

IBM DB2 Alphablox



시작하기

버전 8.4

IBM DB2 Alphablox



시작하기

버전 8.4

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 37 페이지의 『주의사항』의 정보를 읽으십시오.

제 2 판(2006년 3월)

이 개정판은 새 개정판에서 별도로 명시하지 않는 한 Linux, UNIX 및 Windows용 IBM DB2 Alphablox(제품 번호 5724-L14)의 버전 8, 릴리스 4와 모든 후속 릴리스 및 수정판에 적용됩니다.

IBM으로 의견을 보내면, IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

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

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

목차

제 1 장 자습서: 첫 번째 응용프로그램 빌드	1	개발 환경 준비	23
응용프로그램 정의	1	DB2 Alphablox 툴킷 설치	23
데이터 액세스	2	WebSphere 통합 테스트 환경에 DB2 Alphablox	
응용프로그램 홈 페이지 작성	3	설치	24
디폴트 홈 페이지 설정	4	WebSphere 서버 인스턴스 작성	25
첫 번째 분석 뷰 작성	4	WebSphere 5.1 서버 대체 변수 작성	26
첫 번째 분석 뷰 페이지의 구조	8	WebSphere 5.1 서버 인스턴스 구성	26
두 번째 분석 뷰 작성	10	DB2 Alphablox 관리자 그룹에 게스트 사용자 추가	27
요약	11	DB2 Alphablox 응용프로그램 작성	28
제 2 장 자습서: Blox 구성요소로 첫 번째 포틀렛		DB2 Alphablox 콘텐츠로 JSP 파일 작성	28
빌드	13	DB2 OLAP Server 및 Essbase 데이터 소스 액세스	
샘플 포틀렛 설치	13	스	29
샘플 포틀렛 실행	14	제 4 장 자습서: DB2 Cube Views를 사용한 DB2	
Blox 구성요소가 있는 포틀렛 JSP 페이지 구조		Alphablox 큐브 빌드	31
점검	15	DB2 관계 데이터 소스 정의	31
Blox 구성요소로 사용자 고유의 포틀렛 JSP 페이지		Alphablox 큐브 서버 어댑터 데이터 소스 정의	33
작성	17	DB2 Alphablox 큐브 정의	33
Blox 구성요소를 작성하기 위한 포틀렛 프로젝트 작		DB2 Alphablox 큐브 시작	34
성	19	주의사항	37
Rational Application Developer를 사용한 포틀		상표	39
렛 프로젝트 구성	20	색인	41
다음 단계	21		
포틀렛 개발 추가 정보	21		
제 3 장 자습서: Rational Developer 도구를 사용			
하여 응용프로그램 빌드	23		

제 1 장 자습서: 첫 번째 응용프로그램 빌드

이 자습서는 Blox 구성요소 개념에 대한 중요한 정보를 알리고, DB2® Alphablox 응용프로그램을 신속하게 빌드하기 위해 수행할 수 있는 단계를 제공합니다.

DB2 Alphablox 응용프로그램을 빌드하려면, JSP(JavaServer Pages) 기술을 사용해야 하지만 이 자습서에서 JSP에 대한 지식은 필요하지 않습니다. 이 자습서는 분석 응용프로그램 빌드를 시작하기 위해 알아야 하는 JSP 요소를 설명합니다. 많은 우수한 책들을 사용하여 JSP 기술에 대해 자세히 알 수 있습니다.

전제조건

이 자습서의 단계에서는 WebSphere® Application Server에서 실행 중인 DB2 Alphablox를 사용하는 것으로 간주합니다. 응용프로그램 서버(AS)로서 BEA WebLogic 또는 Apache Tomcat을 사용하는 경우, 특정 차이점을 사용자 서버에 맞추기 위해 다음 단계 중 일부를 수정해야 합니다.

또한 이 자습서를 시작하려면 DB2 Alphablox 큐브(qcc_2003) 및 기본 관계형 데이터 소스(qcc2003-rdb)를 설치해야 합니다. 이 자습서에 필요한 DB2 Alphablox 큐브 및 관계형 데이터베이스 구성에 대한 샘플 데이터베이스 파일 및 지시사항(readme.txt 파일)을 DB2 Alphablox 설치 CD의 `sampledata/qcc/acs` 디렉토리에서 찾을 수 있습니다.

응용프로그램 정의

J2EE 개발 접근 방법을 사용하여 응용프로그램을 작성하려면, 응용프로그램 디스크립터 파일(web.xml)을 포함하는 WEB-INF 디렉토리로 디렉토리 구조를 작성해야 합니다. DB2 Alphablox에서 이 구조를 작성하는 가장 단순한 방법은 DB2 Alphablox 관리 페이지 내의 응용프로그램 페이지에서 새 응용프로그램을 작성하는 것입니다.

응용프로그램 및 폴더를 작성하려면 다음을 수행하십시오.

1. *Administrator's Guide*의 Application Definitions 섹션에서 설명한 단계를 수행하여 DB2 Alphablox에 MyApp라는 새 응용프로그램을 작성하십시오.
 - 이름 필드에 MyApp를 입력하십시오.
 - 응용프로그램 페이지의 목록에 표시되는 응용프로그램의 레이블을 정의하는 표시 이름 필드에 My App(공백 있음)를 입력하십시오.
2. 페이지 상단 왼쪽에 있는 **응용프로그램** 탭을 누르십시오. 사용 가능한 응용프로그램 목록이 열립니다.

3. 응용프로그램 목록에서 새로 작성한 응용프로그램(My App)을 누르십시오. 아직 파일을 작성하지 않았으므로, 파일 디렉토리가 비어 있습니다.
4. 응용프로그램 서버의 새 응용프로그램 폴더를 탐색하십시오.(WebSphere의 경우, 폴더는 WebSphere installedApps 디렉토리에 위치합니다.) 응용프로그램에는 web.xml 파일이 있는 WEB-INF 디렉토리와 tlds 디렉토리가 포함된다는 점에 유의하십시오. web.xml 파일은 응용프로그램 정보를 정의합니다. tlds 디렉토리에는 Blox 태그 라이브러리 디스크립터 파일(blox.tld)이 포함되며, 이 파일은 이 자습서에서 분석 뷰를 작성하는 데 사용할 Blox 태그를 정의합니다. 디렉토리에는 bloxform.tld, bloxlogic.tld, bloxreport.tld 및 bloxui.tld 등 다른 Blox 태그 라이브러리(이 자습서에서는 사용하지 않음)를 위한 다른 TLD 파일도 있습니다.

작성한 DB2 Alphablox 응용프로그램 프레임워크에 분석 뷰를 추가할 수 있습니다.

데이터 액세스

이 자습서의 경우 자습서 시작 전에 작성한 DB2 Alphablox 큐브 데이터를 사용합니다. 관계형 데이터베이스 및 Alphablox Cube Server를 실행 중이어야 합니다. 사용 중인 DB2 Alphablox 큐브를 또한 실행 중이어야 합니다.

필수 데이터 소스가 사용 가능하며 작동하는지 확인하려면 다음을 수행하십시오.

1. 브라우저에서 DB2 Alphablox 관리 페이지를 여십시오.
2. 관리 탭을 누르십시오.
3. 데이터 소스를 누르십시오.
4. 데이터 소스 페이지에서 페이지 왼쪽에 있는 메뉴에 qcc2003-rdbdata 소스가 표시되어야 합니다. 그렇지 않을 경우 DB2 Alphablox 설치 CD에서 sampledata/qcc/acs 디렉토리의 readme.txt 파일에 있는 단계를 검토하십시오.
5. 사용 가능한 데이터 소스 목록에서 qcc2003-rdb를 누른 다음 선택된 데이터 소스 테스트 단추를 누르십시오. 데이터베이스를 실행 중이고 데이터 소스 정의가 적절하게 구성된 경우, 성공 메시지가 표시됩니다. 오류 메시지가 표시되면, 데이터 소스 정의(데이터 소스 선택 후 편집 단추를 눌러) 또는 데이터베이스에 대한 액세스 권한이 있는 사용자로 데이터베이스가 적절히 구성되었는지 여부를 점검해야 합니다.
6. 데이터 소스의 목록에서 qcc2003-acs 데이터 소스를 선택한 다음 선택된 데이터 소스 테스트 단추를 누르십시오. Alphablox Cube Server 데이터 소스가 올바르게 구성된 경우, 성공 메시지가 표시됩니다. 오류 메시지가 표시될 경우, 데이터 소스 정의를 검토하십시오.

데이터 소스를 성공적으로 테스트하였으면 한 쌍의 단순 DB2 Alphablox 분석 뷰를 빌드하는 방법을 학습할 준비가 되었습니다.

응용프로그램 홈 페이지 작성

이 태스크의 경우, 기본 웹 기술을 사용하여 단순한 홈 페이지를 작성할 수 있습니다.

이 자습서에서는 매우 간단한 웹 응용프로그램 구조만을 작성합니다. 홈 페이지에서 해당 자습서를 작성하는 두 개의 분석 뷰를 액세스하기 위한 두 개의 링크를 작성해야 합니다.

단순 홈 페이지를 작성하려면 다음을 수행하십시오.

1. 텍스트 편집기 또는 선호하는 개발 도구를 사용하여 `index.html`로 이름을 지정하십시오. 이 자습서에서는 이 파일을 매우 단순하게 유지할 수 있습니다.
2. 파일을 편집하여 응용프로그램 제목(예: “내 DB2 Alphablox 응용프로그램”)과 작성할 두 개의 분석 뷰에 대한 두 링크를 지정하십시오.

