieren Sie die Serverinstanzen von WebSphere 5.1. Installieren Sie IBM DB2 OLAP Server oder Hyperion Essbase auf Ihrer Entwicklungsmaschine.

Auf Grund der Einschränkungen in den integrierten Testumgebungen von WebSphere 5.1 im Rational Developer-Tool müssen Sie eine Stapeldatei zum Starten des Rational Developer-Tools erstellen, um auf DB2 OLAP Server- oder Hyperion Essbase-Datenquellen zuzugreifen. Wenn Sie das Rational Developer-Tool mit der Stapeldatei starten, werden dem Java-Bibliothekspfad die erforderlichen Essbase-Clientbibliotheken hinzugefügt.

Gehen Sie wie folgt vor, um eine Stapeldatei zu erstellen, die den Zugriff auf DB2 OLAP Server oder Essbase mit Serverinstanzen von WebSphere 5.1 ermöglicht:

1. Erstellen Sie mit Hilfe eines Texteditors ein neues Textdokument.
2. Fügen Sie eine Codezeile hinzu, um die DB2 Alphablox-Datei `aasetup.bat` aufzurufen, die sich in Ihrem DB2 Alphablox-Installationsverzeichnis befindet. Der folgende Code führt z. B. die Datei `aasetup.bat` aus, die sich in der angegebenen DB2 Alphablox-Installation befindet:

```
call C:\alphablox\analytics\bin\aasetup.bat
```
3. Fügen Sie eine zweite Zeile hinzu, um die RAD-Datei `rationalsdp.exe` aufzurufen, die RAD startet. Mit dem folgenden Code wird z. B. das Rational Developer-Tool ausgeführt:

```
call C:\Programme\IBM\Rational\SDP\6.0\rationalsdp.exe
```
4. Speichern Sie diese Datei als `startRAD.bat` auf Ihrem Workstation-Desktop (oder an einer anderen günstigen Speicherposition).

Wenn Sie die Datei `startRAD.bat` doppelt anklicken, wird die DB2 Alphablox-Datei `aasetup.bat` ausgeführt. Außerdem legt sie die erforderlichen Pfad- und Umgebungsvariablen fest. Anschließend wird Rational Application Developer gestartet. Die Version der Essbase-Clientbibliotheken, die mit DB2 Alphablox verwendet wird, muss mit der Version von DB2 OLAP Server (oder Hyperion Essbase) übereinstimmen, die Sie als Datenquelle verwenden. Zum Modifizieren der Essbase-Clientbibliotheken, die mit DB2 Alphablox verwendet werden, führen Sie das Dienstprogramm DB2 OLAP Server / Essbase Client Library Utility (`ChangeEssbase.bat`) aus, das sich im Verzeichnis `db2_alphablox\analytics\bin` befindet. Dabei ist `db2_alphablox` das Stammverzeichnis Ihrer DB2 Alphablox-Installation. Dieses Dienstprogramm modifiziert eine der Stapeldateien, die ausgeführt werden, wenn Sie RAD mit der von Ihnen erstellten Stapeldatei `startRAD.bat` ausführen.

Nachfolgend finden Sie ein vollständiges Beispiel für die Datei `startRAD.bat`, die oben beschrieben wurde:

```
call C:\alphablox\analytics\bin\aasetup.bat
call C:\Programme\IBM\Rational\SDP\6.0\rationalsdp.exe
```

Immer wenn Sie beabsichtigen, auf DB2 OLAP Server- oder Essbase-Datenquellen zuzugreifen, muss das Rational Developer-Tool mit der Datei `startRAD.bat` gestartet werden.

Kapitel 4. Lerntext: Erzeugen eines DB2 Alphablox-Würfels mit DB2 Cube Views

Dieser Lerntext beinhaltet das Erstellen eines DB2 Alphablox-Würfels, der unter Verwendung einer Musterdatenbank von DB2 Cube Views erzeugt wird.

Die Tasks in diesem Lerntext enthalten keine Details zum Erzeugen von angepassten DB2 Alphablox-Würfeln. Ziel ist es stattdessen, Ihnen zu zeigen, wie Sie schnell einen DB2 Alphablox-Würfel erstellen können, der für das Erproben der Funktionalität von DB2 Alphablox Cube Server verwendet werden kann. Die resultierende Datenquelle kann auch zum Testen und Erzeugen von DB2 Alphablox-Anwendungen verwendet werden.

