

IBM DB2 Alphablox



Guía de iniciación

Versión 84

IBM DB2 Alphablox



Guía de iniciación

Versión 84

Nota:

Antes de utilizar esta información y el producto al que da soporte, lea la información del apartado "Avisos" en la página 33.

Segunda edición (marzo de 2006)

Esta publicación es la traducción del original inglés *IBM DB2 Alphablox Version 8.4 Getting Started*, (GC18-9607-01).

Esta edición se aplica a la versión 8, release 4, de IBM DB2 Alphablox para Linux, UNIX y Windows (número de producto 5724-L14) y a todos los releases y modificaciones posteriores hasta que se indique lo contrario en nuevas ediciones.

Cuando envía información a IBM, otorga a IBM el derecho no exclusivo de utilizar o distribuir esa información de la manera que crea más apropiada sin incurrir en ninguna obligación con usted y sin restringir el uso que pueda hacer usted de ella.

Copyright © 1996 - 2006 Alphablox Corporation. Reservados todos los derechos.

© Copyright International Business Machines Corporation 1996, 2006. Reservados todos los derechos.

Contenido

Capítulo 1. Guía de aprendizaje:

Creación de su primera aplicación 1

Definición de la aplicación.	1
Acceso a los datos	2
Creación de la página inicial de la aplicación	2
Configuración de una página inicial por omisión	3
Creación de la primera vista analítica	4
Estructura de la página de su primera vista analítica	7
Creación de la segunda vista analítica	9
Resumen	10

Capítulo 2. Guía de aprendizaje:

Creación de su primer portlet con componentes Blox. 11

Instalación de los portlets de ejemplo.	11
Ejecución de los portlets de ejemplo	12
Examen de la estructura de una página JSP del portlet con un componente Blox	12
Creación de su propia página JSP de portlet con componentes Blox	14
Creación de un proyecto de portlet para utilizar componentes Blox	16
Configuración de un proyecto de portlet utilizando Rational Application Developer	17
Siguientes pasos	18
Consejos sobre el desarrollo de portlets	18

Capítulo 3. Guía de aprendizaje:

Creación de aplicaciones utilizando herramientas de Rational Developer 21

Preparación del entorno de desarrollo	21
Instalación de DB2 Alphablox Toolkit.	21
Instalación de DB2 Alphablox en entornos de prueba integrados de WebSphere	22
Creación de una instancia de servidor WebSphere	23
Creación de variables de sustitución de servidor WebSphere 5.1	24
Configuración de instancias de servidor de WebSphere 5.1	24
Adición del usuario huésped al grupo Administradores de DB2 Alphablox	25
Creación de aplicaciones de DB2 Alphablox	25
Creación de archivos JSP con contenido DB2 Alphablox	26
Acceso a fuentes de datos de DB2 OLAP Server y Essbase.	26

Capítulo 4. Guía de aprendizaje:

Creación de un cubo de DB2 Alphablox utilizando DB2 Cube Views 29

Definición de una fuente de datos relacionales DB2	29
Definición de una fuente de datos de Alphablox Cube Server Adaptor	30
Definición de un cubo de DB2 Alphablox	31
Inicio de un cubo DB2 Alphablox	32

Avisos 33

Marcas registradas	35
------------------------------	----

Índice 37

Capítulo 1. Guía de aprendizaje: Creación de su primera aplicación

Esta guía de aprendizaje le enseña los conocimientos básicos importantes sobre qué son los componentes Blox y proporciona unos pasos que puede seguir para crear rápidamente una aplicación de DB2 Alphablox.

Para crear aplicaciones de DB2 Alphablox, debe utilizar tecnología JavaServer Pages (JSP), pero no se requiere ningún conocimiento de JSP para esta guía de aprendizaje. Esta guía de aprendizaje explica los aspectos de JSP que necesita conocer para empezar a crear aplicaciones analíticas. Para saber más acerca de la tecnología JSP, existen muchos libros excelentes disponibles.

Prerrequisitos

En los pasos de esta guía de aprendizaje se da por supuesto que utiliza DB2 Alphablox en WebSphere Application Server. Si está utilizando BEA WebLogic o Apache Tomcat como servidor de aplicaciones, es posible que necesite modificar algunos de los pasos siguientes para adecuar diferencias concretas en el servidor.

Además, antes de empezar esta guía de aprendizaje, necesita instalar un cubo de DB2 Alphablox (qcc_2003) y su fuente de datos relacional subyacente (qcc2003-rdb). Los archivos y las instrucciones de base de datos de ejemplo (en los archivos readme.txt) sobre la configuración del cubo de DB2 Alphablox y la base de datos relacional necesaria para esta guía de aprendizaje se pueden encontrar en el directorio `sampledata/qcc/acs` del CD de instalación de DB2 Alphablox.

Definición de la aplicación

Para crear una aplicación utilizando el enfoque de desarrollo J2EE, debe crear una estructura de directorios con un directorio WEB-INF que contenga un archivo descriptor de aplicaciones (web.xml). La forma más sencilla de crear esta estructura en DB2 Alphablox es crear una nueva aplicación utilizando la página Aplicación dentro de las páginas de administración de DB2 Alphablox.

Para crear la aplicación y la carpeta:

1. Cree una nueva aplicación denominada MiApl en DB2 Alphablox siguiendo los pasos que se describen en la sección Definiciones de aplicación de la publicación *Administrator's Guide*.
 - Entre MiApl en el campo **Nombre**.
 - Entre Mi Apl (con un espacio) en el campo **Nombre de visualización**, que define la etiqueta para la aplicación que aparece en la lista de la página Aplicaciones.
2. Pulse la pestaña superior **Aplicaciones** en la parte superior izquierda de la página. Se abre una lista de las aplicaciones disponibles.
3. Pulse el nombre de la aplicación recientemente creada (Mi Apl) en la lista de aplicaciones. Puesto que todavía no ha creado ningún archivo, el directorio de archivos está vacío.
4. Navegue hasta la carpeta de la nueva aplicación en el servidor de aplicaciones (para WebSphere, la carpeta está en el directorio de WebSphere `installedApps`).

Observe que la aplicación incluye un directorio WEB-INF con un archivo web.xml y el directorio tlds. El archivo web.xml define la información de la aplicación. El directorio tlds incluye el archivo descriptor de la biblioteca de códigos Blox (blox.tld), que define los códigos Blox que utilizará para crear vistas analíticas en esta guía de aprendizaje. El directorio contiene otros archivos de descriptor de la biblioteca de códigos (TLD) para otras bibliotecas de códigos Blox (no utilizadas en esta guía de aprendizaje), que incluyen bloxform.tld, bloxlogic.tld, bloxreport.tld y bloxui.tld.

Ha creado una infraestructura de aplicaciones de DB2 Alphablox a la que ahora se pueden añadir vistas analíticas.

Acceso a los datos

Para esta guía de aprendizaje, debe utilizar la fuente de datos del cubo de DB2 Alphablox que ha creado antes de empezar esta guía de aprendizaje. La base de datos relacional y Alphablox Dube Server deben estar en ejecución. Además, debe estar en ejecución el cubo de DB2 Alphablox que se está utilizando.

Para verificar que las fuentes de datos necesarias están disponibles y en funcionamiento:

1. Abra el navegador en las páginas de administración de DB2 Alphablox.
2. Pulse sobre la pestaña **Administración**.
3. Pulse **Fuentes de datos**.
4. En la página Fuentes de datos, busque la fuente qcc2003-rdbdata en el menú del lado izquierdo de la página. Si no está ahí, es necesario que revise los pasos del archivo readme.txt del directorio sampledata/qcc/acs del CD de instalación de DB2 Alphablox.
5. Pulse en qcc2003-rdb en la lista de fuentes de datos disponibles y, a continuación, pulse el botón **Probar fuente de datos seleccionada**. Si la base de datos se está ejecutando y la definición de la fuente de datos está configurada correctamente, aparecerá un mensaje de resultado satisfactorio. Si aparece un mensaje de error, es necesario que compruebe la definición de la fuente de datos (seleccionando la fuente de datos y pulsando el botón **Editar**) o si la base de datos está correctamente configurada con un usuario que tiene derechos de acceso a la base de datos.
6. Seleccione la fuente de datos qcc2003-acs de la lista de fuentes de datos disponibles y, a continuación, pulse el botón **Probar fuente de datos seleccionada**. Si la fuente de datos de Alphablox Cube Server está correctamente configurada, aparecerá un mensaje de resultado satisfactorio. Si aparece un mensaje de error, revise la definición de la fuente de datos.

Después de probar satisfactoriamente las fuentes de datos, está preparado para saber cómo crear un par de vistas analíticas sencillas de DB2 Alphablox.

Creación de la página inicial de la aplicación

Para esta tarea, puede crear una página inicial sencilla utilizando funciones básicas de Web.

Para el propósito de esta guía de aprendizaje, sólo necesita crear una estructura de aplicación web muy simple. En la página inicial, necesita crear dos enlaces para acceder a las dos vistas analíticas que crea en esta guía de aprendizaje.

Para crear una página inicial simple:

1. Utilizando un editor de texto o su herramienta de desarrollo favorita, llámela `index.html`. En esta guía de aprendizaje, la página será muy sencilla.
2. Edite el archivo para especificar un título de aplicación (por ejemplo, “Mi aplicación de DB2 Alphablox”) y dos enlaces a las dos vistas analíticas que creará.

Copie y pegue, o escriba, el siguiente código en el archivo `index.html`.

```
<html>
<head>
<title>Mi aplicación de DB2 Alphablox</title>
</head>
<body>
<h2>Mi aplicación de DB2 Alphablox</h2>
<p>
<a href="PresentBloxView.jsp">Vista simple PresentBlox</a>
</p>
<p>
<a href="ChartView.jsp">Vista de diagrama personalizada</a>
</p>
</body>
</html>
```

3. Guarde el archivo en el directorio `MiApl` que ha creado al definir la aplicación.
4. Abra el navegador Web en la página inicial de DB2 Alphablox. Por omisión, el navegador muestra la pestaña **Aplicaciones**.
5. Pulse el enlace **Mi Apl** para verificar que el archivo se ha incluido en el listado de directorios. Pulse el enlace **index.html** para ver la página inicial. Si DB2 Alphablox está instalado en IBM WebSphere Application Server o en un servidor BEA WebLogic, el listado de directorios debe estar habilitado para que pueda realizar este paso. Para instalaciones de DB2 Alphablox en Apache Tomcat, no es necesario realizar ningún paso adicional. Si ve un error HTTP 403 (por omisión, los listados de directorios no están permitidos), tendrá que modificar el servidor para permitir los listados de directorios o simplemente añadir el nombre de archivo `miapl.html` al final del URL (por ejemplo, `http://nombre_sist_pral:9080/MiApl/miapl.html`). Cuando se utiliza la dirección completa del enlace, se debe mostrar la nueva página.