다음 코드를 `index.html` 파일에 복사한 다음 붙여넣거나 입력하십시오.

```
<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. 이 파일을 응용프로그램을 정의할 때 작성한 `MyApp` 디렉토리에 저장하십시오.
4. 웹 브라우저에서 DB2 Alphablox 홈 페이지를 여십시오. 디폴트로, 브라우저는 응용프로그램 탭을 표시합니다.
5. **My App** 링크를 눌러 파일이 디렉토리 목록에 포함되어 있는지 확인하십시오. **index.html** 링크를 눌러 홈 페이지를 보십시오. DB2 Alphablox를 IBM WebSphere Application Server 또는 BEA WebLogic 서버에 설치한 경우, 해당 단계를 수행하기 전에 디렉토리 목록을 사용 가능하게 해야 합니다. Apache Tomcat에 DB2 Alphablox를 설치할 경우, 추가적인 단계가 필요하지 않습니다. HTTP 403 오류가 표시될 경우(디렉토리 목록이 디폴트로 허용되지 않음), 디렉토리 목록을 허용하도록 서버를 수정하거나 또는 `myapp.html` 파일 이름을 URL 끝에 추가할 수 있습니다(예: `http://localhost:9080/MyApp/myapp.html`). 완전한 링크 주소 사용 시 새 페이지가 표시됩니다.

이제 다음 번에 작성할 두 개의 분석 뷰에 대한 링크가 있는 홈 페이지가 표시됩니다.

디폴트 홈 페이지 설정

이전 단계에서는 MyApp 응용프로그램 디렉토리에 있는 파일 이름 링크를 누르거나 URL에서 index.html 홈 페이지 파일을 지정하여 직접 홈 페이지를 액세스했습니다. 이 단계에서는 응용프로그램의 디폴트 홈 페이지를 지정합니다.

디폴트 응용프로그램 홈 페이지를 설정하려면 다음을 수행하십시오.

1. 브라우저에서 DB2 Alphablox 홈 페이지를 여십시오.
2. 관리 탭을 누르십시오.
3. **응용프로그램** 링크를 누르십시오.
4. 응용프로그램 목록에서 **MyApp**를 선택하고 목록 아래에 있는 편집 단추를 누르십시오.
5. MyApp 응용프로그램 편집 페이지에서 홈 URL 필드에 index.html을 입력하십시오.
6. 저장 단추를 눌러 변경사항을 저장하십시오.
7. 맨 위 왼쪽 폴더 탭인 기본 응용프로그램 탭을 눌러 응용프로그램 페이지로 리턴하십시오.
8. **My App** 응용프로그램 이름을 누르십시오. 정의된 홈페이지가 응용프로그램에서 직접 열립니다.

이제 홈 페이지가 정의되었으며, 서버는 사용자가 해당 URL에 홈 페이지를 지정하는데 실패하더라도 응용프로그램을 어디에서 시작해야 하는지 알게 되며 응용프로그램에서 해당 페이지를 열도록 지시합니다. 그러므로 사용자가 `http://yourServerName/MyApp/`를 브라우저 주소 막대에 입력하면, 응용프로그램은 정의된 index.html 홈 페이지를 자동으로 엽니다.

다음 단계에서는 DB2 Alphablox 쿼리 빌더 응용프로그램을 사용하여 생성된 간단한 Blox 태그를 사용하여 DB2 Alphablox 분석 뷰 작성을 시작합니다.

첫 번째 분석 뷰 작성

이 태스크에서는 PresentBlox 구성요소에 결합된 격자 및 차트를 표시하는 DB2 Alphablox 분석 뷰를 작성합니다.

전제조건: DB2 Alphablox 및 해당 구성요소에 대해 알려면 *Developer's Guide*의 도입 섹션을 읽으십시오. DB2 Alphablox 기능 및 Blox 사용자 정의 태그 라이브러리를 JSP 기술을 사용하는 분석 응용프로그램 빌드에 사용할 수 있는 방법에 대한 기본적인 이해를 제공합니다.

작성할 첫 번째 분석 뷰의 경우, DB2 Alphablox 쿼리 빌더 응용프로그램을 사용하여 데이터를 표시하는 데 사용되는 가장 인기있는 Blox 구성요소인, PresentBlox 구성요소에 대해 정의된 JSP 사용자 정의 태그를 생성합니다. PresentBlox 구성요소는 격자 및 차트 뷰, 데이터 레이아웃 패널, 도구 모음 및 기타 중첩 구성요소를 포함할 수 있습니다. 격자 및 차트 뷰는 동일한 데이터의 동기화된 대체 뷰를 표시합니다.

쿼리 빌더를 사용하여 첫 번째 분석 뷰를 작성하려면 다음을 수행하십시오.

1. 브라우저에서 DB2 Alphablox 관리 페이지를 여십시오. 디폴트로, 응용프로그램 탭이 표시됩니다.
2. DB2 Alphablox 쿼리 빌더를 눌러서 응용프로그램을 여십시오.
3. 연결 설정 단추를 누르십시오. 데이터베이스 연결 창이 표시됩니다.
4. qcc2003-acs 데이터 소스를 선택한 다음 연결 단추를 누르십시오. 창이 닫히며 쿼리 빌더 페이지의 데이터베이스 상태 섹션에 "연결됨" 메시지가 표시됩니다.

주: 선택한 데이터 소스를 데이터 소스 정의에서 자동으로 검색하므로 스키마, 카탈로그, 사용자 이름 및 암호 필드에 값을 입력할 필요가 없습니다. 또한 디폴트 쿼리 실행 선택란을 선택하지 마십시오.

5. 페이지의 상태 섹션 아래 쿼리 섹션이 표시됩니다. 다음과 같은 특징을 주의하십시오.
 - 이 자습서의 경우 DB2 Alphablox 큐브(qcc2003-acs)를 사용합니다. 쿼리 섹션 헤더 바로 밑에 사용 가능한(현재 실행 중인) 큐브 목록이 표시됩니다. qcc2003-acs 큐브가 나열되어야 합니다.
 - 해당 데이터 소스와 연관된 디폴트 쿼리 문자열이 텍스트 창에 표시됩니다. 데이터 소스가 DB2 Alphablox 큐브이므로, 알파벳 순으로 나열된 첫 번째 큐브가 쿼리 명령문에 추가됩니다(예: "select from [*datasourceName*]").

주: DB2 Alphablox Cube Server의 연결 풀(pool) 기능이 사용 가능할 경우, 디폴트 쿼리 가져오기 단추를 사용하여 DB2 Alphablox 쿼리에 대한 기본 쿼리를 가져올 수 없습니다.

- 페이지 맨 아래 있는 PresentBlox에 데이터가 표시되면, 표준 사용자 인터페이스를 사용하여 축 스왑, 드릴 업 또는 드릴 다운, 축 간 차원 이동 등을 수행하십시오.
 - 쿼리 결과를 보려면 쿼리 실행 단추를 누르십시오. 결과 세트가 페이지 맨 아래 있는 PresentBlox에 표시됩니다.
6. 디폴트 쿼리의 대괄호 안에 qcc2003-acs를 입력한 다음(예를 들어, select from [qcc2003-acs]) 쿼리 실행 단추를 누르십시오.

PresentBlox 섹션에 쿼리 기반 디폴트 뷰가 표시됩니다.

7. PresentBlox 사용자 인터페이스를 사용하여 새로운 뷰를 작성하십시오.

예를 들어, 제품 차원을 왼쪽에 있는 데이터 레이아웃 패널의 행 축으로 끌 수 있습니다. 그런 다음 시간(달력) 차원을 열 축으로 끌 수 있습니다. 그런 다음 Measures 차원을 페이지 축으로 끌어서 Measures 메뉴를 표시 영역 위의 페이지 패널(페이지 필터를 표시함)에 표시하고자 할 수도 있습니다. 두 개의 기타 차원을 선택하여 세 개의 페이지 필터가 표시 영역 위에 표시될 수 있도록 페이지 축에 추가하십시오. 처음으로 DB2 Alphablox 사용자 인터페이스를 사용할 경우, 인터페이스 사용 방법에 대해 자세히 알려면 PresentBlox 도구 모음에서 도움말을 선택하십시오.

8. 새 뷰 작성을 완료한 후 쿼리 섹션에 표시되는 쿼리 명령문을 주의하십시오. 쿼리가 동적으로 생성되지 않을 경우(자동 쿼리 갱신 옵션을 선택 해제했을 때), 현재 쿼리 가져오기 단추를 눌러서 현재 쿼리를 가져올 수 있습니다. 텍스트 상자는 현재 데이터 뷰를 개발하는 데 필요한 쿼리 명령문을 표시합니다. DB2 Alphablox 응용프로그램 개발에 사용할 쿼리 명령문을 쿼리 필드에서 복사한 후 붙여넣을 수 있습니다.
9. **Blox** 태그 생성 단추를 누르십시오. PresentBlox 레이아웃 및 결과 세트를 재생하기 위한 태그(및 중첩 태그)를 표시하는 대화 상자가 열립니다.
10. 텍스트 편집기에서 새 파일을 열어서 다음 텍스트를 파일에 추가한 후 파일을 PresentBloxView.jsp로 저장하십시오. 다음 주제는 해당 페이지에 있는 여러 가지 중요한 코드의 요소를 설명합니다.

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

11. 저장된 파일(PresentBloxView.jsp)을 서버의 MyApp 디렉토리로 복사한 후 해당 페이지의 링크를 눌러서 MyApp 홈 페이지의 새 분석 뷰인, 단순 PresentBlox 뷰를 테스트하십시오. 또다른 방법으로 URL을 브라우저 주소에 직접 입력할 수 있습니다(예를 들어, <http://localhost:9080/MyApp/PresentBloxView.jsp>). 표시되는 뷰는 DB2 Alphablox 쿼리 빌더에서 처리한 PresentBlox와 유사합니다.

단순 PresentBlox 뷰 페이지의 완전한 버전에 대한 코드 페이지가 아래 표시됩니다.

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

```

위의 코드 예제에는 판독성을 위해 쿼리 명령문에 행 구분이 있습니다. 이 페이지를 적절하게 실행하려면 제거할 속성 값(쿼리 및 위의 selectableSlicerDimensions 속성과 같은)에 행 구분이 필요합니다. DB2 Alphablox 쿼리 빌더가 생성한 코드에서 포함된 많은 태그는 반드시 필요하지 않습니다. 그럼에도 불구하고 여기에 있는 모든 태그를 관찰하면 Blox 태그가 상위 및 중첩 태그로 작동하는 방법에 대한 식견이 생깁니다. 또한 쿼리 빌더의 PresentBlox에서 수행한 처리의 결과로 추가된 태그 속성의 용도를 볼 수 있습니다. Blox 태그 사용, 태그 속성 및 DB2 Alphablox 응용프로그램 개발 방법의 세부사항에 대해 자세히 알려면, DB2 Alphablox 정보 센터를 참조할 수 있습니다.

첫 번째 분석 뷰 페이지의 구조

이 주제에는 이전 주제에서 작성한 첫 번째 분석 뷰(PresentBloxView.jsp)의 구조가 요약되어 있습니다.

다음은 이전 주제에서 작성한 페이지의 일부 강조표시에 대한 간략한 개요입니다. 코드를 완전하게 이해하려면 DB2 Alphablox 정보 센터에서 관심있는 주제를 읽어야 합니다.

PresentBloxView.jsp 페이지는 다음과 같은 행으로 시작합니다.

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

이 행은 Blox 태그 라이브러리를 사용하려고 한다는 것을 서버에 알리는 JSP taglib 지시문입니다. uri는 태그 라이브러리 디스크립터 파일에 대한 포인터입니다. blox로 정의된 prefix 값은 이 페이지에서 blox로 시작하는 모든 태그를 찾은 후 Blox 태그 라이브러리 디스크립터 파일에서 정의한 대로 처리하도록 서버에 알려줍니다.

taglib 지시문 바로 밑에 있는 PresentBlox 구성요소는 다음 태그와 해당 태그 속성으로 지정됩니다.