Voraussetzungen:

- Installieren Sie DB2 Alphablox. Details zur Installation finden Sie im *Installationshandbuch*.
- Verschaffen Sie sich Zugriffsberechtigungen für eine unterstützte DB2 Cube Views-Implementierung, in der die Musterdatenbank CVSAMPLE von DB2 Cube Views installiert ist. Details zu unterstützten Versionen von DB2 Cube Views finden Sie im *Installationshandbuch*.

In diesem Lerntext erfahren Sie, wie ein DB2 Alphablox-Würfel auf der Basis der DB2 Cube Views-Musterdatenbank CVSAMPLE erzeugt wird. Sie werden die im Folgenden beschriebenen Tasks ausführen.

Definieren einer relationalen DB2-Datenquelle

Diese Task beinhaltet das Definieren einer Datenquelle in DB2 Alphablox für eine DB2-Datenbank.

Voraussetzungen: DB2 Alphablox muss auf erforderliche DB2-JDBC-Treiber zugreifen können.

Für einen DB2 Alphablox-Würfel ist es erforderlich, dass die zu Grunde liegende relationale Datenquelle als DB2 Alphablox-Datenquelle vordefiniert ist. DB2 Alphablox-Würfel werden mit den Metadaten und Daten generiert, die in den relationalen Datenbanken verfügbar sind.

Gehen Sie wie folgt vor, um eine DB2 Alphablox-Datenquelle für eine DB2-Datenbank zu definieren:

1. Melden Sie sich als Administrator (oder als Benutzer mit Administratorberechtigung) an den DB2 Alphablox-Verwaltungsseiten an.
2. Klicken Sie die Indexzunge **Administration** an.
3. Klicken Sie den Link **Data Sources** an.
4. Klicken Sie den Knopf **Create** an.
5. Wählen Sie im Menü **Adapter** den entsprechenden IBM DB2-JDBC-Treiber für Ihren Datenbankserver aus.

Wählen Sie **IBM DB2 JDBC Type 4 Driver** oder **IBM DB2 UDB on iSeries Driver** aus.

6. Geben Sie in das Feld **Data Source Name** die Zeichenfolge CVSAMPLE als Namen ein, der für Ihre Datenquelle verwendet werden soll.
7. Geben Sie die entsprechenden Werte für **Server Name**, **Port Number** und **Database Name** (dieser sollte CVSAMPLE lauten) ein.

Anmerkung: Wenn Sie Hilfe beim Ermitteln der korrekten Werte für diese Felder benötigen, wenden Sie sich an Ihren Datenbank-administrator.

8. Geben Sie **Default Username** (Standardbenutzernamen) und ein **Default Password** (Standardkennwort) ein.

Der Benutzername und das Kennwort müssen in der relationalen Datenbank gültig sein. Der Standardbenutzername und das Standardkennwort werden immer verwendet, wenn ein DB2 Alphablox-Würfel auf die relationale Datenbank zugreift. Der von Ihnen verwendete Datenbankbenutzer sollte über Lesezugriff für die Datenbank verfügen.

Anmerkung: Der Wert für **Use DB2 Alphablox Username and Password** wird ignoriert, wenn die angegebene relationale Datenquelle verwendet wird, um einen DB2 Alphablox-Würfel aufzufüllen. Zugriffssteuerungslisten (ACLs) können verwendet werden, um den Zugriff auf DB2 Alphablox-Würfel einzuschränken. Informationen zu ACLs finden Sie im Handbuch *Administrator's Guide*.

9. Die Werte für **Maximum Rows** und **Maximum Columns** werden ignoriert, wenn die Datenquelle zum Auffüllen eines DB2 Alphablox-Würfels verwendet wird. Sie können trotzdem Werte eingeben. Diese werden verwendet, wenn andere Anwendungen die Datenquelle verwenden.
10. Setzen Sie den Wert **JDBC Tracing Enabled** auf **No**, es sei denn, Sie wollen JDBC-Protokolldaten in die DB2 Alphablox-Protokolldatei schreiben. Aktivieren Sie die JDBC-Tracefunktion nur, wenn Sie auf Fehler stoßen und Sie deren Ursachen beheben müssen.
11. Klicken Sie den Knopf **Save** an, um die Datenquellendefinition zu speichern.

Sie haben jetzt die DB2 Alphablox-Datenquellendefinition für CVSAMPLE definiert. Jetzt können Sie einen DB2 Alphablox-Würfel für den Zugriff auf die Cube Views-Würfelmetadaten in dieser DB2-Datenquelle definieren.