Ahora ya tiene una página inicial con enlaces a las dos vistas analíticas que creará posteriormente.

Configuración de una página inicial por omisión

En pasos anteriores, ha accedido a la página inicial directamente pulsando sobre el enlace del nombre de archivo en el directorio de la aplicación `MiApl`, o bien especificando el archivo de la página inicial, `index.html`, en el URL. En este paso, se especifica la página inicial por omisión para la aplicación.

Para establecer la página inicial por omisión de la aplicación:

1. Abra el navegador en la página inicial de DB2 Alphablox.
2. Pulse la pestaña **Administración**.
3. Pulse el enlace **Aplicaciones**.
4. Seleccione **MiApl** en la lista de aplicaciones y pulse el botón **Editar** que hay bajo la lista
5. En la página **Editar aplicación** correspondiente a `MiApl`, entre `index.html` en el campo **URL inicial**.

6. Pulse el botón **Guardar** para guardar el cambio realizado.
7. Pulse la pestaña principal Aplicaciones, la pestaña de la carpeta superior izquierda, para volver a la página Aplicaciones.
8. Pulse el nombre de aplicación **Mi Apl**. La aplicación se abre ahora directamente en la página inicial definida.

Ahora que la página inicial está definida, el servidor sabrá dónde iniciar la aplicación y le indicará que se abra en esta página, incluso si el usuario no especifica la página inicial en su URL. Por lo tanto, si un usuario entra `http://NombreServidor/MiApl/` en la barra de direcciones del navegador, la aplicación se abrirá automáticamente en la página inicial definida, `index.html`.

En los temas siguientes, empezará a crear vistas analíticas de DB2 Alphablox utilizando códigos simples Blox generados utilizando la aplicación DB2 Alphablox Query Builder.

Creación de la primera vista analítica

En esta tarea, creará una vista analítica de DB2 Alphablox que muestra una cuadrícula y un diagrama combinados en el componente PresentBlox.

Requisito previo: lea las secciones de introducción de la publicación *Developer's Guide* para obtener información sobre DB2 Alphablox y sus componentes. Esto le proporcionará unos conocimientos básicos de la funcionalidad de DB2 Alphablox y de cómo las bibliotecas de códigos personalizados de Blox pueden utilizarse en la creación de aplicaciones analíticas utilizando tecnología JSP.

Para la primera vista analítica que creará, utilizará la aplicación DB2 Alphablox Query Builder para generar un código personalizado JSP definido para un componente PresentBlox, el componente Blox más popular utilizado para presentar datos. Un componente PresentBlox puede incluir vista de cuadrícula y de diagrama, un panel de diseño de datos, barras de herramientas y otros componentes anidados. Las vistas de cuadrícula y de diagrama representan vistas alternativas sincronizadas de los mismos datos.

Para crear la primera vista analítica utilizando Query Builder:

1. Abra un navegador en las páginas de administración DB2 Alphablox. Por omisión, el código Applications es visible.
2. Pulse en DB2 Alphablox Query Builder para abrir la aplicación.
3. Pulse el botón **Valores de la conexión**. Aparecerá la ventana Conexión de la base de datos.
4. Seleccione la fuente de datos `qcc2003-acs` y, a continuación, pulse el botón **Connect**. Esta ventana se cerrará y deberá ver "Conectado" en la sección Estado de la base de datos de la página Query Builder.

Nota: No es necesario que rellene los valores de los campos **Esquema**, **Catálogo**, **Nombre de usuario**, y **Contraseña** ya que se recuperan automáticamente de la definición de la fuente de datos para la fuente de datos que selecciona. Además, no seleccione el recuadro de selección **Ejecutar consulta por omisión**.

5. Busque la sección Consulta en la página, debajo de la sección Estado. Tenga en cuenta las características siguientes:

- Para esta guía de aprendizaje, está utilizando un cubo de DB2 Alphablox (qcc2003-acs). Justo debajo de la cabecera de la sección Consulta, debe ver un listado de cubos disponibles (actualmente en ejecución). El cubo qcc2003-acs debe aparecer en la lista.
- Una serie de consulta por omisión asociada con esta fuente de datos aparece en la ventana de texto. Debido a que la fuente de datos es un cubo de DB2 Alphablox, el primer cubo, listado alfabéticamente, se añade a la sentencia de consulta (por ejemplo, "select de [datasourceName]").

Nota: Cuando la característica de agrupamiento de conexiones de DB2 Alphablox Cube Server esté habilitada, no podrá utilizar el botón **Obtener consulta por omisión** para obtener una consulta por omisión para un cubo de DB2 Alphablox.

- - Cuando aparezcan datos en el PresentBlox en la parte inferior de la página, utilice la interfaz de usuario estándar para conmutar los ejecutar, detallar más o menos, mover las dimensiones entre ejes, etc.
 - Para ver los resultados de la consulta, pulse el botón Examinar consulta. El conjunto de resultados aparece en el PresentBlox en la parte inferior de la página.
6. Escriba qcc2003-acs entre los corchetes de la consulta por omisión (por ejemplo, select from [qcc2003-acs]) y, a continuación, pulse el botón **Ejecutar consulta**.
La sección PresentBlox debe mostrar una vista por omisión basada en la consulta.
 7. Utilice la interfaz de usuario de PresentBlox para crear una vista nueva.
Por ejemplo, puede arrastrar la dimensión de Productos al eje Fila del panel Diseño de datos de la izquierda. A continuación, puede arrastrar la dimensión Tiempo (Calendario) al eje Columna. A continuación, es posible que desee arrastrar la dimensión Medidas al eje Página, dando como resultado que aparezca un menú Medidas en el panel Página, que muestra filtros de páginas, encima del área de visualización. Tome otras dos dimensiones y añádalas al eje Página de forma que aparezcan tres filtros de página encima de la pantalla. Si es nuevo en la utilización de la interfaz de usuario de DB2 Alphablox, seleccione **Ayuda** en la barra de herramientas de PresentBlox para saber más acerca de cómo utilizar la interfaz.
 8. Cuando haya acabado de crear la nueva vista, vea la sentencia de consulta que aparece en la sección Consulta. Si la consulta no está siendo generada dinámicamente (cuando la opción Actualizar consultas automáticamente está deseleccionada), puede pulsar el botón **Obtener consulta actual** para obtener la consulta actual. El recuadro de texto visualiza la sentencia de consulta necesaria para desarrollar la vista de datos actual. Puede copiar y pegar sentencias de consulta del campo Consulta para utilizar en el desarrollo de aplicaciones DB2 Alphablox.
 9. Pulse el botón **Generar código Blox**. Se abrirá un recuadro de diálogo que muestra el código (y los códigos anidados) para reproducir el diseño de PresentBlox y el conjunto de resultados.
 10. Abra un nuevo archivo en el editor de texto y añada el texto siguiente al archivo; a continuación, guarde el archivo como PresentBloxView.jsp. El tema siguiente explicará muchos de los elementos importantes del código de esta página.

```

<%@ taglib uri="bloxtld" prefix="blox" %>
//Copie y pegue la información de código del código Blox generado
//inmediatamente debajo de esta línea, quitando los saltos de línea
//de la sentencia de consulta. También puede eliminar este comentario
//después de añadir el código de DB2 Alphablox Query Builder.
<html>
<head>
<blox:header/>
</head>
<body>
<h2>Vista simple de PresentBlox</h2>
<p>
<blox:display bloxRef="MyPresentBloxView"/>
</p>
</body>
</html>

```

11. Copie el archivo guardado (PresentBloxView.jsp) en el directorio MiApl del servidor y pruebe la nueva vista analítica pulsando en el enlace a esta página, Vista simple de PresentBlox, en la página inicial MiApl. De forma opcional, puede entrar el URL directamente en una dirección del navegador (por ejemplo, http://sist_pral_local:9080/MiApl/PresentBloxView.jsp). La vista que verá debe tener un aspecto similar al PresentBlox que ha manipulado en DB2 Alphablox Query Builder.

A continuación se muestra un ejemplo de código de una versión completa de la página Vista simple de PresentBlox (PresentBloxView.jsp):

```

<%@ 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>Vista simple de PresentBlox</h2>
<blox:display bloxRef="queryBuilder4_present"/>
</p>
</body>
</html>

```

En el anterior ejemplo de código, la sentencia de consulta contiene saltos de línea para facilitar su lectura. La correcta ejecución de esta página requiere que se eliminen los saltos de línea en los valores de atributo (como la consulta y los atributos de selectableSlicerDimensions anteriores). En el código generado por DB2 Alphablox Query Builder, muchos códigos que se incluyen no son necesarios. Sin embargo, ver todos los códigos aquí le debería proporcionar una visión de cómo funcionan los códigos Blox como códigos padre y anidados. Además, puede ver la utilización de los atributos de códigos que se añaden como resultado de las manipulaciones que realice en PresentBlox de Query Builder. Para saber más acerca de los detalles de la utilización de los códigos Blox, de los atributos de códigos y sobre cómo desarrollar aplicaciones de DB2 Alphablox, puede consultar el Centro de información de DB2 Alphablox.

Estructura de la página de su primera vista analítica

Este tema resume la estructura de la primera vista analítica (PresentBloxView.jsp) que ha creado en el tema anterior.

A continuación se presenta una breve visión general de algunos de los aspectos más destacados de la página creada en el tema anterior. Para comprender de forma más completa el código, debe leer los temas de interés del Centro de información de DB2 Alphablox.

La página PresentBloxView.jsp empieza con la línea siguiente:

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

Esta línea es una directiva taglib de JSP que informa al servidor de que el usuario tiene la intención de utilizar la biblioteca de códigos Blox. El uri apunta al archivo descriptor de bibliotecas de códigos. El valor prefix, definido como blox, indica al servidor que busque códigos en esta página que empiecen con blox y, a continuación, que procese el contenido de la biblioteca de códigos Blox tal y como está definido en el archivo descriptor de la biblioteca de códigos.

Justa encima de la directiva taglib, se especifica el componente PresentBlox mediante los códigos siguientes y sus atributos de código:

```

<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>

```

El código `<blox:present>` especifica que desea que `PresentBlox` aparezca aquí con un valor de atributo `id` de `queryBuilder4_present`. El atributo `id` le permite identificar este Blox en particular para propósitos de creación de scripts. Tenga en cuenta que el código `<blox:present>` incluye un conjunto de atributos `visible` establecidos en `false`. Cuando esté establecido en `false`, `PresentBlox` no se interpretará hasta que el compilador encuentre el código `<blox:display>` en la página.