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

```
[qcc_2003].[Locations],[qcc_2003].[Chocolate Type]"
useAliases="true"/>
<blox:toolbar/>
<blox:dataLayout/>
<bloxui:calculationEditor />
</blox:present>
```

<blox:present> 태그는 PresentBlox가 queryBuilder4_present의 id 속성 값으로 여기에 표시되도록 지정합니다. id 속성을 사용하여 스크립팅용으로 이 특정 Blox를 식별할 수 있습니다. <blox:present> 태그에는 false로 설정된 visible 속성이 포함됩니다. false로 설정될 경우, 컴파일러에서 <blox:display> 태그를 발견할 때까지 페이지에서 PresentBlox를 렌더링하지 않습니다.

태그를 사용하여 정의된 Blox마다 사용 가능한 속성은 많으나 디폴트값과 다른 속성값을 지정해야 하는 경우를 제외하고는 태그에 이러한 속성을 포함시킬 필요가 없습니다. 이 예제에서 height 값 500, width 값 100%. 높이와 너비는 픽셀이나 백분율로 정의할 수 있습니다.

dataSourceName 속성을 qc2003-acs로 정의하는 중첩된 DataBlox가 포함됩니다. 데이터 소스가 지정되지 않으면, 페이지를 표시할 때 "사용 가능한 데이터가 없음" 메시지가 표시됩니다. 쿼리 속성은 DB2 Alphablox Query Builder에서 생성한 MDX 쿼리 명령문을 포함합니다. DB2 Alphablox Cube Server에서 사용하는 MDX 쿼리 언어를 이해하면, 쿼리를 훨씬 더 짧게 만들어서 단순하게 할 수 있습니다. 그럼에도 불구하고 예제의 쿼리 명령문은 완전하며 쿼리 빌더 응용프로그램에서 작성한 뷰를 생성합니다.

페이지의 <head> 섹션에는 페이지가 렌더링되기 전에 중요한 코드를 페이지에 추가하는 데 필요한 특수 Blox 태그가 있습니다.

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

이 태그는 DB2 Alphablox에 의해 필수 HTML, JavaScript™ 및 CSS 코드를 페이지의 헤드 섹션으로 자동으로 추가하는 데 사용됩니다. 서버에서 페이지를 렌더링할 경우, 이 태그는 캐싱을 방지하기 위해 정의된 HTML 테마 및 메타 태그에 CSS 링크를 추가합니다. Blox 구성요소를 사용하는 모든 JSP 페이지에 이 태그를 입력해야 합니다. 이 태그가 포함되지 않으면, 이 페이지는 제대로 렌더링되지 않습니다.

페이지의 <body> 섹션에는 다음과 같은 행이 있습니다.

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

이 Blox 태그는 위에서 지정한 PresentBlox를 여기에 렌더링합니다. 이를 사용하여 Blox 구성요소 또는 해당 중첩 구성요소를 하나의 위치에서 변경할 수 있으며, 또한 여기에는 Blox 구성요소의 동작을 제어하는 다른 Blox 태그 및 JSP 스크립틀릿이 있을 수도 있습니다. 복합 JSP 파일에서와 DB2 Alphablox와 보다 익숙해지면, Blox 구성요

소를 웹 브라우저에 렌더링하기 전에 DataBlox 구성요소를 PresentBlox 태그와 분리하고 Blox 구성요소 및 해당 동작을 수정하는 JSP 스크립틀릿을 추가하도록 선택할 수도 있습니다.

요약하면 이 페이지에서 사용되는 가장 중요한 Blox 태그에는 <blox:header/>, <blox:present> 및 <blox:display>가 포함됩니다. 이러한 세 개의 태그와 중첩 태그는 페이지에서 Java™ 코드를 사용하지 않고도 분석 뷰를 지정합니다. 프리젠테이션 논리의 복잡도는 Blox 태그로 관리합니다. 중첩 Blox를 추가하고 속성값을 수정함으로써 비즈니스 요구사항에 맞게 사용자가 뷰를 정의할 수 있습니다.

보다 복잡한 뷰 및 응용프로그램의 경우 추가 JavaBeans 구성요소 및 Java 코드가 필요할 수도 있습니다.

두 번째 분석 뷰 작성

이 태스크에서는 해당 지습서의 첫 번째 분석 뷰의 코드를 근거로 페이지 필터가 있는 차트만을 표시하는 분석 뷰를 작성합니다.

많은 응용프로그램에서 데이터는 격자 또는 차트를 사용하여 표시됩니다. PresentBlox 구성요소는 PresentBlox 구성요소 내에서 함께 표시되는 중첩 구성요소로서 GridBlox 및 ChartBlox를 결합하지만, 표시하지 않으려면 중첩 구성요소를 사용 불가능하게 하거나 또는 숨길 수 있습니다. PresentBlox 구성요소를 사용한 페이지 필터가 있는 차트 뷰 작성은 상대적으로 간단하며, 일부 중첩된 Blox 태그에 대한 태그 속성을 수정하여 달성합니다.

이 태스크에서는 첫 번째 분석 뷰의 PresentBlox 태그를 다시 사용하여 두 번째 사용자 정의된 분석 뷰를 작성합니다.

페이지 필터가 있는 차트를 표시하는 분석 뷰를 작성하려면 다음을 수행하십시오.

1. 편집기에서 이전에 작성한 PresentBloxView.jsp 파일을 열고 CustomizedChartBlox.jsp로 저장하십시오.
2. 아래 나열된 PresentBlox 태그의 중첩된 구성요소에서 visible 속성을 각각의 태그에 추가하고 값을 false로 설정하십시오. 이 태그를 추가하면 페이지 필터 막대가 있는 차트 뷰를 남겨두며, 중첩된 구성요소는 표시하지 않습니다.

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

3. labelPlacement속성을 추가하고 값을 top으로 설정하여 PageBlox 태그 (<blox:page>)를 수정하십시오. 기본적으로 페이지 필터 레이블은 메뉴 왼쪽에 표시됩니다. 이 속성을 추가한 다음 top으로 설정하면 기본 동작이 대체되며 페이지 필터 위에 레이블이 배치됩니다.


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

4. `<blox:present>` 태그에서 ID 속성의 이름을 다음 값으로 수정하십시오. `chartview`. 오류를 방지하려면 페이지에 표시되거나 렌더링된 각각의 Blox 구성요소에 고유 ID가 필요합니다.
5. 다음과 같이 변경사항을 저장하고 파일을 MyApp 디렉토리에 추가하십시오.
6. 웹 브라우저를 열고 홈 페이지에서 링크를 누르거나 또는 URL을 사용하여 직접 액세스함으로써 새 페이지를 테스트하십시오(예: <http://localhost:9080/MyApp/CustomizedChartView.jsp>).

이제 DB2 Alphablox Query Builder 응용프로그램을 이용하여 시작에 도움을 주는 두 개의 분석 뷰를 작성했습니다. 또한 기존 태그에서 속성을 수정하거나 추가하여 Blox 및 해당 중첩 구성요소의 모양 및 동작을 제어할 수 있음을 학습했습니다.

요약

모든 지습서 태스크를 완료한 경우, DB2 Alphablox 관리 페이지, DB2 Alphablox 쿼리 빌더 응용프로그램 및 Blox 태그를 사용하여 기본 DB2 Alphablox 응용프로그램을 빌드하는 방법을 다 배웠습니다. 이미 *Developer's Reference*를 읽었다면 JSP 페이지에서 Blox를 정의하고 조작하는 데 사용 가능한 등록 정보 및 메소드가 많다는 사실을 알 것입니다. 그렇다면 다음 단계는 무엇일까요?

DB2 Alphablox 뷰로 기업 데이터를 보고 싶은 경우, 응용프로그램을 작성하고 응용프로그램에 분석 뷰를 추가하는 방법을 알고 있습니다. 즉시 무언가를 보려는 경우, MyApp 응용프로그램을 간단히 수정할 수 있습니다. qcc2003-acs 데이터 소스를 사용하지 않고 기업 데이터베이스에서 데이터를 표시하려면, 기업 데이터베이스를 가리키는 데이터 소스를 작성하고 새로 정의된 데이터 소스를 가리키도록 `dataSourceName` 속성을 수정한 다음 적절한 `query` 속성을 추가해야 합니다. DB2 Alphablox 쿼리 빌더를 사용하여 종종 사용 가능한 쿼리 명령문 및 Blox 태그를 빠르게 작성할 수 있습니다. 데이터 소스 작성에 대한 세부사항은 *Administrator's Guide*를 참조하십시오. 적절한 쿼리 작성 방법 및 DB2 Alphablox 응용프로그램 빌드에 대해 자세히 알려면 *Developer's Guide*의 Retrieving Data 및 *Developer's Reference*의 DataBlox 섹션을 참조하십시오.

제 2 장 자습서: Blox 구성요소로 첫 번째 포틀릿 빌드

이 자습서에서는 Blox 구성요소를 포틀릿에 추가하는 방법에 대해 배우게 됩니다. 다음을 수행합니다.

1. 사전 빌드된 샘플 포틀릿을 설치합니다. 이 단계를 사용하여 Blox 구성요소가 포틀릿 JSP 페이지에 추가되는 방법과 포털 페이지에 표시되는 방법을 신속하게 볼 수 있습니다.
2. GridBlox가 있는 JSP 페이지를 작성합니다.

이 자습서의 태스크에는 일반적인 포틀릿 개발에 대한 세부사항이 나와 있지는 않습니다. 이 자습서에서는 DB2 Alphablox 고유의 태스크에 초점을 맞추고 있으며, 포털 환경 및 포틀릿 개발에 대한 일반 개념에 익숙한 것으로 가정합니다.

전제조건

- DB2 Alphablox는 WebSphere Portal 버전 5.1 서버에 설치되어야 합니다. 설치 세부사항에 대해서는 설치 안내서를 참조하십시오.
- WebSphere Portal 서버를 시작해야 합니다.
- WebSphere Portal 서버에 관리 차원의 액세스를 할 수 있어야 합니다.
- WebSphere Portal에 있는 관리 기능 및 사용자 인터페이스에 익숙해야 합니다.
- Java 및 JSP에 대한 기본 지식이 있어야 합니다.
- JSP 편집기가 설치되어야 합니다.

이 자습서에서 JSP 편집기 또는 텍스트 편집기를 사용하더라도, 사용자 자신의 포틀릿 개발 작업을 위해서는, Rational® Application Developer처럼 WebSphere Portal에서 권장하는 개발 도구를 사용해야 합니다.