Definieren einer Alphablox Cube Server-Adapterdatenquelle

Diese Task beinhaltet das Definieren einer DB2 Alphablox-Datenquelle, die den Alphablox Cube Server-Adapter verwendet.

Voraussetzungen: Definieren Sie die DB2 Alphablox-Datenquelle für Ihre DB2-Datenbank CVSAMPLE.

Gehen Sie wie folgt vor, um eine DB2 Alphablox Cube Server-Adapterdatenquelle zu definieren:

1. Melden Sie sich als Administrator (oder als Benutzer mit Administratorberechtigung) an den DB2 Alphablox-Verwaltungsseiten an.
2. Klicken Sie die Indexzunge **Administration** an.
3. Klicken Sie den Link **Data Sources** an.
4. Klicken Sie den Knopf **Create** an.
5. Wählen Sie im Menü **Adapter** die Option **Alphablox Cube Server Adapter** aus.

6. Geben Sie in das Feld **Data Source Name** die Zeichenfolge DB2AlphabloxCubes als Namen für Ihre Datenquelle ein.
7. Klicken Sie den Knopf **Save an**, um Ihre Datenquellendefinition zu speichern.

Sie haben eine DB2 Alphablox-Datenquelle definiert, die für den Zugriff auf DB2 Alphablox-Würfel verwendet werden kann. Als Nächstes müssen Sie einen DB2 Alphablox-Würfel definieren, auf den Sie zugreifen können.

Definieren eines DB2 Alphablox-Würfels

Diese Task beinhaltet das Definieren eines DB2 Alphablox-Würfels auf der Basis der Metadaten eines DB2 Cube Views-Würfels CVSAMPLE.

Voraussetzungen: Definieren Sie die relationale DB2-Datenquelle. Definieren Sie eine DB2 Alphablox Cube Server-Adapterdatenquelle.

Gehen Sie wie folgt vor, um die allgemeinen Merkmale eines DB2 Alphablox-Würfels zu definieren:

1. Melden Sie sich als Administrator (oder als Benutzer mit Administratorberechtigung) an den DB2 Alphablox-Verwaltungsseiten an.
2. Klicken Sie die Indexzunge **Administration** an.
3. Klicken Sie den Link **Cubes** an.
4. Klicken Sie den Knopf **Create** an. Das Fenster **Cube Administration** wird geöffnet.
5. Definieren Sie den neuen Würfel wie folgt:
 - a. Geben Sie in das Feld **DB2 Alphablox Cube Name** die Zeichenfolge CVSales ein.
 - b. Wählen Sie die Option **Enabled** aus, die sich neben dem Feld **DB2 Alphablox Cube Name** befindet. Die Auswahl dieser Option führt dazu, dass Ihr Würfel automatisch gestartet wird, wenn Ihr Server erneut gestartet wird.
 - c. Wählen Sie im Menü **Relational Data Source** den Eintrag MyDB2 aus. Dies ist die relationale Datenquelle, die Sie für diesen Lerntext erstellt haben.
 - d. Wählen Sie die Option **Security Role** nicht aus. Diese Option kann verwendet werden, um die Anzahl der Benutzer zu begrenzen, die auf einen bestimmten Würfel zugreifen können.
6. Aktivieren Sie die DB2 Cube Views-Einstellungen, und geben Sie die zu verwendenden Metadaten an:
 - a. Wählen Sie die Option **Enable DB2 Cube Views Settings** aus.
 - b. Wählen Sie im Menü **Cube Model** die Option **CVSAMPLE.Sales** aus.
 - c. Wählen Sie im Menü **Cube** den Würfel **General Sales** aus.
 - d. Wählen Sie den Radioknopf **Use Business Names** aus, um die entsprechenden Namen anzugeben. Durch die Auswahl dieser Option werden die normalerweise aussagekräftigeren und lesbareren Teildateinamen verwendet.
 - e. Klicken Sie den Knopf **Import Cube Definition** an. Durch die Verwendung dieser Option können Sie eine Würfeldefinition und vordefinierte Kennzahlen und Dimensionen in Ihren DB2 Alphablox-Würfel importieren. Die importierte Würfeldefinition spiegelt den DB2 Cube Views-Würfel so genau wider, wie DB2 Alphablox sie mit Hilfe der DB2 Cube Views-Metadaten abgleichen kann. Wenn Sie mehr Erfahrung in der Arbeit mit Würfeln gesammelt haben, können Sie die Kennzahlen und Dimensionen entsprechend Ihren Anforderungen modifizieren.