Existen muchos atributos disponibles para cada Blox definido utilizando códigos, pero a menos que necesite especificar valores de atributos distintos a los valores por omisión, no es necesario incluirlos en el código. En este ejemplo, un valor `height` de 500 y un valor `width` de 100%. La altura y la anchura se pueden definir en píxeles o en porcentajes.

Se incluye un `DataBlox` anidado, que define el atributo `dataSourceName` como `qc2003-acs`. Si no se especifica ninguna fuente de datos, verá un mensaje "No hay datos disponibles" cuando se visualice la página. El atributo de consulta incluye la sentencia de consulta MDX generada por `DB2 Alphablox Query Builder`. Cuando entienda el lenguaje de consulta MDX utilizado con `DB2 Alphablox Cube Server`, podrá simplificar la consulta, acortándola. Sin embargo, aquí la sentencia de consulta está completa y generará la vista que ha creado en la aplicación `Query Builder`.

La sección `<head>` de la página contiene un código Blox especial para añadir código importante a la página antes de que ésta se interprete:

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

Este código lo utiliza `DB2 Alphablox` para añadir automáticamente el código HTML, JavaScript y CSS necesario a la sección de cabecera de la página. Cuando el servidor interpreta una página, este código añade enlaces CSS a temas HTML definidos y un metacódigo para evitar la colocación en antememoria. Recuerde

entrar este código en cada página JSP que utilice componentes Blox. Si este código no se incluye, la página no se interpretará correctamente.

En la sección <body> de la página se encuentra la línea siguiente:

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

Este código Blox da como resultado que se proporcione aquí el PresentBlox especificado anteriormente. Esto le permite realizar cambiar en un componente Blox o en sus componentes anidados en un lugar, donde también puede tener otros códigos Blox y scriptlets JSP que controlen el comportamiento de los componentes Blox. En archivos JSP complejos, y conforme se familiarice con DB2 Alphablox, podrá elegir tener un componente DataBlox separado de los códigos PresentBlox y añadir scriptlets JSP que modifiquen los componentes Blox y su comportamiento antes de que se entregue el componente Blox en el navegador web.

En resumen, los códigos Blox más importantes de esta página incluyen:

<blox:header/>, <blox:present> y <blox:display>. Estos tres códigos, y los códigos anidados, especifican una vista analítica sin necesitar ningún código Java en la página. La complejidad de la lógica de presentación está siendo gestionado por estos códigos Blox. Añadiendo un Blox anidado y modificando los valores de atributo, puede personalizar las vistas en base a sus necesidades comerciales.

Para vistas y aplicaciones más complejas, puede que se requieran componentes de JavaBeans y código Java adicionales.

Creación de la segunda vista analítica

En esta tarea, creará una vista analítica que visualizará sólo un diagrama con filtros de página, en base al código de la primera vista analítica de esta guía de aprendizaje.

En muchas aplicaciones los datos se representan utilizando simplemente una cuadrícula o un diagrama. Mientras que el componente PresentBlox combina tanto GridBlox como ChartBlox como componentes anidados que se visualizan conjuntamente dentro del componente PresentBlox, puede inhabilitar u ocultar los componentes anidados que no desee que se vean. Utilizando un componente PresentBlox para crear una vista de diagrama con filtros de página es relativamente simple, y se consigue modificando los atributos de códigos en algunos de los códigos Blox anidados.

En esta tarea, reutilizará el código PresentBlox de la primera vista analítica para crear una segunda vista analítica personalizada.

Para crear una vista analítica que muestra un diagrama con filtros de página:

1. Abra el archivo PresentBloxView.jsp creado anteriormente en el editor y guárdelo como CustomizedChartBlox.jsp.
2. En los componentes anidados del código de PresentBlox que se listan a continuación, añada un atributo `visible` a cada uno de los códigos y establezca el valor en `false`: La adición de este atributo hará que estos componentes anidados no aparezcan, dejando una vista de diagrama con una barra de filtro de página.

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

3. Modifique el código PageBlox (<blox:page>), añadiendo el atributo labelPlacement y estableciendo el valor en top. Por omisión, la etiqueta para filtros de página aparece a la izquierda de los menús. Si se añade este atributo y se establece en top, se modificará el comportamiento por omisión y colocará las etiquetas encima de los filtros de página.

```
<blox:page labelPlacement="top" />
```
4. Modifique el nombre del atributo id del código <blox:present> por el valor siguiente: chartview. Cada componente Blox proporcionado, o mostrado, en una página necesita tener un ID exclusivo a fin de evitar errores.
5. Guarde los cambios y añada el archivo al directorio MiApl del servidor:
6. Abra el navegador web y prueba la nueva página, pulsando en el enlace de la página inicial o accediendo a la misma directamente utilizando el URL (por ejemplo, <http://localhost:9080/MiApl/CustomizedChartView.jsp>)

Ahora ha creado dos vistas analíticas, utilizando la aplicación DB2 Alphablox Query Builder para ayudarle a empezar a utilizar el producto. Además, ha aprendido que modificando o añadiendo atributos a los códigos existentes, puede controlar el aspecto y el comportamiento de Blox y de sus componentes anidados.

Resumen

Si ha finalizado todas las tareas de la guía de aprendizaje, ha aprendido cómo crear una aplicación básica de DB2 Alphablox utilizando las páginas de administración de DB2 Alphablox, la aplicación DB2 Alphablox Query Builder y los códigos Blox. Si ya ha leído la publicación *Developer's Reference*, sabrá que hay muchas propiedades y métodos disponibles para definir y manipular Blox en páginas JSP. ¿Qué hacer a continuación?

Si desea ver los datos de la empresa en una vista de DB2 Alphablox, ahora ya sabe cómo crear una aplicación partiendo de cero y cómo añadir vistas analíticas a la aplicación. Si desea ver algo inmediatamente, puede realizar algunas modificaciones sencillas a la aplicación MiApl. Para visualizar los datos de las bases de datos de la empresa en lugar de utilizar la fuente de datos qcc2003-acs, puede crear fuentes de datos que apunten a bases de datos de la empresa, modifique los atributos de dataSourceName para que apunten a la fuente de datos definida recientemente y, a continuación, añada un atributo query adecuado. Utilizando DB2 Alphablox Query Builder, puede a menudo crear rápidamente sentencias de consulta utilizables y códigos Blox. Para obtener detalles acerca de cómo crear fuentes de datos, consulte la publicación *Administrator's Guide*. Para saber cómo crear consultas adecuadas y mucho más sobre la creación de aplicaciones de DB2 Alphablox, consulte la sección Retrieving Data de la publicación *Developer's Guide* ya la sección DataBlox de la publicación *Developer's Reference*.

Capítulo 2. Guía de aprendizaje: Creación de su primer portlet con componentes Blox

En esta guía de aprendizaje, aprenderá a añadir componentes Blox a los portlets. Hará lo siguiente:

1. Instalará un portlet de ejemplo creado previamente. Con este paso, puede ver rápidamente cómo se añade un componente Blox a una página JSP de portlet y cómo ésta se visualiza en una página del portal.
2. Escribirá su propia página JSP con un GridBlox.

Las tareas en esta guía de aprendizaje no proporcionan detalles sobre el desarrollo general de portlets. Esta guía de aprendizaje se centra en las tareas específicas de DB2 Alphablox y presupone cierta familiaridad con los conceptos generales del desarrollo de portales y de portlets.

Prerrequisitos

- DB2 Alphablox se debe instalar en un servidor WebSphere Portal Versión 5.1. Consulte la publicación *Installation Guide* para obtener detalles sobre la instalación.
- El servidor WebSphere Portal debe estar iniciado.
- Debe tener acceso administrativo al servidor WebSphere Portal.
- Debe estar familiarizado con las funciones administrativas y la interfaz de usuario en WebSphere Portal.
- Debe tener un conocimiento básico de Java y JSP.
- Debe tener un editor de JSP instalado.

Aunque puede utilizar cualquier editor de JSP o incluso un editor de texto para esta guía de aprendizaje, cuando desarrolle sus propios portlets, utilice una herramienta de desarrollo recomendada por WebSphere Portal, como por ejemplo Rational Application Developer.

Esta guía de aprendizaje utiliza una fuente de datos predefinida que se ha instalado con DB2 Alphablox. Puede utilizar esta fuente de datos para desarrollar rápidamente una aplicación básica. Para esta guía de aprendizaje no es necesario preocuparse sobre la configuración de una fuente de datos personalizada.

Instalación de los portlets de ejemplo

La mejor forma de saber cómo añadir un Blox al portlet es instalar los portlets de ejemplo proporcionados en DB2 Alphablox y cargarlos en una página del portal. Esto le permite examinar la estructura básica del código JSP y hacerla coincidir con la salida en el portal.

Instale los portlets de ejemplo proporcionados en DB2 Alphablox:

1. Abra el navegador e inicie sesión en el portal como usuario administrador (el URL está en el formato `http://<ServidorPortal>:<port>/wps/portal`).
2. Pulse el botón **Administración**.
3. En la sección Gestión de portlets, pulse **Módulos Web**. La página Gestionar módulos Web aparece a la derecha.

4. Pulse **Instalar**. Se le solicitará el módulo Web que desea instalar.
5. Pulse el botón **Examinar** y navegue hasta el directorio `installableApps` en el directorio de instalación de DB2 Alphablox.
6. Seleccione `AlphabloxSamplePortlets.war` y pulse **Siguiente**. En la tabla Aplicación de portlets se visualiza la aplicación Portlets de ejemplo de DB2 Alphablox con un portlet llamado Portlet de ejemplo de la página JSP de DB2 Alphablox.
7. Pulse **Finalizar (Finish)**.

La aplicación Portlets de ejemplo de DB2 Alphablox y los portlets incluidos están ahora instalados. Compruebe el directorio `installedApps` en la instalación de WebSphere Portal. El nombre del directorio recientemente creado empieza por DB2 Alpha y acaba con un ID de portlet generado dinámicamente en el patrón de `_PA_x_x_xx.ear`.

Ejecución de los portlets de ejemplo

Para ejecutar los portlets de ejemplo en la página del portal:

1. Vaya a la página del portal.
2. Cree o edite una página existente. Puede crear una nueva página para probar este portlet de ejemplo, o bien puede pulsar una página existente para editar la página.
3. Pulse uno de los botones **Añadir portlets** de la página de diseño de portal.
4. Escriba DB2 en el campo Buscar y pulse **Buscar**. Aparece el recuadro de selección **Portlet de ejemplo de la página JSP de DB2 Alphablox**.
5. Seleccione el recuadro de selección y pulse **Aceptar**.
6. Pulse **Finalizado**.