이 자습서에서는 DB2 Alphablox와 같이 설치된 사전 정의 데이터 소스를 사용할 것입니다. 이 데이터 소스를 사용하여 신속하게 기본 응용프로그램을 개발할 수 있습니다. 이 자습서의 사용자 정의 데이터 소스 구성에 대해서는 걱정할 필요가 없습니다.

샘플 포틀릿 설치

Blox를 포틀릿에 추가하는 방법을 배우는 가장 좋은 방법은 먼저 DB2 Alphablox에서 제공하는 샘플 포틀릿을 설치하고 이들을 포털 페이지로 로드하는 것입니다. 이렇게 하면 JSP 코드의 기본 구조를 점검하고 이들을 포털의 출력에 일치시킬 수 있습니다.

다음과 같이 DB2 Alphablox에 제공된 샘플 포틀릿을 설치하십시오.

1. 브라우저를 열고 포털에 관리자로 로그인하십시오. (URL은 `http://<yourPortalServer>:<port>/wps/portal`입니다.)
2. 관리 단추를 누르십시오.
3. 포틀렛 관리 섹션에서 웹 모듈을 누르십시오. 웹 모듈 관리 페이지가 오른쪽에 표시됩니다.
4. 설치를 누르십시오. 웹 모듈을 설치하도록 프롬프트됩니다.
5. 찾아보기 단추를 누르고 DB2 Alphablox 설치 디렉토리 아래의 `installableApps` 디렉토리를 탐색하십시오.
6. `AlphabloxSamplePortlets.war`를 선택한 후 다음을 누르십시오. DB2 Alphablox JSP 페이지 샘플 포틀렛 이름의 포틀렛이 있는 DB2 Alphablox 샘플 포틀렛 응용 프로그램이 포틀렛 응용프로그램 테이블에 표시됩니다.
7. 완료를 누르십시오.

DB2 Alphablox 샘플 응용프로그램과 포함된 포틀렛이 이제 설치되었습니다. WebSphere Portal 설치 아래에 있는 `installedApps` 디렉토리를 점검하십시오. 새로 작성된 디렉토리 이름은 DB2 Alpha로 시작하고 동적으로 생성된 `_PA_x_x_xx.ear` 패턴의 포틀렛 ID로 종료합니다.

샘플 포틀렛 실행

포털 페이지에 샘플 포틀렛을 실행하려면 다음을 수행하십시오.

1. 포털 페이지로 찾아가십시오.
2. 기존 페이지를 작성하거나 편집하십시오. 이 샘플 포틀렛을 테스트하기 위해 새 페이지를 작성하거나 기존 페이지를 눌러 페이지를 편집할 수 있습니다.
3. 포털 레이아웃 페이지에서 포틀렛 추가 단추 중 하나를 누르십시오.
4. 검색 필드에 DB2를 입력한 후, 검색을 누르십시오. **DB2 Alphablox JSP** 페이지 샘플 포틀렛 선택란이 표시됩니다.
5. 선택란을 선택하고 확인을 누르십시오.
6. 완료를 누르십시오.

포털 페이지가 새로 고쳐지면 포털 페이지에 PresentBlox가 표시됩니다. 이 PresentBlox에는 다음이 들어 있습니다.

- 위쪽의 메뉴 표시줄
- 메뉴 표시줄 아래 두 개의 도구 모음 패널
- 다른 축으로 차원을 이동시킬 수 있는 왼쪽의 데이터 레이아웃 패널
- 표 형식으로 데이터를 표시하는 격자
- 오른쪽의 3차원 막대 도표

이 present.jsp 파일은 샘플에 포함된 BloxJSPPagePortlet servlet에 지정된 대로 로드할 디폴트 페이지입니다. 이 샘플 servlet의 소스 코드는 WEB-INF/src/ 디렉토리에 있습니다.

이제 샘플 포틀릿을 설치하고 Blox가 있는 포틀릿을 포털 페이지에 추가했습니다. 다음 태스크는 JSP 페이지에서 코드 구조를 점검합니다.

Blox 구성요소가 있는 포틀릿 JSP 페이지 구조 점검

이 섹션에서는 Blox가 포함된 JSP 페이지의 코드 구조에 대해 검토합니다. 모든 JSP 페이지에는 동일한 키 요소가 있어야 합니다.

JSP 파일을 열려면 다음을 수행하십시오.

1. WebSphere Portal 설치 아래에 있는 installedApps/ 디렉토리를 탐색하여 DB2 Alpha로 시작하는 새로 작성된 응용프로그램 폴더를 찾으십시오.
2. PA_x_x_xx.war/jsp/html/ 디렉토리를 탐색하십시오.
3. JSP 또는 Java 편집기에서 present.jsp를 여십시오.
4. 다음 코드를 검사한 후 키 요소에 주목하십시오.

```
<%@ 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>
```

이 코드 블록에는 6개의 키 요소가 있습니다.

1. 첫 행에서는 출력이 HTML임을 브라우저에게 알려줍니다.

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

2. 다음 코드 세트에서는 이 페이지에서 사용된 두 개의 JSP 태그 라이브러리를 지정합니다.

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

uri는 태그 라이브러리 디스크립터 파일이 있는 디렉토리 위치에 대한 포인터입니다. blox 및 portletAPI로 정의된 prefix 값은 서버에 다음을 알립니다.

- 이 페이지에서 blox로 시작하는 태그를 찾은 후 Blox 태그 라이브러리를 사용하여 그 콘텐츠를 태그 라이브러리 디스크립터 파일에 정의된 것처럼 처리합니다.

- 이 페이지에서 portletAPI로 시작하는 태그를 찾은 후 포틀릿 태그 라이브러리를 사용하여 그 콘텐츠를 태그 라이브러리 디스크립터 파일에서 정의한 대로 처리합니다.

3. 다음으로 포틀릿 초기화 태그를 추가합니다.

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

이 태그는 PortletRequest, PortletResponse 및 PortletConfig 오브젝트에 액세스할 수 있도록 합니다. PortletResponse로, encodeNamespace() 메소드를 호출하여 Blox 이름이 같은 페이지에서 실행되는 다른 포틀릿의 오브젝트와 충돌하지 않도록 할 수 있습니다.

4. 다음 태그에서는 페이지에 추가할 Blox의 이름 스페이스를 인코딩합니다.

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

이렇게 하면 나중에 Blox를 작성하고 이 고유한 이름을 지정할 수 있습니다.

5. 다음 코드 블록에서는 Blox 렌더링 및 서버-클라이언트 통신에 필요한 Blox 헤더 태그를 추가합니다.

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

이 태그는 DB2 Alphablox에서 사용되며 필수 HTML, JavaScript 및 CSS 코드를 페이지의 헤드 섹션에 자동으로 추가합니다. 서버에서 페이지를 렌더링할 경우, 이 태그는 캐싱을 예방하기 위해 CSS 링크를 정의된 HTML 테마 및 메타 태그에 포함시킵니다. 이 태그는 Blox 구성요소가 들어 있는 JSP 페이지마다 추가되어야 하며 그렇지 않은 경우 구성요소는 적절하게 렌더링되지 않습니다.

6. Blox 태그 라이브러리에서 제공하는 태그를 사용하여 PresentBlox를 추가하십시오.

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

- 이 코드는 id가 presentBlox이고 bloxName이 xx_x_x_xxx_presentBlox인 PresentBlox를 추가하며 이것은 이름 스페이스를 인코딩한 결과입니다.
- 이 Blox의 너비는 800 픽셀이고 높이는 400 픽셀(디폴트값)입니다.
- canned로 설정된 dataSourceName 태그 속성으로 중첩된 DataBlox가 포함됩니다.

PresentBlox의 id 태그 속성이 필요합니다. 이 태그는 JSP 페이지에서 사용할 Java 스크립트 이름을 지정합니다. bloxName 속성은 서버에서의 오브젝트 이름을 지정합니다. bloxName이 인코딩되면, 인스턴스가 서버에서 고유하게 됩니다.

주: 데이터 소스가 지정되지 않으면, 일반적으로 사용 가능한 데이터가 없음 메시지가 격자에 표시됩니다. 미리 준비된 데이터 소스는 설치 도중 미리 정의되고 소량의 샘플 데이터를 포함합니다. 실제 외부 데이터베이스의 설치와 구성이 필요하지 않고 문제점 해결 및 학습에 사용될 수 있습니다. 쿼리를 지정할 필요가 없으므로 코드에서 query 속성은 없습니다.

이 JSP 페이지가 다른 포틀릿과 함께 포털 페이지에 표시되므로 이 페이지에는 가장 바깥쪽의 <html> 태그가 없습니다. 추가 <html> 또는 <body> 태그는 필요하지 않습니다.

JSP 구조를 점검하고 포틀릿 JSP 페이지에 포함할 필수 코드에 대해 학습했습니다. 다음 태스크에서는 Blox가 다른 새 JSP 페이지를 작성하고 포털 페이지에 적합한 페이지가 되도록 일부 공통 Blox 속성을 지정합니다.

Blox 구성요소로 사용자 고유의 포틀릿 JSP 페이지 작성

이 태스크에서는 GridBlox가 있는 새 JSP 페이지를 작성하고 몇몇 등록 정보를 설정할 것입니다. 여기서는 일반 포털 페이지에 적합한 GridBlox를 작성하면서 일반 Blox 태그 구문에 익숙해지도록 합니다.

GridBlox의 디폴트 크기는 400x400 픽셀입니다. 메뉴 표시줄과 도구 모음도 함께 표시됩니다. 메뉴 표시줄과 도구 모음을 표시하지 않고 격자 높이가 100 픽셀이 되도록 일반적으로 사용되는 GridBlox 등록 정보를 설정합니다. 아래와 같이 GridBlox 등록 정보를 설정하면 됩니다.

- height : 100 픽셀로 설정
- menubarVisible: 거짓으로 설정
- toolbarVisible: 거짓으로 설정

13 페이지의 제 2 장 『자습서: Blox 구성요소로 첫 번째 포틀릿 빌드』에서 지정한 요구사항을 충족하는지와 13 페이지의 『샘플 포틀릿 설치』 섹션에서 설명한 바와 같이 샘플 포틀릿 응용프로그램을 설치했는지 확인하십시오.

다음 단계를 수행하십시오.

1. PresentBlox가 표시된 브라우저 창에서 포틀릿의 편집 단추(연필 아이콘이 있는 단추)를 누르십시오. 선택 드롭다운 목록이 표시됩니다.
2. 드롭다운 목록에서 "격자 Blox"를 선택하고 확인을 누르십시오.

페이지가 새로 고쳐지면 400x400 GridBlox가 표시됩니다. 이 GridBlox에는(디폴트로) 메뉴 표시줄과 도구 모음이 표시되어 있습니다. 메뉴 표시줄과 도구 모음을 표시하지 않고 크기를 400x100으로 변경합니다.