- f. Klicken Sie den Knopf **Show Import Log** an, um ein Protokoll anzuzeigen, das die Informations- und Debugnachrichten bezüglich der Importoperation angibt. Dieser Schritt ist nur in diesem Lerntext enthalten, damit Sie sich mit dieser Funktion vertraut machen können.
 - g. Wählen Sie die Option **Import cube definition on start, rebuild, and edit** aus. Diese Option führt dazu, dass Ihr DB2 Alphablox-Würfel, die neueste DB2 Cube Views-Definition verwendet, sobald Ihr DB2 Alphablox-Würfel gestartet, erneut erstellt oder zum Bearbeiten geöffnet wird. Wenn Sie mit DB2 Alphablox und DB2 Cube Views mehr vertraut sind, können Sie die Würfeldefinition importieren und anpassen.
7. Klicken Sie den Knopf **Save** an, um die Würfeldefinition zu speichern.

Sie haben jetzt einen DB2 Alphablox-Würfel definiert. Jetzt können Sie Ihren neuen Würfel CVSales starten.

Starten Ihres DB2 Alphablox-Würfels

Diese Task beinhaltet das Starten des Würfel CVSales unter Verwendung der DB2 Alphablox-Verwaltungsseiten.

Voraussetzungen: Definieren Sie die relationale DB2-Datenquelle. Definieren Sie eine DB2 Alphablox Cube Server-Adapterdatenquelle. Definieren Sie den DB2 Alphablox-Würfel.

Gehen Sie wie folgt vor, um den Würfel CVSales zu starten:

1. Melden Sie sich als Administrator (oder als Benutzer mit Administratorberechtigung) an den DB2 Alphablox-Verwaltungsseiten an.
2. Klicken Sie die Indexzunge **Administration** an.
3. Klicken Sie unter dem Abschnitt **Runtime Management** den Link **Cubes** an.
4. Wählen Sie im Menü **DB2 Alphablox Cubes** den DB2 Alphablox-Würfel aus, den Sie starten wollen.
5. Klicken Sie den Knopf **Start** an. Wenn der Würfel gestartet ist, wird im Statusfeld **Running** angezeigt.

Jetzt verfügen Sie über einen aktiven DB2 Alphablox-Musterwürfel. Sie können jetzt mit dem Erzeugen von Anwendungen mit dem DB2 Alphablox-Würfel beginnen, den Sie gerade erstellt haben. Sie können eine schnelle Überprüfung Ihres neuen Würfels durchführen, indem Sie die Anwendung Query Builder verwenden, Ihren Würfel als Datenquelle auswählen und MDX-Abfragen für den Würfel ausführen.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. An Stelle der Produkte, Programme oder Services können auch andere ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb von Fremdprodukten, Fremdprogrammen und Fremdservices liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen oder in Technical News Letters (TNLs) bekannt gegeben. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt; die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Corporation, J46A/G4, 555 Bailey Avenue, San Jose, CA 95141-1003 USA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Die oben genannten Erklärungen bezüglich der Produktstrategien und Absichtserklärungen von IBM stellen die gegenwärtige Absicht der IBM dar, unterliegen Änderungen oder können zurückgenommen werden, und repräsentieren nur die Ziele der IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Musterprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Marken

Folgende Namen sind in gewissen Ländern Marken oder eingetragene Marken der International Business Machines Corporation:

DB2
IBM

DB2 OLAP Server
WebSphere

DB2 Universal Database

Alphablox und Blox sind in gewissen Ländern Marken oder eingetragene Marken der Alphablox Corporation.

Java und alle Java-basierten Marken sind in gewissen Ländern Marken von Sun Microsystems, Inc.

Linux ist in gewissen Ländern eine Marke von Linus Torvalds.

Andere Namen von Unternehmen, Produkten oder Services können Marken oder Servicemarken anderer Unternehmen sein.

Index

A

Abfragen
 unter Verwendung von Query Builder
 generieren 4

L

Lerntexte
 Anwendungsentwicklung 1

P

Portlet
 JSP-Struktur 15
 Muster 13

Q

Query Builder
 verwenden 4
Query Builder, DHTML
 verwenden 4



Programmnummer: 5724-L14

GC12-3572-01