Cuando se renueva la página del portal, aparece un PresentBlox en la página del portal. Este PresentBlox tiene:

- Una barra de menús en la parte superior
- Dos paneles de barra de herramientas debajo de la barra de menús
- Un panel de diseño de datos a la izquierda que le permite mover las dimensiones entre los diferentes ejes
- Una cuadrícula que muestra los datos en formato tabular
- Un diagrama de barras tridimensional a la derecha

Este archivo `present.jsp` es la página por omisión que se debe cargar de la forma especificada en el servlet `BloxJSPPagePortlet` incluido en el ejemplo. El código fuente para este servlet de ejemplo está disponible en el directorio `WEB-INF/src/`.

Ha instalado satisfactoriamente los portlet de ejemplo y añadido un portlet con Blox en la página del portal. La siguiente tarea es examinar la estructura del código en esta página JSP.

Examen de la estructura de una página JSP del portlet con un componente Blox

En esta tarea, se revisa la estructura del código de una página JSP que contiene un Blox. Todas las páginas JSP deben tener los mismos elementos clave.

Para abrir un archivo JSP:

1. Navegue hasta el directorio `installedApps/` en la instalación de WebSphere Portal y localice la carpeta de la aplicación creada recientemente que empieza por DB2 Alpha.
2. Navegue hasta el directorio `PA_x_x_xx.war/jsp/html/`.
3. Abra `present.jsp` en la JSP o en el editor de Java.
4. Examine el siguiente código y observe los elementos clave:

```
<%@ 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>
```

Este bloque de código contiene seis elementos clave:

1. La primera línea indica al navegador que la salida es HTML:

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

2. El siguiente conjunto de código especifica dos bibliotecas de códigos JSP utilizadas en esta página:

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

El `uri` apunta a la ubicación del directorio donde está ubicado el archivo descriptor de biblioteca de códigos. Los valores de `prefix`, definidos como `blox` y `portletAPI`, indican al servidor que haga lo siguiente:

- Busque códigos en esta página que empiecen por `blox` y, a continuación, procese el contenido utilizando la biblioteca de códigos Blox, tal y como está definida en el archivo descriptor de la biblioteca.
 - Busque códigos en esta página que empiecen por `portletAPI` y, a continuación, procese el contenido utilizando la biblioteca de códigos Portlet, tal y como está definida en el archivo descriptor de la biblioteca.
3. A continuación, se añade un código de inicialización del portlet:

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

Este código proporciona acceso a los objetos `PortletRequest`, `PortletResponse` y `PortletConfig`. Con `PortletResponse`, puede invocar el método `encodeNamespace()` para asegurarse de que el nombre del Blox no entre en conflicto con otros objetos en otros portlets que se ejecuten en la misma página.

4. El siguiente código codifica el espacio de nombres para el Blox que se va a añadir a la página:

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

Esto le permite crear posteriormente un Blox y asignarle este nombre exclusivo.

5. El siguiente bloque de código añade el código de cabecera del Blox necesario para la interpretación de Blox y para la comunicación servidor-cliente:

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

DB2 Alphablox utiliza este código para añadir automáticamente el código HTML, JavaScript y CSS necesario a la sección de la cabecera de la página. Cuando el servidor interpreta una página, este código da como resultado la inclusión de enlaces CSS en temas HTML definidos y un metacódigo para evitar la colocación en antememoria. Este código se debe añadir a cada página JSP que contenga componentes Blox, o de lo contrario los componentes no se interpretarán correctamente.

6. Añada un PresentBlox utilizando códigos proporcionados en la biblioteca de códigos Blox:

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

- Este código añade un PresentBlox con un id de presentBlox y un bloxName de xx_x_x_xxx_presentBlox, que es el resultado de la codificación de espacios de nombres.
- Este Blox tiene una anchura de 800 píxeles y una altura de 400 píxeles (el valor por omisión).
- Se incluye un DataBlox anidado, con el atributo de código dataSourceName establecido en canned.

Es necesario el atributo del código id del PresentBlox. Especifica el nombre de script de Java que se debe utilizar en la página JSP. El atributo bloxName especifica el nombre del objeto en el servidor. Un bloxName codificado garantiza que la instancia es exclusiva en el servidor.

Nota: Si no se especifica ninguna fuente de datos, normalmente se verá el mensaje **No hay datos disponibles** en la cuadrícula. La fuente de datos herméticos queda predefinida durante la instalación e incluye una pequeña cantidad de datos de ejemplo. No necesita la instalación y configuración de una base de datos externa real, y se puede utilizar para la resolución de problemas y para el aprendizaje. Puesto que no es necesario especificar una consulta, no hay ningún atributo query en el código.

Esta página no tiene un código <html> extremo porque esta página JSP se visualiza en una página del portal con otros portlets. No se necesita ningún código <html> o <body> adicional.

Ahora que ha examinado la estructura JSP y ha aprendido el código esencial que se debe incluir en las páginas JSP de portlet, en la siguiente tarea creará una nueva página JSP con un Blox distinto y especificará algunos atributos comunes de Blox para que la página se adecue mejor a la página del portal.

Creación de su propia página JSP de portlet con componentes Blox

En esta tarea, creará una nueva página JSP con un GridBlox y establecerá algunas de sus propiedades. El propósito es que se familiarice con la creación general de código Blox al mismo tiempo que crea un GridBlox que se adecue bien a esta página típica de portal.

Un GridBlox tiene un tamaño por omisión de 400x400 píxeles. También viene con una barra de menús y con una barra de herramientas. Establecerá algunas de las propiedades de GridBlox habitualmente utilizadas, de forma que la cuadrícula sólo ocupe 100 píxeles de altura, teniendo tanto su barra de menús como su barra de herramientas desactivadas. Esto se realiza estableciendo las siguientes propiedades de GridBlox:

- `height`: establecida en 100 píxeles
- `menubarVisible`: establecida en "false"
- `toolbarVisible`: establecida en "false"

Asegúrese de que ha cumplido los requisitos especificados en el Capítulo 2, "Guía de aprendizaje: Creación de su primer portlet con componentes Blox", en la página 11 y de que ha instalado la aplicación del portlet de ejemplo tal y como se describe en "Instalación de los portlets de ejemplo" en la página 11.

Siga los pasos siguientes:

1. En la ventana del navegador donde se visualiza el PresentBlox, pulse el botón de edición del portlet (el botón con el icono de un lápiz). Se le presentará una lista desplegable de selección.

2. En la lista desplegable, seleccione "Grid Blox" y pulse **Aceptar**.

Cuando se renueve la página, deberá ver un GridBlox de 400x400. Este GridBlox, por omisión, tiene su barra de menús y su barra de herramientas activado. Cambiará su tamaño a 400x100, teniendo tanto la barra de menús como la barra de herramientas desactivadas.

3. Navegue hasta el directorio `PA_x_x_xx.war/jsp/html/` en la aplicación Portlets de ejemplo de DB2 Alphablox instalada anteriormente.

4. Abra `grid.jsp` en el editor de JSP. Esta página tiene prácticamente el mismo aspecto que `present.jsp`, excepto que:

- el código `<blox:present>` ahora indica `<blox:grid>` y el valor de `id` es distinto:

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

- el valor de `bloxName` es distinto:

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

5. Establezca la altura de este GridBlox en 100 píxeles añadiendo el atributo `height` y estableciendo su valor en 100:

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

6. Desactive la barra de menús y la barra de herramientas del área superior estableciendo los atributos `menubarVisible` y `toolbarVisible` en `false`:

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

Asegúrese de especificar correctamente los nombres de los atributos, incluyendo las mayúsculas y minúsculas ("V" en mayúsculas para ambos nombres de atributo). Además, asegúrese de que se añadan los atributos antes de la llave de cierre (">").

7. Cambie el espacio de nombres a myFirstGrid:

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

Al cambiar el espacio de nombres para este GridBlox, se asegura que los cambios que acaba de realizar se reflejarán cuando cargue este JSP en el portal. Debido a que ha cargado anteriormente esta página, ya hay una instancia de este GridBlox en ejecución en el servidor para esta sesión. A menos que cambie el espacio de nombres, los cambios que acaba de realizar no se reflejarán incluso si renueva la página. Cambiar el espacio de nombres es una forma rápida de comprobar los cambios en un entorno de desarrollo. Como alternativa, también puede abrir una nueva ventana del navegador, creando así un nuevo objeto en el servidor para la nueva sesión.

8. Guarde el archivo.

Ahora está preparado para probar este archivo en WebSphere Portal.

Para probar los cambios que acaba de realizar:

1. Vuelva a la página del portal.
2. Pulse el botón Actualizar del navegador para volver a cargar la página.

Debe ver un GridBlox de 400x100 sin ninguna barra de menús ni barra de herramientas.

Nota: Esta lista desplegable de selección de JSP no es una funcionalidad incorporada de WebSphere Portal, sino que ha sido creado por este portlet de ejemplo. Seleccione el archivo edit.jsp y los archivos fuente de Java en el directorio WEB-INF/src/.

Creación de un proyecto de portlet para utilizar componentes Blox

Al crear el proyecto de portlet en la herramienta de desarrollo, asegúrese de que la correlación del servlet de AlphabloxServer y las referencias a la biblioteca de códigos necesarias para los componentes de Blox se hayan añadido al archivo web.xml del proyecto, y de que los archivos .tld de la bibliotecas de códigos de DB2 Alphablox se hayan copiado en el proyecto.

1. Modifique el archivo web.xml del proyecto para que incluya las línea siguientes:

- Para la definición del servlet y la correlación del servlet:

```
<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>
```
- Para las referencias de la biblioteca de códigos:

```
<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>bloxlogic.tld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxlogic.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxreport.tld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxreport.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxportlet.tld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxportlet.tld</taglib-location>
</taglib>
<taglib>
<taglib-uri>bloxui.tld</taglib-uri>
<taglib-location>/WEB-INF/tlds/bloxui.tld</taglib-location>
</taglib>

```

2. Copie los archivos .tld para las bibliotecas de códigos de DB2 Alphablox en el directorio WEB-INF/tlds/ del proyecto. Estos archivos se encuentran en <dir_db2alphablox>/bin/

donde <dir_db2alphablox> es el directorio de instalación de DB2 Alphablox.