- 이전에 설치한 DB2 Alphablox 샘플 포틀릿 응용프로그램에서 PA_x_x_xx.war/jsp/html/ 디렉토리를 탐색하십시오.
- JSP 편집기에서 grid.jsp를 여십시오. 이 페이지는 다음을 제외하면 present.jsp와 거의 유사합니다.

- <blox:present> 태그가 <blox:grid>이고 id 값이 다릅니다.

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

- bloxName 값이 다릅니다.

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

- height 속성을 추가하고 그 값을 100으로 설정하여 GridBlox의 높이를 100으로 설정합니다.

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

- menubarVisible 및 toolbarVisible 속성을 false로 설정하여 위쪽에 있는 메뉴 표시줄과 도구 모음을 표시하지 않습니다.

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

대소문자를 구분하여 속성 이름을 정확하게 입력했는지 확인하십시오(속성 이름에서 "V"는 대문자임). 또한 종료 꺾쇠(">") 앞에 속성을 추가했는지 확인하십시오.

- 이름 스페이스를 myFirstGrid로 변경하십시오.

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

GridBlox의 이름 스페이스를 변경하면 JSP를 포털에 로드할 때 방금 변경한 내용이 반영됩니다. 이전에 이 페이지를 로드했으므로 해당 세션에 대한 이 GridBlox의 인스턴스가 서버에서 이미 실행되고 있습니다. 이름 스페이스를 변경하지 않으면, 페이지를 새로 고치더라도 변경 내용이 적용되지 않습니다. 이름 스페이스를 변경하는 것이 개발 환경에서 변경 사항을 테스트하는 가장 빠른 방법입니다. 또는 새 세션에 대한 새 오브젝트를 서버에서 작성할 수 있도록 새 브라우저 창을 열 수도 있습니다.

- 파일을 저장합니다.

이제 WebSphere 포털에서 이 파일을 테스트할 준비가 되었습니다.

변경 내용을 테스트하려면 다음을 수행하십시오.

- 포털 페이지로 이동하십시오.

2. 브라우저의 새로 고침 단추를 눌러 페이지를 다시 로드하십시오.
메뉴 표시줄이나 도구 모음 없이 400x100 GridBlox가 표시되어야 합니다.

주: 이 JSP 선택 드롭다운 목록은 WebSphere 포털의 내장 기능이 아니라 이 샘플 포털에서 작성된 것입니다. WEB-INF/src/ 디렉토리에서 edit.jsp 파일 및 Java 소스 파일을 점검하십시오.

Blox 구성요소를 작성하기 위한 포틀릿 프로젝트 작성

개발 도구로 포틀릿 프로젝트를 작성할 때, Blox 구성요소로 필요한 AlphabloxServer servlet 맵핑 및 태그 라이브러리 참조가 프로젝트의 web.xml 파일에 추가되었는지 확인하고, DB2 Alphablox Tag 라이브러리에 대한 .tld 파일이 프로젝트로 복사되었는지 확인하십시오.

1. 다음 행을 포함하도록 프로젝트의 web.xml 파일을 수정하십시오.

- servlet 정의 및 servlet 맵핑의 경우,

```
<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>
```

- 태그 라이브러리 참조의 경우,

```
<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>
```

- DB2 Alphablox 태그 라이브러리에 대한 .tld 파일을 프로젝트의 WEB-INF/tlds/ 디렉토리로 복사하십시오. 파일의 위치는 다음과 같습니다.

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

여기서 <db2alphablox_dir>은 DB2 Alphablox 설치 디렉토리입니다.

Rational Application Developer를 사용한 포틀릿 프로젝트 구성

WebSphere Portal에서는 포틀릿 개발에 Rational Application Developer를 사용하도록 권장합니다. 특히 Rational Application Developer 버전 6.0은 WebSphere Portal 버전 5.1에 필요합니다. Rational Application Developer에서는 포틀릿 프로젝트의 설정 및 작성을 안내할 마법사를 제공합니다. 사용자 선택에 따라, 적합한 구조와 전개 디스크립터 파일을 설정하고, 제어기에 필요한 Java 클래스와 포틀릿 뷰를 위한 JSP 페이지를 자동으로 작성합니다. 새 포틀릿 프로젝트 마법사를 수행하면서 다음이 올바르게 구성되었는지 확인하십시오.

- 목표 서버에 대해 "WebSphere Portal v5.1 스택"을 선택하십시오. 이것은 고급 표시>>단추를 눌러 마법사의 첫 화면에서 지정합니다.
- WebContent/WEB-INF/에 작성된 web.xml 파일로 DB2 Alphablox에 대한 servlet 맵핑을 추가하십시오. 아래의 servlet 정의 및 맵핑 코드를 프로젝트의 web.xml에 입력하면 됩니다.

```
<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>
```

- Alphablox 태그 라이브러리를 프로젝트의 web.xml로 추가하십시오. 다음 단계를 수행하십시오.
 1. web.xml을 열고 변수 탭을 누르십시오.
 2. 맨 아래에 있는 태그 라이브러리 참조 섹션으로 스크롤하십시오.
 3. 추가를 누르십시오.
 4. 다음 테이블을 기초로 URL 및 위치에 값을 입력하십시오.

URL	위치
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

URL	위치
bloxuitld	/WEB-INF/tlds/bloxui.tld

URL과 위치로 구성된 각 쌍에 대해 완료 버튼을 눌러 태그 라이브러리 참조를 추가하고, 추가 버튼을 눌러 다음 쌍을 추가해야 합니다.

주: 다른 태그 라이브러리 및 이들의 사용에 대한 세부사항은 JavaServer 페이지 및 Blox 사용을 참조하십시오.

마지막으로 DB2 Alphablox 태그 라이브러리의 .tld 파일을 프로젝트의 WEB-INF/tlds/ 디렉토리로 복사하십시오. 파일의 위치는 다음과 같습니다.

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

여기서 <db2alphablox_dir>은 DB2 Alphablox 설치 디렉토리입니다.

다음 단계

Blox 구성요소로 포틀릿을 작성하는 방법과 필수 코드 구조에 대해 학습했습니다. 데이터베이스에서 데이터를 표시하려면 다음을 수행해야 합니다.

1. 데이터 소스를 가리키는 DB2 Alphablox를 새 데이터 소스로 정의하십시오.
2. 정의된 데이터 소스를 가리키도록 DataBlox의 dataSourceName 속성을 수정하십시오.
3. 쿼리 문자열이 있는 query 속성을 DataBlox에 추가하십시오.

데이터 소스 작성에 대한 세부사항은 *Administrator's Guide*를 참조하십시오. 해당 쿼리 작성법을 학습하려면 *Developer's Guide*의 Retrieving Data 및 *Developer's Reference*의 DataBlox 섹션을 참조하십시오.

일단 데이터가 Blox 뷰에 표시되면, Blox 태그를 사용하여 설정할 수 있는 많은 등록 정보를 탐색할 수 있습니다.

포틀릿 개발 추가 정보

DB2 Alphablox 문서에서 제공하는 정보를 사용하여 포틀릿 개발을 계속하면서 알아야 하는 일부 포틀릿 고유의 주제, 개념 및 일반 지침이 있습니다. 다음 목록은 관심있는 문서의 특정 섹션에 대한 포인터뿐 아니라 일반 개발 지침도 함께 들어 있습니다.

- 항상 포틀릿의 이름 스페이스를 사용하여 Blox의 이름을 인코딩하십시오. 이름 스페이스는 현재 J2EE 세션의 Blox 이름을 고유하게 합니다.

- 항상 픽셀을 사용하여 Blox의 높이와 너비를 설정하십시오. 포털 환경 외부에서는 "50%" 또는 "100%"처럼 백분율 값으로 너비와 높이를 설정할 수 있습니다. 포털 환경에서는 페이지에 여러 개의 포틀릿이 공존하므로 백분율 값은 작동하지 않습니다.
- 포틀릿 내에서 자원을 호출하는 데 상대 URL을 사용하지 마십시오. URL은 PortletResponse 클래스의 encodeURL() 메소드를 사용하여 인코딩되어야 합니다. 문서에서 상대 URL 사용법을 표시할 경우, 항상 URL을 인코딩해야 합니다.
- Blox 포틀릿에서 포털 테마와 유사한 테마를 사용하려면, 포털 테마 유틸리티를 사용하십시오. 이 유틸리티는 두 개의 소스에서 사용 가능합니다.

- DB2 Alphablox 홈 페이지.

관리 탭에서 일반 링크를 누르십시오. 포털 섹션에 유틸리티가 나열됩니다. 세부 사항은 *Administrator's Guide* 및 온라인 도움말을 참조하십시오.

- AlphabloxAdminPortlets.war 파일.

이것은 포털 테마 유틸리티의 포틀릿 버전입니다. DB2 Alphablox 홈 페이지에 개별적으로 로그인하지 않고도 포털에서 실행할 수 있도록 이 포틀릿을 설치할 수 있습니다.

이 유틸리티는 포털 환경의 스타일과 DB2 Alphablox의 스타일을 하나로 결합하여 Blox가 페이지에 있는 다른 포틀릿과 유사한 색상과 글꼴로 표시되도록 합니다.

- 계획 단계에서 고려해야 하는 이슈에 대해서는 *Developer's Guide*에 있는 Blox 포틀릿 태그 라이브러리 주제를 점검하십시오.
- DB2 Alphablox에는 강력한 Blox UI 모델이 있습니다. UI 모델에는 페이지의 Blox 구성요소를 눌렀을 때 지정된 URL을 로드할 수 있는 ClientLink 오브젝트가 포함됩니다. 포틀릿 환경의 경우, Blox 포틀릿 태그 라이브러리를 사용하여 ClientLink를 작성하십시오. 태그는 페이지를 새로 고친 후 링크가 부실해지지 않도록 URL 형성과 처리를 동적으로 수행합니다. 세부사항은 *Developer's Guide*에 있는 포틀릿 개발 계획 부분을 참조하십시오.
- 포틀릿 및 조치 URI의 경우, Blox 포틀릿 태그 라이브러리를 사용하여 포틀릿 링크 또는 조치 링크를 작성하십시오. 그런 다음 포털의 포틀릿 API를 사용하여 조치나 포틀릿 링크를 처리할 수 있습니다. 세부사항은 *Developer's Guide*의 Blox 포틀릿 태그 라이브러리 부분을 참조하십시오.

제 3 장 자습서: Rational Developer 도구를 사용하여 응용프로그램 빌드

Rational Developer 도구를 DB2 Alphablox 툴킷과 함께 사용하면 DB2 Alphablox 구성요소 및 DB2 Alphablox Java API를 사용하여 더 신속하게 응용프로그램을 개발할 수 있습니다.

Rational Developer(Rational Application Developer 도구 또는 Rational Web Developer)는 Eclipse 기반의 플러그인 세트인 DB2 Alphablox 툴킷과 함께 사용하여 DB2 Alphablox 기술을 사용하여 빌드된 응용프로그램을 개발 및 테스트할 수 있습니다. 이 자습서는 Rational 통합 개발 환경 구성을 안내하여 DB2 Alphablox 툴킷으로 추가된 개선된 사용자 정의 기능을 사용 가능하게 하며 태그 및 메소드도 완료할 수 있게 해 줍니다.

DB2 Alphablox 툴킷으로 Rational Developer 도구 구성을 시작하여 DB2 Alphablox 응용프로그램을 개발 및 테스트할 수 있습니다.

개발 환경 준비

DB2 Alphablox 툴킷을 설치 및 사용하기 전에 모든 필수 소프트웨어가 있는지 확인하여 개발 환경을 준비하십시오.

개발 환경을 준비하려면 다음을 수행하십시오.

1. 워크스테이션에 Rational Developer 도구(Rational Application Developer 또는 Rational Web Developer)를 설치하십시오.
2. 필요한 갱신사항을 설치하여 최소한 Rational Application Developer 또는 Rational Web Developer 버전 6.0.0.1이 있는지 확인하십시오.
3. WebSphere 5.1 통합 테스트 환경을 사용하려는 경우, 선택적 설치를 포함하도록 Rational Developer 버전을 갱신했는지 확인하십시오.

주: Rational Developer 도구의 디폴트 설치에는 WebSphere 5.1 통합 테스트 환경이 포함되지 않습니다.

이제 이 자습서에 있는 태스크를 수행할 준비가 되었습니다.

DB2 Alphablox 툴킷 설치

DB2 Alphablox 툴킷을 Rational Developer 도구와 함께 사용하면 DB2 Alphablox 콘텐츠가 있는 웹 기반의 응용프로그램을 개발하기 쉬워집니다.

DB2 Alphablox 툴킷에서는 Rational Application Developer 또는 Rational Web Developer 버전 6.0.0.1을 사용해야 합니다.

DB2 Alphablox 툴킷을 Rational Developer 도구에 설치하면, Java 메소드 및 JSP 사용자 정의 태그에 대한 콘텐츠를 완료할 수 있게 하는 다수의 수동 단계를 수행하지 않아도 됩니다. 또한, 사용자 정의 마법사 및 치트 시트는 DB2 Alphablox 콘텐츠로 응용프로그램 작성 및 WebSphere 서버 인스턴스 정의를 안내하는 데 사용될 수도 있습니다.

DB2 Alphablox 툴킷을 Rational Application Developer 또는 Rational Web Developer 에 설치하려면 다음을 수행하십시오.

1. DB2 Alphablox 설치 디스크를 워크스테이션의 CD 드라이브에 삽입하십시오.
2. plugin 디렉토리에서 UpdateSite 서브디렉토리를 찾으십시오.
3. UpdateSite 디렉토리를 하드 디스크의 편리한 위치로 복사하십시오. 예를 들어 C:\DB2Alphablox\UpdateSite로 복사하십시오.
4. Rational Developer 도구를 시작하십시오.
5. 메뉴 표시줄에서 도움말 > 소프트웨어 갱신 > 찾기 및 설치를 선택하십시오.
6. 설치 창이 열리면 설치할 새 기능 검색 옵션을 선택한 후 다음을 누르십시오.
7. 방문할 갱신 사이트 창에서 새 로컬 사이트 단추를 누른 후 UpdateSite 디렉토리 위치를 찾아보십시오.
8. 다음을 누르고 **DB2 Alphablox Toolkit** 기능을 선택한 후 다시 다음을 누르십시오.
9. 다음 창에서 라이선스 계약에 동의하고 DB2 Alphablox 툴킷 기능을 설치할 위치를 선택하십시오.
10. 설치가 완료되면 Rational Developer 도구를 다시 시작하십시오.

Rational Developer 도구를 다시 시작한 후, DB2 Alphablox 툴킷 기능을 사용할 수 있습니다.

WebSphere 통합 테스트 환경에 DB2 Alphablox 설치

이 태스크에서는 Rational Developer 도구로 WebSphere 통합 테스트 환경에 DB2 Alphablox를 설치합니다.

전제조건: 지원되는 WebSphere 통합 테스트 환경은 Rational Developer 도구로 설치되어야 합니다. DB2 Alphablox 설치 디스크에 액세스해야 합니다. *DB2 Alphablox 설치 안내서*의 사전 설치 태스크 섹션에서 설명한 바와 같이 모든 필수 사전 설치 단계를 검토하고 수행하십시오.

DB2 Alphablox는 독립형 WebSphere Application Server 또는 Rational Developer 도구 내에서 사용 가능한 WebSphere 통합 테스트 환경에 설치할 수 있습니다. 아래에서 대충 설명한 요약 단계에서는 Rational Developer 도구에서 사용 가능한 WebSphere 런타임의 DB2 Alphablox를 설치할 때 수행해야 할 설치에서의 차이점에 대해 설명합니다.

WebSphere 통합 테스트 환경에서 DB2 Alphablox를 설치하려면 다음을 수행하십시오.

다음 예외사항에 유의하면서 *DB2 Alphablox* 설치 안내서의 설치 섹션에 있는 단계를 따라 DB2 Alphablox를 설치하십시오.

1. 설치 프로그램의 WebSphere 구성 창에서 **WebSphere** 루트 디렉토리 필드에 통합 테스트 환경 서버로 사용하려는 WebSphere 런타임의 위치를 지정하십시오.

예를 들어, 디폴트 설치 시 WebSphere 통합 테스트 환경에 DB2 Alphablox를 설치하려면 `base_v6` 디렉토리의 경로를 지정하십시오. 일반적인 Rational Application Developer 설치의 경우, 경로는 다음과 유사합니다.

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

2. WebSphere 설정 창에서 WebSphere 관리자의 이름 및 암호를 지정하십시오. 이 입력 값은 WebSphere 통합 테스트 환경에서 사용되지는 않지만, DB2 Alphablox 설치 프로그램에서는 필요합니다.

중요사항: *DB2 Alphablox* 설치 안내서에서 설명한 설치 후 단계는 수행하지 마십시오.

DB2 Alphablox를 WebSphere 런타임에 설치한 후에는 Rational Developer 도구 내에서 DB2 Alphablox 응용프로그램을 테스트하는 데 필요한 WebSphere 서버 인스턴스를 작성할 수 있습니다.

WebSphere 서버 인스턴스 작성

DB2 Alphablox 콘텐츠가 들어 있는 응용프로그램이나 JSP 파일을 실행하려면, 필수 DB2 Alphablox 서비스 및 Java 클래스에 액세스할 수 있는 WebSphere 서버 인스턴스를 작성해야 합니다.

전제조건: Rational Developer 도구를 구성하십시오. DB2 Alphablox 툴킷을 설치하십시오. WebSphere 통합 테스트 환경에서 DB2 Alphablox를 설치하십시오.

DB2 Alphablox 콘텐츠가 들어 있는 응용프로그램 및 JSP 파일을 테스트하는 데 필요한 WebSphere 서버 인스턴스를 작성하려면 다음을 수행하십시오.

1. Rational Developer 도구를 여십시오.
2. 서버 탭을 누르십시오(탭이 보이지 않을 경우, 창 > 뷰 표시 > 서버를 선택).

3. 뷰 창 인쪽을 마우스 오른쪽 단추로 누른 다음 새로 작성 > 서버를 선택하십시오. 새 서버 창이 열립니다.
4. 호스트 이름으로 localhost를 입력하고 서버 유형을 선택하십시오. 다음을 누르십시오.
5. 서버 포트 번호를 입력하고 다음을 누르십시오.
6. 구성된 프로젝트 목록에 이 서버 인스턴스에서 실행하려는 사용 가능 프로젝트를 추가하십시오. 완료를 누르십시오. 새 서버 인스턴스가 서버 뷰에서 열립니다.

WebSphere 5.1 서버 인스턴스를 작성했다면 다음 단계인 "WebSphere 5.1 서버 대체 변수 작성"을 수행해야 합니다. WebSphere 6 서버 인스턴스의 경우, "DB2 Alphablox 관리자 그룹에 게스트 추가"를 진행하십시오.

WebSphere 5.1 서버 대체 변수 작성

이 태스크에서는 WebSphere 통합 테스트 환경 내에서 DB2 Alphablox를 올바르게 실행하도록 WebSphere 5.1 서버 인스턴스의 서버 대체 변수를 수정하게 됩니다.

1. 서버 뷰를 열고 수정하려는 WebSphere 5.1 서버 인스턴스를 더블 클릭하십시오.
2. 서버 인스턴스의 구성 탭을 누르십시오.
3. 관리 콘솔 사용을 선택하고 공통 테스트 클라이언트 사용을 선택 취소하십시오.
4. 서버 인스턴스의 변수 탭을 누르십시오. 대체 변수 창이 열립니다.
5. 추가 단추 (노드 설정 목록 옆에 있음)를 사용하여 두 개의 새 변수를 추가하십시오.
 - a. WS_EAR_AlphabloxPlatform 변수를 추가하고 그 값을 \$(APP_INSTALL_ROOT)/localhost/AlphabloxPlatform.ear로 설정하십시오.
 - b. 두 번째 변수 WS_EAR_ApplicationStudio를 추가하고 그 값을 \$(APP_INSTALL_ROOT)/localhost/ApplicationStudio.ear로 설정하십시오.
6. 변경사항을 저장하십시오.

WebSphere 5.1 서버 인스턴스의 경우, "WebSphere 5.1 서버 인스턴스 구성"의 단계를 수행하여 서버 인스턴스 수정을 완료해야 합니다.

WebSphere 5.1 서버 인스턴스 구성

WebSphere 5.1 서버 인스턴스의 경우, DB2 Alphablox 응용프로그램 및 파일을 실행하기 위한 서버 인스턴스를 구성해야 합니다.

전제조건: 필수 WebSphere 5.1 서버 대체 변수를 작성하십시오.

WebSphere 5.1 서버 인스턴스를 구성하려면 다음을 수행하십시오.