Configuración de un proyecto de portlet utilizando Rational Application Developer

WebSphere Portal le recomienda que utilice Rational Application Developer para el desarrollo de portlets. En particular, se requiere Rational Application Developer Versión 6.0 para WebSphere Portal Versión 5.1. Rational Application Developer proporciona un asistente que le guía por la configuración y la creación de un proyecto de portlet. En base a la selección realizada, configura la estructura y el archivo descriptor de despliegue adecuados, y crea automáticamente las clases de Java necesarias para el controlador y las páginas JSP para la vista del portlet. Conforme avance por el asistente New Portlet Project (Nuevo proyecto de portlet), asegúrese de que lo siguiente esté configurado correctamente:

- Seleccione "WebSphere Portal v5.1 stub" como servidor de destino. Esto se especifica en la primera pantalla del asistente, pulsando el botón **Show Advanced (Mostrar avanzadas)**>>.
- Añada la correlación de servlet para DB2 Alphablox en el archivo web.xml creado en WebContent/WEB-INF/. Esto se realiza especificando la siguiente definición de servlet y correlacionando el código en el archivo web.xml del proyecto:

```

<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>

```

- Añada las bibliotecas de códigos Alphablox al web.xml del proyecto. Siga los siguientes pasos:
 1. Con el archivo web.xml abierto, pulse la pestaña **Variables**.
 2. Desplácese hasta la sección Tag Libraries References (Referencias a las bibliotecas de códigos) en el área inferior.
 3. Pulse **Add (Añadir)**.
 4. En **URL** y **Location (Ubicación)**, especifique los valores según la siguiente tabla:

URL	Ubicación
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

Para cada par de URL y Ubicación, necesitará pulsar en **Finish (Finalizar)** para añadir la referencia a la biblioteca de códigos y, a continuación, pulse **Add (Añadir)** para añadir el siguiente par.

Nota: Para obtener detalles sobre las distintas bibliotecas de códigos y los usos de las mismas, consulte Utilización de JavaServer Pages y Blox.

Por último, no olvide copiar los archivos .tld para las bibliotecas de códigos de DB2 Alphablox en el directorio WEB-INF/tlds/ del proyecto. Estos archivos se encuentran en

<db2alphablox_dir>/bin/

donde <dir_db2alphablox> es el directorio de instalación de DB2 Alphablox.

Siguientes pasos

Ha aprendido cómo crear un portlet con componentes de Blox y la estructura esencial del código. Para visualizar datos de las bases de datos, necesita hacer lo siguiente:

1. Definir una nueva fuente de datos en DB2 Alphablox que apunte a la fuente de datos.
2. Modificar el atributo dataSourceName de DataBlox para que apunte a la fuente de datos definida.
3. Añadir un atributo query con la serie de consulta en DataBlox.

Para obtener detalles acerca de cómo crear fuentes de datos, consulte la publicación *Administrator's Guide*. Para saber cómo crear consultas adecuadas, consulte la sección Retrieving Data de la publicación *Developer's Guide* y la sección DataBlox de la publicación *Developer's Reference*.

Una vez que sus propios datos se visualicen en las vistas de Blox, también puede empezar a explorar las muchas propiedades que se pueden establecer utilizando los códigos Blox.

Consejos sobre el desarrollo de portlets

Cuando utilice la información proporcionada en la documentación de DB2 Alphablox y continúe con el desarrollo de portlets, habrá algunos temas, conceptos y directrices generales específicos del portlet que será necesario que tenga en cuenta. La lista siguiente recopila estas directrices generales sobre el desarrollo y apunta a secciones específicas de la documentación que son de su interés.

- Codifique siempre el nombre del Blox utilizando el espacio de nombres del portlet. Un espacio de nombres garantiza un nombre de Blox exclusivo para la sesión J2EE actual.

- Establezca siempre la anchura y la altura de un Blox utilizando píxeles. Fuera del entorno de portal, puede establecer la anchura y la altura en un valor de porcentaje como por ejemplo el "50%" o el "100%". Los valores porcentuales no funcionan en el entorno de portal porque ya coexisten varios portlets en la página.
- No utilice URL relativos para invocar recursos dentro de un portlet. Los URL se deben codificar utilizando el método `encodeURL()` de la clase `PortletResponse`. En los lugares donde la documentación muestra la utilización de URL relativos, debe codificar siempre el URL.
- Para garantizar que el portlet de Blox utiliza un tema similar al tema de portal, utilice Portal Theme Utility. Este programa de utilidad está disponible en dos fuentes:
 - La página inicial de DB2 Alphablox.
En la pestaña **Administración**, pulse el enlace **General**. El programa de utilidad se lista en la sección Portal. Consulte la publicación *Administrator's Guide* y la ayuda en línea para obtener más detalles.
 - El archivo `AlphabloxAdminPortlets.war`.
Es una versión de portlet de Portal Theme Utility. Puede instalar este portlet para ejecutarlo mediante el portal sin tener que iniciar sesión por separado en la página inicial de DB2 Alphablox.

Este programa de utilidad combinará los estilos del entorno de portal y los estilos de DB2 Alphablox en uno, de forma que Blox se visualizará en colores y fonts similares que otros portlets en la página.

- Compruebe el tema Blox Portlet Tag Library de la publicación *Developer's Guide* para ver temas que es necesario tener en cuenta durante la fase de planificación.
- DB2 Alphablox dispone de un potente Modelo UI Blox. El Modelo UI incluye un objeto `ClientLink` que le permite cargar un URL especificado cuando se pulsa un componente de Blox en la página. Para el desarrollo de portlets, utilice la biblioteca de códigos de portlet de Blox para crear el `ClientLink`. Los códigos manejarán la formación del URL y el manejo dinámicamente, de forma que los enlaces caducarán cuando se renueve la página. Consulte el tema, *Planning for portlet development*, en la publicación *Developer's Guide* para obtener más detalles.
- Para los URI de portlets y de acción, utilice la biblioteca de códigos de portlet de Blox para crear el enlace de portlet o el enlace de acción. Podrá entonces utilizar la API del portlet del portal para procesar el enlace de acción o el enlace de portlet. Consulte el tema *Blox Portlet Tag Library* en la publicación *Developer's Guide*.

Capítulo 3. Guía de aprendizaje: Creación de aplicaciones utilizando herramientas de Rational Developer

Puede utilizar herramientas de Rational Developer con DB2 Alphablox Toolkit para desarrollar con mayor rapidez aplicaciones utilizando componentes de DB2 Alphablox y las API de Java de DB2 Alphablox.

Las herramientas de Rational Developer (Rational Application Developer o Rational Web Developer) se pueden utilizar con DB2 Alphablox Toolkit, un conjunto de plugins basados en Eclipse, para desarrollar y probar aplicaciones creadas utilizando tecnología DB2 Alphablox. Esta guía de aprendizaje le guiará en la configuración del entorno de desarrollo integrado de Rational para habilitar la utilización de la terminación de método y código, así como mejoras personalizadas añadidas en DB2 Alphablox Toolkit.

Para empezar a configurar la herramienta Rational Developer con DB2 Alphablox Toolkit para desarrollar y probar aplicaciones de DB2 Alphablox.

Preparación del entorno de desarrollo

Antes de poder instalar y comenzar a utilizar DB2 Alphablox Toolkit, prepare el entorno de desarrollo asegurándose de que tiene todo el software necesario.

Para preparar el entorno de desarrollo:

1. Instale una herramienta de Rational Developer (Rational Application Developer o Rational Web Developer) en la estación de trabajo.
2. Instale las actualizaciones necesarias para asegurarse de que tiene como mínimo Rational Application Developer o Rational Web Developer versión 6.0.0.1.
3. Si piensa utilizar entornos de prueba integrados de WebSphere 5.1, asegúrese de haber actualizado la versión de Rational Developer para incluir esta instalación opcional.

Nota: Las instalaciones por omisión de las herramientas de Rational Developer no incluyen los entornos de prueba integrados de WebSphere 5.1.

Ahora ya está preparado para realizar las tareas en esta guía de aprendizaje.

Instalación de DB2 Alphablox Toolkit

La utilización de DB2 Alphablox Toolkit junto con la herramienta Rational Developer le facilitará empezar a desarrollar aplicaciones basadas en la Web con contenido de DB2 Alphablox.

DB2 Alphablox Toolkit requiere la utilización de Rational Application Developer o bien de Rational Web Developer versión 6.0.0.1.

La instalación de DB2 Alphablox Toolkit en una herramienta Rational Developer ahorra tener que realizar múltiples pasos manuales para habilitar la utilización de terminación de contenido para métodos Java y códigos personalizados JSP. También se pueden utilizar asistentes personalizados y hojas con las soluciones

para guiarle en la creación de aplicaciones con contenido de DB2 Alphablox y en la definición de instancias de servidor WebSphere.

Para instalar DB2 Alphablox Toolkit en Rational Application Developer o Rational Web Developer:

1. Inserte el disco de instalación de DB2 Alphablox en la unidad de CD de la estación de trabajo.
2. En el directorio plugin, localice el subdirectorio llamado UpdateSite.
3. Copie el directorio UpdateSite en una ubicación adecuada del disco duro. Por ejemplo, copie el directorio en C:\DB2Alphablox\UpdateSite.
4. Inicie la herramienta Rational Developer.
5. En la barra de menús, seleccione **Help (Ayuda) > Software Updates (Actualizaciones de software) > Find and Install (Buscar e instalar)**.
6. En la ventana Install (Instalar) que se abre, seleccione la opción **Search for new features to install (Buscar nuevas características para instalar)** y luego pulse **Next (Siguiente)**.
7. En la ventana Update sites to visit (Actualizar sitios a visitar), pulse el botón **New Local Site (Nuevo sitio local)** y navegue hasta la ubicación del directorio UpdateSite.
8. Pulse **Next (Siguiente)**, seleccione la característica **DB2 Alphablox Toolkit** y pulse **Next (Siguiente)** de nuevo.
9. En las siguientes ventanas, acepte el acuerdo de licencia y seleccione la ubicación donde se instalará la característica DB2 Alphablox Toolkit.
10. Una vez haya finalizado la instalación, reinicie la herramienta Rational Developer.

Después de reiniciar la herramienta Rational Developer, las características de DB2 Alphablox Toolkit estarán disponibles.

Instalación de DB2 Alphablox en entornos de prueba integrados de WebSphere

En esta tarea, instalará DB2 Alphablox en los entornos de prueba integrados de WebSphere en una herramienta Rational Developer.

Requisitos previos: el entorno de prueba integrado de WebSphere soportado se debe instalar en la herramienta Rational Developer. Tiene que acceder al disco de instalación de DB2 Alphablox. Revise y realice los pasos necesarios previos a la instalación descritos en la sección Tareas previas a la instalación de la publicación *DB2 Alphablox Installation Guide*.

DB2 Alphablox se puede instalar en servidores de aplicaciones autónomos de WebSphere o en los entornos de prueba integrados de WebSphere disponibles dentro de la herramienta Rational Developer. El resumen de los pasos descritos a continuación explica las diferencias en la instalación que se deben realizar al instalar DB2 Alphablox en los tiempos de ejecución de WebSphere disponibles dentro de la herramienta Rational Developer.

Para instalar DB2 Alphablox en entornos de prueba integrados de WebSphere:

Instale DB2 Alphablox siguiendo los pasos de la sección sobre instalación del manual *DB2 Alphablox Installation Guide*, con las siguientes excepciones:

1. En la ventana Configurar WebSphere del programa de instalación, en el campo **Directorio raíz de WebSphere**, especifique la ubicación del entorno de ejecución de WebSphere que desea utilizar como su servidor del entorno de prueba integrado.

Por ejemplo, para instalar DB2 Alphablox en el entorno de prueba integrado de WebSphere 6 en la instalación por omisión, seleccione la vía de acceso al directorio base_v6. Para una instalación típica de Rational Application Developer, la vía de acceso sería parecida a la siguiente:

C:\Archivos de programa\IBM\Rational\SDP\6.0\runtimes\base_v6

2. En la ventana Valores de WebSphere, especifique un nombre y una contraseña para el administrador de WebSphere. Estos valores de entrada no se utilizan en el entorno de prueba integrado de WebSphere, sino que los necesita el programa de instalación de DB2 Alphablox.

Importante: No siga los pasos posteriores a la instalación descritos en la publicación *DB2 Alphablox Installation Guide*.

Después de instalar DB2 Alphablox en el entorno de ejecución de WebSphere, puede crear instancias del servidor de WebSphere para probar aplicaciones de DB2 Alphablox dentro de la herramienta Rational Developer.

Creación de una instancia de servidor WebSphere

Para ejecutar aplicaciones o archivos JSP con contenido de DB2 Alphablox, debe crear una instancia de servidor WebSphere que pueda acceder a los servicios requeridos de DB2 Alphablox y a las clases de Java.

Requisitos previos: configure la herramienta Rational Developer. Instale DB2 Alphablox Toolkit. Instale DB2 Alphablox en los entornos de prueba integrados de WebSphere.

Para crear una instancia del servidor WebSphere para probar aplicaciones y archivos JSP con contenido de DB2 Alphablox:

1. Abra la herramienta Rational Developer.
2. Pulse la pestaña **Servers (Servidores)** (si la pestaña no está visible, seleccione **Window (Ventana) > Show View (Mostrar vista) > Servers (Servidores)**).
3. Pulse con el botón derecho del ratón dentro de la ventana y seleccione **New (Nuevo) > Server (Servidor)**. Se abre la ventana New Server (Nuevo servidor).
4. Entre `sistprallocal` como nombre de sistema principal y seleccione el tipo de servidor. Pulse **Next (Siguiente)**.
5. Entre el número de puerto del servidor y pulse **Next (Siguiente)**.
6. Añada los proyectos disponibles que desee ejecutar en esta instancia de servidor a la lista de proyectos configurados. Pulse **Finalizar (Finish)**. La nueva instancia de servidor se abre en la vista **Servers (Servidores)**.

Si ha creado una instancia del servidor WebSphere 5.1, tiene que realizar el siguiente paso, "Creación de variables de sustitución de servidor WebSphere 5.1". Para instancias de servidor WebSphere 6, continúe con "Adición de un huésped al grupo Administradores de DB2 Alphablox".

Creación de variables de sustitución de servidor WebSphere 5.1

En esta tarea, modificará variables de sustitución de servidor para instancias de servidor WebSphere 5.1 para ejecutar correctamente DB2 Alphablox dentro de entornos de prueba integrados de WebSphere.

1. Abra la vista **Servers (Servidores)** y efectúe una doble pulsación con el ratón sobre la instancia de servidor creada con WebSphere 5.1 que desea modificar.
2. Pulse la pestaña **Configuration (Configuración)** de la instancia de servidor.
3. Seleccione **Enable administration console (Habilitar consola de administración)** y deseleccione **Enable universal test client (Habilitar cliente de prueba universal)**.
4. Pulse la pestaña **Variables** de la instancia de servidor. Se abre la ventana **Substitution Variables (Variables de sustitución)**.
5. Añada dos nuevas variables utilizando el botón **Add (Añadir)** (situado junto a la lista **Node Settings (Configuración de nodos)**).
 - a. Añada una variable llamada `WS_EAR_AlphabloxPlatform` y establezca su valor en `$(APP_INSTALL_ROOT)/localhost/AlphabloxPlatform.ear`.
 - b. Añada una segunda variable llamada `WS_EAR_AlphabloxStudio` y establezca su valor en `$(APP_INSTALL_ROOT)/localhost/ApplicationStudio.ear`.
6. Guarde los cambios.

Para instancias de servidor de WebSphere 5.1, necesitará seguir los pasos descritos en "Configuración de instancias de servidor de WebSphere 5.1" para terminar de modificar la instancia del servidor.

Configuración de instancias de servidor de WebSphere 5.1

Para instancias de servidor WebSphere 5.1, debe configurar las instancias de servidor para que ejecuten aplicaciones y archivos de DB2 Alphablox.

Requisitos previos: cree las variables de sustitución de servidor WebSphere 5.1 necesarias.

Para configurar instancias de servidor WebSphere 5.1:

1. Abra la vista **Servers (Servidores)** en la herramienta Rational Developer e inicie la instancia del servidor que desee configurar.
2. Pulse con el botón derecho del ratón sobre la instancia de servidor y seleccione **Run Administrative Console (Ejecutar consola de administración)**.
3. No entre ningún valor en el campo **ID**. Pulse **OK (Aceptar)**.
4. Pulse el botón **OK (Aceptar)** que hay bajo la tabla de configuración.
5. Abra la vista de navegación **Enterprise Applications (Aplicaciones empresariales)** y pulse sobre el nombre de aplicación **ApplicationStudio**.
6. Establezca para **Application Binaries** el valor `$(WS_EAR_ApplicationStudio)`.
7. Seleccione **Use Metadata from Binaries (Utilizar metadatos a partir de binarios)**.
8. Pulse el botón **OK (Aceptar)** que hay bajo la tabla de configuración.
9. Pulse el botón **Save (Guardar)** en la consola de administración de WebSphere.
10. En la ventana **Save to Master Configuration (Guardar en configuración maestra)**, pulse **Save (Guardar)**.

11. Inicie las aplicaciones AlphabloxPlatform y ApplicationStudio en la consola de administración de WebSphere bajo **Applications (Aplicaciones) > Enterprise Applications (Aplicaciones empresariales)**.

La instancia de servidor ya está configurada. A continuación, debe seguir el paso "Adición de un huésped al grupo Administradores de DB2 Alphablox".

Adición del usuario huésped al grupo Administradores de DB2 Alphablox

La adición del usuario huésped al grupo Administradores de DB2 Alphablox le permite acceder a las páginas de administración de DB2 Alphablox mientras utiliza la herramienta Rational Developer.

Requisitos previos: cree la instancia de servidor WebSphere. Configure la instancia de servidor WebSphere.

Para añadir el usuario huésped al grupo Administradores de DB2 Alphablox:

Importante: No le otorgue derechos de administrador al usuario huésped en los servidores WebSphere de producción.

1. Inicie una sesión en la consola de DB2 Alphablox utilizando el siguiente mandato telnet: `telnet sistema_principal_local númeroPuerto .` donde *númeroPuerto* es el puerto especificado durante la instalación de DB2 Alphablox.
2. En el indicador de la consola telnet, escriba el siguiente mandato de la consola DB2 Alphablox: `set Administrators guest` y pulse Intro.
3. En la consola telnet, escriba `save` y luego pulse Intro.
4. Cierre la sesión telnet.

La instancia de servidor WebSphere ya está lista para su utilización.

Creación de aplicaciones de DB2 Alphablox

Al crear nuevas aplicaciones en Rational Developer, es necesario añadir el contenido DB2 Alphablox necesario para que se ejecuten correctamente dichas aplicaciones.

Requisitos previos: instale DB2 Alphablox Toolkit en la herramienta Rational Developer.

1. En la barra de menús de la herramienta Rational Developer, seleccione **File (Archivo) > New Project (Nuevo proyecto)**. Se abrirá el asistente **New Project (Nuevo proyecto)**.
2. Amplíe la opción **Web**, seleccione **Dynamic Web Project (Proyecto Web dinámico)** y luego pulse **Next (Siguiente)**.
3. Escriba un nombre para el proyecto y pulse el botón **Show Advanced (Mostrar opciones avanzadas)**. Aparecerán opciones adicionales.
4. Seleccione la versión de servlet y el servidor de destino adecuados.
5. Pulse **Next (Siguiente)**. Se abre la ventana **Features options (Opciones de características)**.
6. Seleccione la opción **DB2 Alphablox Content (Contenido DB2 Alphablox)** y luego pulse **Finish (Finalizar)**.

El nuevo proyecto de aplicación Web está ahora habilitado para DB2 Alphablox. Las bibliotecas de códigos de DB2 Alphablox y la API Java de Blox están disponibles y el archivo del descriptor de aplicación (web.xml) se ha modificado a fin de incluir la información de DB2 Alphablox necesaria. Ahora ya es posible añadir archivos JSP con contenido de DB2 Alphablox al proyecto.

Creación de archivos JSP con contenido DB2 Alphablox

En esta tarea, se crean nuevos archivos JSP con acceso a bibliotecas de códigos Blox utilizando la herramienta Rational Developer con DB2 Alphablox Toolkit.

Requisitos previos: instale DB2 Alphablox Toolkit en la herramienta Rational Developer.

1. En la vista **Project Explorer (Explorador de proyectos)** de Rational Developer, seleccione **File (Archivo) > New (Nuevo) > JSP File (Archivo JSP)**.
2. Escriba un nombre de archivo en el campo **File Name (Nombre de archivo)**.
3. Pulse **Configure advanced options (Configurar opciones avanzadas)** y luego **Next (Siguiendo)**.
4. Añada la biblioteca de códigos de DB2 Alphablox que va a utilizar en el archivo JSP.
 - a. Pulse el botón **Add (Añadir)** para abrir la ventana Add Tag Libraries (Añadir bibliotecas de códigos).
 - b. Seleccione las bibliotecas de códigos de DB2 Alphablox que piense utilizar en el archivo JSP.
 - c. Pulse **Next (Siguiendo)**.
5. Seleccione **ISO 10646/Unicode(UTF-8)** en la lista **Encoding (Codificación)**. La codificación UTF-8 es necesaria para que se ejecuten correctamente las aplicaciones de DB2 Alphablox.
6. Pulse **Finish (Finalizar)**. El nuevo archivo JSP aparecerá en el listado de proyectos.
7. Efectúe una doble pulsación sobre el nombre de archivo para abrir el archivo en la ventana del editor JSP.
8. Coloque el cursor dentro del código HTML <head>, pero detrás de las directrices de códigos JSP de DB2 Alphablox que ha añadido anteriormente.
9. Entre el código de cabecera de Blox escribiendo lo siguiente en una nueva línea: <blox:header/> El código de cabecera de Blox es necesario para añadir archivos DB2 Alphablox JavaScript y CSS requeridos al ejecutar el archivo JSP.