1. Rational Developer 도구에서 서버 뷰를 열고 구성하려는 서버 인스턴스를 시작하십시오.
2. 서버 인스턴스를 마우스 오른쪽 단추로 누른 다음 관리 콘솔 실행을 선택하십시오.
3. ID 필드에 값을 입력하지 마십시오. 확인을 누르십시오.
4. 구성 테이블 아래에 있는 확인 단추를 누르십시오.
5. 엔터프라이즈 응용프로그램 탐색 뷰를 열고 ApplicationStudio 응용프로그램 이름을 누르십시오.
6. 응용프로그램 2진을 \$(WS_EAR_ApplicationStudio)로 설정하십시오.
7. 2진에서 메타데이터 사용을 선택하십시오.
8. 구성 테이블 아래에 있는 확인 단추를 누르십시오.
9. WebSphere 관리 콘솔에 있는 저장 단추를 누르십시오.
10. 마스터 구성에 저장 창에서, 저장을 누르십시오.
11. 응용프로그램 > 엔터프라이즈 응용프로그램에 있는 WebSphere 관리 콘솔에서 AlphabloxPlatform 및 ApplicationStudio 응용프로그램을 시작하십시오.

서버 인스턴스가 구성됩니다. 다음에는 "DB2 Alphablox 관리자 그룹에 게스트 추가" 단계를 수행해야 합니다.

DB2 Alphablox 관리자 그룹에 게스트 사용자 추가

DB2 Alphablox 관리자 그룹에 게스트 사용자를 추가하면 Rational Developer 도구를 사용하는 동안 DB2 Alphablox 관리 페이지에 액세스할 수 있습니다.

전제조건: WebSphere 서버 인스턴스를 작성하십시오. WebSphere 서버 인스턴스를 구성하십시오.

DB2 Alphablox 관리자 그룹에 게스트 사용자를 추가하려면 다음을 수행하십시오.

중요사항: 게스트 사용자에게는 WebSphere 서버 제품에 대한 관리자 권한을 제공하지 마십시오.

1. 다음 telnet 명령을 사용하여 DB2 Alphablox 콘솔에 로그인하십시오. telnet localhost portNumber 여기서, portNumber는 DB2 Alphablox 설치 시 지정된 포트입니다.
2. Telnet 콘솔 프롬프트에서 DB2 Alphablox 콘솔 명령 set Administrators guest 을 입력한 다음 Enter를 누르십시오.
3. Telnet 콘솔에서 save를 입력한 다음 Enter를 누르십시오.
4. Telnet 세션을 닫으십시오.

사용자의 WebSphere 서버 인스턴스를 이제 사용할 수 있습니다.

DB2 Alphablox 응용프로그램 작성

Rational Developer에서 새 응용프로그램을 작성할 때, 응용프로그램을 올바르게 실행하려면 필수 DB2 Alphablox 콘텐츠를 추가해야 합니다.

전제조건: Rational Developer 도구에 DB2 Alphablox 툴킷을 설치하십시오.

1. Rational Developer 도구 메뉴 표시줄에서 **파일 > 새 프로젝트**를 선택하십시오. 새 프로젝트 마법사가 열립니다.
2. 웹 옵션을 확장하고 동적 웹 프로젝트를 선택한 후 다음을 누르십시오.
3. 프로젝트 이름을 입력한 후 고급 표시 단추를 누르십시오. 추가 옵션이 표시됩니다.
4. 해당 servlet 버전 및 목표 서버를 선택하십시오.
5. 다음을 누르십시오. 기능 옵션 창이 열립니다.
6. **DB2 Alphablox** 콘텐츠 옵션을 선택한 후 완료를 누르십시오.

새 웹 응용프로그램 프로젝트는 이제 DB2 Alphablox를 사용할 수 있습니다. DB2 Alphablox 태그 라이브러리 및 Blox Java API를 사용할 수 있으며, 응용프로그램 디스크립터 파일(web.xml)도 필요한 DB2 Alphablox 정보를 포함하도록 수정되었습니다. 이제 DB2 Alphablox 콘텐츠가 들어 있는 JSP 파일을 프로젝트로 추가할 수 있습니다.

DB2 Alphablox 콘텐츠로 JSP 파일 작성

이 태스크에서 Rational Developer 도구를 DB2 Alphablox 툴킷과 함께 사용하여 Blox 태그 라이브러리에 액세스할 수 있는 새 JSP 파일을 작성합니다.

전제조건: Rational Developer 도구에 DB2 Alphablox 툴킷을 설치하십시오.

1. Rational Developer의 프로젝트 탐색기 뷰에서 **파일 > 새로 작성 > JSP** 파일을 선택하십시오.
2. 파일 이름 필드에 파일 이름을 입력하십시오.
3. 고급 옵션 구성을 누른 후 다음을 누르십시오.
4. JSP 파일에 사용할 DB2 Alphablox 태그 라이브러리를 추가하십시오.
 - a. 추가 단추를 눌러 태그 라이브러리 추가 창을 여십시오.
 - b. JSP 파일에 사용할 DB2 Alphablox 태그 라이브러리를 선택하십시오.
 - c. 다음을 누르십시오.
5. 인코딩 목록에서 **ISO 10646/Unicode(UTF-8)**를 선택하십시오. UTF-8 인코딩은 DB2 Alphablox 응용프로그램을 올바르게 실행하는 데 필수입니다.
6. 완료를 누르십시오. 새 JSP 파일이 프로젝트 목록에 표시됩니다.
7. 파일 이름을 더블 클릭하여 JSP 편집기 창에서 파일을 여십시오.

8. 커서를 HTML <head> 태그 왼쪽에 두십시오. 그러나 위에서 추가된 DB2 Alphablox JSP taglib 지시문 다음에 두십시오.
9. 새 행에 Blox 헤더 태그인 <blox:header/>를 입력하십시오. JSP 파일을 실행할 때 필요한 DB2 Alphablox JavaScript 및 CSS 파일을 추가하려면 Blox 헤더 태그가 필요합니다.

이제, 선택한 DB2 Alphablox 태그 라이브러리에 액세스하기 위한 새 JSP 파일을 사용할 수 있습니다. Rational Developer 도구의 콘텐츠 보조 기능을 사용하면 선택한 태그 라이브러리의 Blox 태그와 태그 속성을 삽입할 수 있습니다.

DB2 OLAP Server 및 Essbase 데이터 소스 액세스

WebSphere 5.1 서버 인스턴스를 사용할 때 Rational Developer 도구로 DB2 OLAP Server™ 또는 Essbase 데이터 소스에 액세스하려면, 필수 클라이언트 라이브러리를 로드하는 시작 일괄처리 파일을 작성해야 합니다.

전제조건: DB2 Alphablox 툴킷을 설치하십시오. WebSphere 5.1 통합 테스트 환경에 DB2 Alphablox를 설치하십시오. WebSphere 5.1 서버 인스턴스를 구성하십시오. 개발 머신에 IBM DB2 OLAP Server 또는 Hyperion Essbase를 설치하십시오.

Rational Developer 도구에서 WebSphere 5.1 통합 테스트 환경의 제한사항으로 인해, DB2 OLAP Server 또는 Hyperion Essbase 데이터 소스에 액세스하려면 Rational Developer 도구에 대한 시작 일괄처리 파일을 작성해야 합니다. 일괄처리 파일을 사용하여 Rational Developer 도구를 시작할 때 필수 Essbase 클라이언트 라이브러리는 Java 라이브러리 경로에 추가됩니다.

WebSphere 5.1 서버 인스턴스로 DB2 OLAP Server 또는 Essbase에 액세스할 수 있게 하는 일괄처리 파일을 작성하려면 다음을 수행하십시오.

1. 문서 편집기를 사용하여 새 텍스트 문서를 작성하십시오.
2. DB2 Alphablox 설치 디렉토리에 있는 DB2 Alphablox aassetup.bat 파일을 호출할 코드 행을 추가하십시오. 예를 들어, 다음 코드는 지정된 DB2 Alphablox 설치 디렉토리에 있는 aassetup.bat 파일을 실행합니다.

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

3. RAD를 시작하는 RAD rationalsdp.exe 파일을 호출할 두 번째 행을 추가하십시오. 예를 들면, 다음 코드는 Rational Developer 도구를 실행합니다.

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

4. 워크스테이션 데스크탑(또는 다른 편리한 위치)에 startRAD.bat로 이 파일을 저장하십시오.

startRAD.bat 파일을 더블 클릭하면, DB2 Alphablox aassetup.bat가 실행되고 필수 경로 및 환경 변수를 설정한 후 Rational Application Developer가 시작됩니다. DB2

Alphablox에 사용된 Essbase 클라이언트 라이브러리 버전은 데이터 소스로 사용 중인 DB2 OLAP Server(또는 Hyperion Essbase) 버전과 일치해야 합니다. DB2 Alphablox에 사용되는 Essbase 클라이언트 라이브러리를 수정하려면 `db2_alphablox\analytics\bin` 디렉토리에 있는 DB2 OLAP Server / Essbase 클라이언트 라이브러리 유틸리티 (`ChangeEssbase.bat`)를 실행하십시오. 여기서 `db2_alphablox`는 DB2 Alphablox 설치의 루트 디렉토리입니다. 이 유틸리티는 작성한 `startRAD` 일괄처리 파일을 사용하여 RAD를 시작할 때 실행되는 일괄처리 파일 중 하나를 수정합니다.

위에서 설명한 완전한 `startRAD.bat` 파일 예가 다음에 나와 있습니다.

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

DB2 OLAP Server 또는 Essbase 데이터 소스에 액세스할 때마다, `startupRAD.bat` 파일은 Rational Developer 도구를 시작하는 데 사용되어야 합니다.

제 4 장 자습서: DB2 Cube Views를 사용한 DB2 Alphablox 큐브 빌드

이 자습서에서는 DB2 Cube Views 샘플 데이터베이스를 사용하여 빌드된 DB2 Alphablox 큐브 작성을 안내합니다.

자습서의 태스크에는 사용자 정의 DB2 Alphablox 큐브 빌드에 대한 세부사항이 나와 있지 않습니다. 대신 DB2 Alphablox 큐브 서버 성능을 탐색하는 데 사용할 수 있는 DB2 Alphablox 큐브를 신속하게 작성하는 방법을 알려주는 것이 목표입니다. 결과 데이터 소스는 DB2 Alphablox 응용프로그램을 테스트하고 빌드하는 데 사용할 수도 있습니다.

전제조건:

- DB2 Alphablox를 설치하십시오. 설치 세부사항에 대해서는 설치 안내서를 참조하십시오.
- DB2 Cube Views CVSAMPLE 샘플 데이터베이스가 설치된 지원되는 DB2 Cubes Views 구현에 대한 액세스 권한을 확보하십시오. DB2 Cube Views에서 지원되는 버전에 대한 세부사항에 대해서는 설치 안내서를 참조하십시오.

자습서에서는 DB2 Cube Views CVSAMPLE 샘플 데이터베이스에 따라 DB2 Alphablox 큐브를 빌드하는 방법을 배우게 됩니다. 자습서를 통해 다음 태스크를 배우게 됩니다.

DB2 관계 데이터 소스 정의

이 태스크에서는 DB2 데이터베이스에 대한 DB2 Alphablox에 데이터 소스 정의를 정의합니다.