El nuevo archivo JSP está ahora habilitado para acceder a las bibliotecas de códigos de DB2 Alphablox que ha seleccionado. Con la característica Content Assist (Asistente de contenido) de la herramienta Rational Developer, puede insertar códigos y atributos de códigos Blox para las bibliotecas de códigos seleccionadas.

Acceso a fuentes de datos de DB2 OLAP Server y Essbase

Para acceder a fuentes de datos de DB2 OLAP Server o Essbase en la herramienta Rational Developer al utilizar instancias de servidor de WebSphere 5.1, debe crear un archivo de proceso por lotes de inicio que cargue bibliotecas de cliente necesarias.

Requisitos previos: instale DB2 Alphablox Toolkit. Instale DB2 Alphablox en los entornos de prueba integrados de WebSphere 5.1. Configure las instancias de servidor de WebSphere 5.1. Instale IBM DB2 OLAP Server o bien Hyperion Essbase en la máquina de desarrollo.

Debido a limitaciones en los entornos de prueba integrados de WebSphere 5.1 en la herramienta Rational Developer, debe crear un archivo de proceso por lotes de inicio para la herramienta Rational Developer para acceder a fuentes de datos de DB2 OLAP Server o de Hyperion Essbase. Cuando se inicia la herramienta Rational Developer utilizando el archivo de proceso por lotes, se añaden las bibliotecas de cliente de Essbase necesarias a la vía de acceso de la biblioteca de Java.

Para crear un archivo de proceso por lotes para habilitar el acceso a DB2 OLAP Server o bien a Essbase con instancias de servidor de WebSphere 5.1:

1. Utilizando un editor de texto, cree un nuevo documento de texto.
2. Añada una línea de código para invocar el archivo de DB2 Alphablox `aasetup.bat`, ubicado en el directorio de instalación de DB2 Alphablox. Por ejemplo, el siguiente código ejecutará el archivo `aasetup.bat` ubicado en la instalación especificada de DB2 Alphablox:

```
call C:\alphablox\analytics\bin\aasetup.bat
```
3. Añada una segunda línea para invocar el archivo `rationalsdp.exe` de RAD, que iniciará RAD. Por ejemplo, el siguiente código ejecutará la herramienta Rational Developer:

```
call  
C:\Archivos de Programa\IBM\Rational\SDP\6.0\rationalsdp.exe
```
4. Guarde este archivo como `startRAD.bat` en el escritorio de la estación de trabajo (o en otra ubicación que le resulte conveniente).

Cuando se efectúa una doble pulsación sobre el archivo `startRAD.bat`, se ejecuta el archivo de DB2 Alphablox `aasetup.bat` y se establecen las vías de acceso y las variables de entorno necesarias; luego se inicia Rational Application Developer. La versión de las bibliotecas de cliente de Essbase utilizadas con DB2 Alphablox debe coincidir con la versión de DB2 OLAP Server (o bien de Hyperion Essbase) que esté utilizando como fuente de datos. Para modificar las bibliotecas del cliente Essbase utilizadas con DB2 Alphablox, ejecute el programa de utilidad de bibliotecas de cliente de DB2 OLAP Server / Essbase (`ChangeEssbase.bat`), que se encuentra en el directorio `db2_alphablox\analytics\bin`, donde `db2_alphablox` es el directorio raíz de la instalación de DB2 Alphablox. Este programa de utilidad modifica uno de los archivos de proceso por lotes que se ejecuta al iniciar RAD utilizando el archivo de proceso por lotes `startRAD` que se ha creado.

A continuación aparece un ejemplo completo de archivo `startRAD.bat` descrito anteriormente:

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

Siempre que desee acceder a fuentes de datos de DB2 OLAP Server o bien de Essbase, debe utilizar el archivo `startRAD.bat` para iniciar la herramienta Rational Developer.

Capítulo 4. Guía de aprendizaje: Creación de un cubo de DB2 Alphablox utilizando DB2 Cube Views

La guía de aprendizaje le guía por el proceso de creación de un cubo de DB2 Alphablox creado utilizando una base de datos de ejemplo de DB2 Cube Views.

Las tareas en la guía de aprendizaje no proporcionan detalles sobre la creación de cubos personalizados de DB2 Alphablox. El objetivo es, por el contrario, mostrarle cómo crear rápidamente un cubo de DB2 Alphablox que se pueda utilizar en la exploración de las capacidades de DB2 Alphablox Cube Server. La fuente de datos resultante también se puede utilizar para probar y crear aplicaciones de DB2 Alphablox.

Requisitos previos:

- Instale DB2 Alphablox. Consulte la publicación *Installation Guide* para obtener detalles sobre la instalación.
- Obtenga derechos de acceso a una implementación soportada de DB2 Cubes Views que tenga la base de datos de ejemplo CVSAMPLE de DB2 Cube Views instalada. Consulte la publicación *Installation Guide* para obtener detalles sobre las versiones soportadas de DB2 Cube Views.

En la guía de aprendizaje, aprenderá cómo crear un cubo de DB2 Alphablox basado en la base de datos de ejemplo CVSAMPLE de DB2 Cube Views. Durante el aprendizaje, aprenderá las siguientes tareas:

Definición de una fuente de datos relacionales DB2

En esta tarea, definirá una definición de fuente de datos en DB2 Alphablox para una base de datos DB2.

Requisitos previos: los controladores JDBC necesarios de DB2 deben resultar accesibles para DB2 Alphablox.

Un cubo de DB2 Alphablox requiere que la fuente de datos relacional subyacente esté predefinida como una fuente de datos de DB2 Alphablox. Los cubos de DB2 Alphablox se generan utilizando los metadatos y datos disponibles en bases de datos relacionales.

Para definir una fuente de datos de DB2 Alphablox para una base de datos DB2:

1. Inicie una sesión en las páginas de administración de DB2 Alphablox como usuario administrador (o como un usuario con derechos de administración).
2. Pulse la pestaña **Administración**.
3. Pulse el enlace **Fuentes de datos**.
4. Pulse el botón **Crear**.
5. En el menú **Adaptador**, seleccione el controlador JDBC de IBM DB2 adecuado para el servidor de bases de datos.
Seleccione **IBM DB2 JDBC Type 4 Driver** o **IBM DB2 UDB on iSeries Driver**.
6. En el campo **Nombre de fuente de datos**, escriba CVSAMPLE como el nombre a utilizar para la fuente de datos.

- Entre los valores correspondientes en **Nombre de servidor**, **Número de puerto** y **Nombre de base de datos** (este último debe ser CVSAMPLE).

Nota: Si necesita ayuda para determinar los valores correctos para estos campos, póngase en contacto con el administrador de bases de datos.

- Entre un **Nombre de usuario por omisión** y una **Contraseña por omisión**. El nombre de usuario y la contraseña deben ser válidos en la base de datos relacional. El nombre de usuario y la contraseña por omisión siempre se utilizan cuando un cubo de DB2 Alphablox accede a una base de datos relacionales. El usuario de la base de datos utilizado deberá tener acceso de lectura a la base de datos.

Nota: El valor de **Utilizar nombre de usuario y contraseña de DB2 Alphablox** se ignora cuando la fuente de datos relacionales especificada se utiliza para poblar un cubo de DB2 Alphablox. Es posible utilizar ACL (listas de control de acceso) para restringir el acceso a cubos de DB2 Alphablox. Para obtener más información acerca de las ACL, consulte la publicación *Administrator's Guide*.

- Los valores **Número máximo de filas** y **Número máximo de columnas** se pasan por alto cuando la fuente de datos se utiliza para poblar un cubo de DB2 Alphablox. Todavía puede entrar valores que serán utilizados cuando otras aplicaciones utilicen la fuente de datos.
- Establezca para **Rastreo JDBC habilitado** el valor **No**, a no ser que desee escribir información de anotación cronológica de JDBC en el archivo de anotaciones cronológicas de DB2 Alphablox. Habilite el rastreo JDBC sólo si está experimentando problemas y necesita depurar las causas.
- Pulse el botón **Guardar** para guardar la definición de fuente de datos.

Ahora ya ha definido la definición de fuente de datos de DB2 Alphablox para CVSAMPLE. Ya puede crear una definición de cubo de DB2 Alphablox para acceder a los metadatos de cubos de Cube Views de esta fuente de datos DB2.

Definición de una fuente de datos de Alphablox Cube Server Adaptor

En esta tarea, definirá una definición de fuente de datos de DB2 Alphablox que utiliza Alphablox Cube Server Adapter.

Requisitos previos: cree la definición de fuente de datos de DB2 Alphablox para la base de datos CVSAMPLE de DB2.

Para definir una fuente de datos de DB2 Alphablox Cube Server Adapter:

- Inicie una sesión en las páginas de administración de DB2 Alphablox como usuario administrador (o como un usuario con derechos de administración).
- Pulse la pestaña **Administración**.
- Pulse el enlace **Fuentes de datos**.
- Pulse el botón **Crear**.
- En el menú **Adaptador**, seleccione la opción **Alphablox Cube Server Adapter**.
- En el campo **Nombre de fuente de datos**, entre DB2AlphabloxCubes como el nombre a utilizar para la fuente de datos.
- Pulse el botón **Guardar** para guardar la definición de fuente de datos.

Ha definido una fuente de datos de DB2 Alphablox que se puede utilizar para acceder a cubos de DB2 Alphablox. Lo siguiente que debe hacer es definir un cubo de DB2 Alphablox que pueda acceder.

Definición de un cubo de DB2 Alphablox

En esta tarea, definirá un cubo de DB2 Alphablox basado en los metadatos procedentes de un cubo CVSAMPLE de DB2 Cube Views.

Requisitos previos: defina la fuente de datos relacional de DB2. Defina una fuente de datos de DB2 Alphablox Cube Server Adapter.

Para definir las propiedades generales de un cubo DB2 Alphablox:

1. Inicie una sesión en las páginas de administración de DB2 Alphablox como usuario administrador (o como un usuario con derechos de administración).
2. Pulse la pestaña **Administración**.
3. Pulse el enlace **Cubos**.
4. Pulse el botón **Crear**. Se abre la ventana Administración de cubos.
5. Defina el nuevo cubo:
 - a. En el campo **Nombre de cubo de DB2 Alphablox**, escriba **CVSales**.
 - b. Seleccione la opción **Habilitado**, situada junto al campo **Nombre de cubo de DB2 Alphablox**. Al seleccionar esta opción, el cubo se iniciará automáticamente cuando se reinicie el servidor.
 - c. En el menú **Fuente de datos relacionales**, seleccione **MyDB2**, la fuente de datos relacionados que ha creado para esta guía de aprendizaje.
 - d. Deje la opción **Función de seguridad** sin seleccionar. Puede utilizar esta opción para limitar los usuarios con acceso a un cubo en particular.
6. Habilite los valores de DB2 Cube Views y especifique los metadatos que se van a utilizar:
 - a. Seleccione la opción **Habilitar configuración de DB2 Cube Views**.
 - b. En el menú **Modelo de cubo**, seleccione **CVSAMPLE.Sales**.
 - c. En el menú **Cubo**, seleccione el cubo **Ventas generales**.
 - d. Seleccione el botón de selección **Utilizar nombres comerciales** para especificar los nombres. Al seleccionar esta opción se utilizarán los nombres de miembros, normalmente más significativos y legibles.
 - e. Pulse el botón **Importar definición de cubo**. Al utilizar esta opción podrá importar una definición de cubo y repoblar las medidas y dimensiones en el cubo DB2 Alphablox. La definición de cubo importada refleja el cubo de DB2 Cube Views tanto como sea posible para DB2 Alphablox, basándose en el soporte de metadatos de DB2 Cube Views. Cuando adquiera más experiencia en el trabajo con cubos, podrá modificar medidas y dimensiones para ajustarse a sus necesidades.
 - f. Pulse el botón **Mostrar registro de importación** para ver un registro que contiene mensajes de información y de depuración relacionados con la operación de importación. En esta guía de aprendizaje, ese paso se incluye únicamente para ayudarle a familiarizarse con esta característica.
 - g. Seleccione la opción **Importar cubo al iniciar, volver a crear y editar**. Esta opción hará que el cubo de DB2 Alphablox utilice la definición de DB2 Cube Views más reciente cada vez que el inicie el cubo DB2 Alphablox, se vuelva a crear o abra para su edición. Cuando esté más familiarizado con DB2 Alphablox y DB2 Cube Views, podrá importar la definición de cubo y personalizarla.

7. Pulse el botón **Guardar** para guardar la definición de cubo de DB2 Alphablox.

Ahora ya ha creado una definición de cubo de DB2 Alphablox. Ya puede iniciar el nuevo cubo CVSales.

Inicio de un cubo DB2 Alphablox

En esta tarea, se inicia el cubo CVSales utilizando las páginas de administración de DB2 Alphablox.

Requisitos previos: defina la fuente de datos relacional de DB2. Defina una fuente de datos de DB2 Alphablox Cube Server Adapter. Defina el cubo de DB2 Alphablox.

Para iniciar el cubo CVSales:

1. Inicie una sesión en las páginas de administración de DB2 Alphablox como usuario administrador (o como un usuario con derechos de administración).
2. Pulse la pestaña **Administración**.
3. En la sección **Gestión de tiempo de ejecución**, pulse el enlace **Cubos**.
4. En la lista **Cubos de DB2 Alphablox**, seleccione el cubo de DB2 Alphablox que desea iniciar.
5. Pulse el botón **Iniciar**. Cuando se haya iniciado el cubo, el campo de estado mostrará **En ejecución**.

Ahora ya tiene un cubo de DB2 Alphablox de ejemplo en ejecución. Ya puede empezar a crear aplicaciones utilizando el cubo de DB2 Alphablox que acaba de crear. Puede realizar una comprobación rápida del nuevo cubo utilizando la aplicación Query Builder (Creador de consultas), seleccionando el cubo como la fuente de datos y ejecutando consultas MDX en el cubo.

Avisos

Esta información se ha desarrollado para productos y servicios que se ofrecen en los EE.UU.

Es posible que IBM no ofrezca en todos los países los productos, servicios o características que se describen en este documento. Consulte al representante local de IBM para obtener información sobre los productos y servicios que están disponibles actualmente en su área. Cualquier referencia a un producto, programa o servicio IBM, no pretende afirmar ni implicar que sólo se pueda utilizar ese producto, programa o servicio de IBM. En su lugar, se puede utilizar cualquier producto, programa o servicio funcionalmente equivalente que no infrinja ningún derecho de la propiedad intelectual de IBM. Sin embargo, es responsabilidad del usuario evaluar y verificar el funcionamiento de cualquier producto, programa o servicio no IBM.

IBM puede tener patentes o solicitudes de patente pendientes en referencia a los temas descritos en este documento. La entrega de este documento no otorga ninguna licencia sobre estas patentes. Puede enviar consultas sobre licencias, por escrito, a:

IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 EE.UU.

Para realizar consultas sobre licencias con relación a información de doble byte (DBCS), póngase en contacto con el Departamento de la propiedad intelectual de IBM en su país o envíe sus consultas, por escrito, a:

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

El siguiente párrafo no es aplicable al Reino Unido ni a ningún otro país donde tales disposiciones sean incompatibles con la legislación local: INTERNATIONAL BUSINESS MACHINES CORPORATION PROPORCIONA ESTA PUBLICACIÓN "TAL CUAL" SIN GARANTÍA DE NINGUNA CLASE, NI EXPLÍCITA NI IMPLÍCITA, INCLUYENDO, PERO SIN LIMITARSE A ELLAS, LAS GARANTÍAS IMPLÍCITAS DE NO VULNERACIÓN DE DERECHOS, COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN DETERMINADO. Algunos estados no permiten la exclusión de garantías expresas o implícitas en determinadas transacciones, por lo que es posible que esta declaración no sea aplicable en su caso.

Esta publicación puede contener imprecisiones técnicas o errores tipográficos. Periódicamente se efectúan cambios en la información aquí contenida; dichos cambios se incorporarán a las nuevas ediciones de la publicación. IBM puede efectuar, en cualquier momento y sin previo aviso, mejoras y cambios en los productos y programas descritos en esta publicación.

Las referencias hechas en esta publicación a sitios Web que no son de IBM se proporcionan sólo para la comodidad del usuario y no constituyen un aval de esos sitios Web. La información contenida en esos sitios Web no forma parte de la información del presente producto IBM y el usuario es responsable de la utilización de dichos sitios Web.

IBM puede utilizar o distribuir cualquier información que se le facilite de la manera que considere adecuada, sin contraer por ello ninguna obligación con el remitente.

Los licenciarios de este programa que deseen obtener información sobre él con el fin de habilitar: (i) el intercambio de información entre programas creados de forma independiente y otros programas (incluido éste) y (ii) el uso mutuo de la información intercambiada, deben ponerse en contacto con:

IBM Corporation, J46A/G4, 555 Bailey Avenue, San Jose, CA 95141-1003 EE.UU.

Dicha información puede estar disponible, sujeta a los términos y condiciones apropiados, incluido en algunos casos el pago de una tarifa.

El programa bajo licencia descrito en este documento y todo el material bajo licencia asociado a él, los proporciona IBM según los términos del Acuerdo de cliente de IBM, el Acuerdo internacional de programas bajo licencia de IBM o cualquier acuerdo equivalente entre el usuario e IBM.

Los datos de rendimiento contenidos en este documento se han obtenido en un entorno controlado. Por lo tanto, los resultados obtenidos en otros entornos operativos pueden variar significativamente. Algunas mediciones pueden haberse realizado en sistemas experimentales y no es seguro que estas mediciones sean las mismas en los sistemas disponibles en general. Además, algunas mediciones pueden haberse calculado mediante extrapolación. Los resultados reales pueden variar. Los usuarios del presente documento deben verificar los datos aplicables para su entorno específico.

La información referente a productos que no son de IBM se ha obtenido de los proveedores de esos productos, de sus anuncios publicados o de otras fuentes disponibles públicamente. IBM no ha probado esos productos y no puede confirmar la exactitud del rendimiento, la compatibilidad ni ninguna otra afirmación referente a productos que no son de IBM. Las preguntas sobre las prestaciones de productos que no son de IBM deben dirigirse a los proveedores de esos productos.

Todas las declaraciones de intenciones de IBM están sujetas a cambio o cancelación sin previo aviso, y sólo representan objetivos.

Esta información contiene ejemplos de datos e informes utilizados en operaciones comerciales diarias. Para ilustrarlos de la forma más completa posible, los ejemplos incluyen nombres de personas, empresas, marcas y productos. Todos estos nombres son ficticios y cualquier similitud con nombres y direcciones utilizadas por una empresa real es una simple coincidencia.

Esta información contiene programas de aplicaciones de ejemplo en lenguaje fuente, que ilustran técnicas de programación en varias plataformas operativas. Puede copiar, modificar y distribuir estos programas de ejemplo como desee, sin pago alguno a IBM, con la intención de desarrollar, utilizar, comercializar o distribuir programas de aplicaciones de acuerdo con la interfaz de programación de aplicaciones correspondiente a la plataforma operativa para la que están escritos los programas de ejemplo. Estos ejemplos no se han probado exhaustivamente bajo todas las condiciones. Por lo tanto, IBM no puede asegurar ni implicar la fiabilidad, utilidad o función de estos programas. Puede copiar, modificar y distribuir estos programas de ejemplo como desee, sin pago alguno a IBM, con la

intención de desarrollar, utilizar, comercializar o distribuir programas de aplicaciones de acuerdo con las interfaces de programación de aplicaciones de IBM.

Marcas registradas

Los términos siguientes son marcas registradas de International Business Machines Corporation en los Estados Unidos y/o en otros países:

DB2
IBM

DB2 OLAP Server
WebSphere

DB2 Universal Database

Alphablox y Blox son marcas registradas de Alphablox Corporation en los Estados Unidos y/o en otros países.

Java y todas las marcas basadas en Java son marcas registradas de Sun Microsystems, Inc. en los Estados Unidos y/o en otros países.

Linux es una marca registrada de Linus Torvalds en los Estados Unidos y/o en otros países.

Otros nombres de empresas, productos o servicios pueden ser marcas registradas o marcas de servicio de terceros.

Índice

C

consultas
 generación mediante Query
 Builder 4

G

guías de aprendizaje
 desarrollo de aplicaciones 1

P

portlet
 ejemplo 11
 estructura de JSP 12

Q

Query Builder
 utilización 4
Query Builder, DHTML
 utilización 4



Número de Programa: 5724-L14

GC28-3125-01