전제조건: 필수 DB2 JDBC 드라이버는 DB2 Alphablox에 액세스할 수 있어야 합니다.

DB2 Alphablox 큐브에서는 기초적인 관계 데이터 소스가 DB2 Alphablox 데이터 소스로 미리 정의되어야 합니다. DB2 Alphablox 큐브는 관계형 데이터베이스에서 사용 가능한 데이터 및 메타데이터를 사용하여 생성됩니다.

DB2 데이터베이스에 대해 DB2 Alphablox 데이터 소스를 정의하려면 다음을 수행하십시오.

1. DB2 Alphablox 관리 페이지에 관리 사용자로서(또는 관리자 권한을 가진 사용자로서) 로그인하십시오.

2. 관리 탭을 누르십시오.
3. 데이터 소스 링크를 누르십시오.
4. 작성 단추를 누르십시오.
5. 어댑터 메뉴에서 데이터베이스 서버에 해당하는 IBM® DB2 JDBC 드라이버를 선택하십시오.

IBM DB2 JDBC Type 4 드라이버 또는 **iSeries의 IBM DB2 UDB** 드라이버를 선택하십시오.

6. 데이터 소스 이름 필드에 데이터 소스에 사용할 이름으로 CVSAMPLE을 입력하십시오.
7. 서버 이름, 포트 번호 및 데이터베이스 이름에 적합한 값을 입력하십시오(데이터베이스 이름은 CVSAMPLE이어야 함).

주: 이러한 필드에 대한 올바른 값을 판별하는 데 도움이 필요하다면 데이터베이스 관리자에게 문의하십시오.

8. 디폴트 사용자 이름 및 디폴트 암호를 입력하십시오.

사용자 이름과 암호는 관계형 데이터베이스에서 유효해야 합니다. 디폴트 사용자 이름과 암호는 DB2 Alphablox 큐브가 관계형 데이터베이스에 액세스할 때 항상 사용됩니다. 사용하는 데이터베이스 사용자는 데이터베이스에 대해 읽기 액세스 권한을 가지고 있어야 합니다.

주: 지정된 관계 데이터 소스가 DB2 Alphablox 큐브를 채우기 위해 사용되면 **DB2 Alphablox** 사용자 이름 및 암호 사용 값은 무시됩니다. 액세스 제어 목록 (ACL)은 DB2 Alphablox 큐브에 대한 액세스를 제한할 때 사용될 수 있습니다. ACL에 대한 추가 정보는 *Administrator's Guide*를 참조하십시오.

9. 데이터 소스가 DB2 Alphablox 큐브를 채우기 위해 사용될 때는 최대 행 및 최대 컬럼 값은 무시됩니다. 값을 입력할 수는 있지만 그 값은 다른 응용프로그램이 데이터 소스를 사용할 때 사용됩니다.
10. DB2 Alphablox 로그 파일에 JDBC 로그 정보를 기록하려는 경우가 아니면 **JDBC 추적 사용 가능** 값을 **아니오**로 설정하십시오. 문제가 있어서 그 원인을 디버깅해야 하는 경우에만 JDBC 추적을 사용하십시오.
11. 저장 단추를 눌러 데이터 소스 정의를 저장하십시오.

이제 CVSAMPLE에 대해 DB2 Alphablox 데이터 소스를 정의했습니다. 이제 이 DB2 데이터 소스에서 Cube Views 큐브 메타데이터에 액세스하는 데 필요한 DB2 Alphablox 큐브 정의를 작성할 수 있습니다.

Alphablox 큐브 서버 어댑터 데이터 소스 정의

이 태스크에서는 Alphablox 큐브 서버 어댑터를 사용하는 DB2 Alphablox 데이터 소스 정의를 정의합니다.

전제조건: DB2 CVSAMPLE 데이터베이스에 대해 DB2 Alphablox 데이터 소스 정의를 작성하십시오.

DB2 Alphablox 큐브 서버 어댑터 데이터 소스를 정의하려면 다음을 수행하십시오.

1. DB2 Alphablox 관리 페이지에 관리 사용자로서(또는 관리자 권한을 가진 사용자로서) 로그인하십시오.
2. 관리 탭을 누르십시오.
3. 데이터 소스 링크를 누르십시오.
4. 작성 단추를 누르십시오.
5. 어댑터 메뉴에서 **Alphablox** 큐브 서버 어댑터 옵션을 선택하십시오.
6. 데이터 소스 이름 필드에서 데이터 소스에 사용할 이름으로 DB2AlphabloxCubes를 입력하십시오.
7. 저장 단추를 눌러 데이터 소스 정의를 저장하십시오.

DB2 Alphablox 큐브에 액세스하는 데 사용할 수 있는 DB2 Alphablox 데이터 소스를 정의했습니다. 그 다음에는 액세스할 수 있는 DB2 Alphablox 큐브를 정의해야 합니다.

DB2 Alphablox 큐브 정의

이 태스크에서는 DB2 Cube Views CVSAMPLE 큐브의 메타데이터에 따라 DB2 Alphablox 큐브를 정의합니다.

전제조건: DB2 관계형 데이터 소스를 정의하십시오. DB2 Alphablox 큐브 서버 어댑터 데이터 소스를 정의하십시오.

DB2 Alphablox 큐브의 일반 등록 정보를 정의하려면 다음을 수행하십시오.

1. DB2 Alphablox 관리 페이지에 관리 사용자로서(또는 관리자 권한을 가진 사용자로서) 로그인하십시오.
2. 관리 탭을 누르십시오.
3. 큐브 링크를 누르십시오.
4. 작성 단추를 누르십시오. 큐브 관리 창이 열립니다.
5. 다음과 같이 새 큐브를 정의하십시오.
 - a. **DB2 Alphablox** 큐브 이름 필드에 CVSales를 입력하십시오.

- b. **DB2 Alphablox** 큐브 이름 필드 옆에 있는 **사용 가능** 옵션을 선택하십시오. 이 옵션을 선택하면 서버가 재시작할 때 큐브가 자동으로 시작됩니다.
 - c. 이 지습서에서 작성한 관계 데이터 소스인 MyDB2를 **관계 데이터 소스** 메뉴에서 선택하십시오.
 - d. 보안 역할 옵션을 선택하지 않은 채로 두십시오. 이 옵션은 사용자가 특정 큐브에 액세스하지 못하도록 제한하는 데 사용될 수 있습니다.
6. 다음과 같이 DB2 Cube Views 설정을 사용 가능하게 한 후 메타데이터를 지정하십시오.
- a. **DB2 Cube Views** 설정 **사용** 옵션을 선택하십시오.
 - b. 큐브 모델 메뉴에서 **CVSAMPLE.Sales**를 선택하십시오.
 - c. 큐브 메뉴에서 **일반 판매액** 큐브를 선택하십시오.
 - d. **비즈니스 이름 사용** 단일 선택 단추를 선택하여 이름을 지정하십시오. 이 옵션을 선택하면 일반적으로 좀 더 의미있고 읽기 쉬운 구성원 이름을 사용합니다.
 - e. **큐브 정의 импорт** 단추를 누르십시오. 이 옵션을 사용하면 큐브 정의를 импорт하고 DB2 Alphablox 큐브에서 도량 단위와 차원을 미리 채울 수 있습니다. импорт된 큐브 정의는 DB2 Cube Views™ 메타데이터의 지원을 기반으로 DB2 Alphablox가 일치시킬 수 있는 한 가장 가깝게 DB2 Cube Views 큐브를 반영합니다. 큐브 작업에 경험이 있으면, 원하는 대로 도량 단위와 차원을 수정할 수 있습니다.
 - f. **импорт 로그 표시** 단추를 눌러 импорт 조작과 관련된 정보 및 디버깅 메시지를 지정하는 로그를 보십시오. 이 지습서에서는 사용자가 이 기능에 익숙해지도록 돕기 위해 이 단계를 포함하였습니다.
 - g. **시작, 재빌드 및 편집 시 큐브 정의 импорт** 옵션을 선택하십시오. 이 옵션을 사용하면 DB2 Alphablox 큐브는 편집을 위해 시작, 재빌드 또는 열릴 때마다 DB2 Alphablox 큐브에서 최신 DB2 Cube Views 정의를 사용하게 됩니다. DB2 Alphablox 및 DB2 Cube Views에 익숙하면, 큐브 정의를 импорт하고 이를 사용자 정의할 수 있습니다.
7. **저장** 단추를 눌러 DB2 Alphablox 큐브 정의를 저장하십시오.

이제 DB2 Alphablox 큐브 정의를 작성했습니다. 이제 새 CVSales 큐브를 시작할 수 있습니다.

DB2 Alphablox 큐브 시작

이 태스크에서는 DB2 Alphablox 관리 페이지를 사용하여 CVSales 큐브를 시작합니다.

전제조건: DB2 관계형 데이터 소스를 정의하십시오. DB2 Alphablox 큐브 서버 어댑터 데이터 소스를 정의하십시오. DB2 Alphablox 큐브를 정의하십시오.

CVSales 큐브를 시작하려면 다음을 수행하십시오.

1. DB2 Alphablox 관리 페이지에 관리 사용자로서(또는 관리자 권한을 가진 사용자로서) 로그인하십시오.
2. 관리 탭을 누르십시오.
3. 런타임 관리 섹션에서, 큐브 링크를 누르십시오.
4. **DB2 Alphablox** 큐브 목록에서 시작하려는 DB2 Alphablox 큐브를 선택하십시오.
5. 시작 단추를 누르십시오. 큐브가 시작되면 상태 필드에 실행 중이 표시됩니다.

지금 샘플 DB2 Alphablox 큐브를 실행 중입니다. 이제 방금 작성한 DB2 Alphablox 큐브를 사용하여 응용프로그램 빌드를 시작할 수 있습니다. Query Builder 응용프로그램을 사용하고, 데이터 소스로 큐브를 선택한 후 큐브에 대해 MDX 쿼리를 실행하여 새 큐브에서 빠른 점검을 수행할 수 있습니다.

주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다.

IBM은 다른 국가에서는 이 자료에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다는 것이 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation,

Licensing, 2-31

Roppongi 3-chome, Minato-ku,

Tokyo 106-0032, Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 이 책을 현상 태대로 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에 설명한 제품 및(또는) 프로그램을 사전 통지없이 언제든지 개선 및(또는) 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들어, 사용료 지불 등)에 따라 사용될 수 있습니다.

이 정보에 기술된 라이선스가 있는 프로그램 및 이 프로그램에 대해 사용 가능한 모든 라이선스가 있는 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 레벨 상태의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한, 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 본인의 특정 환경에 대해 해당 데이터를 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 대해서는 확신할 수 없습니다. 비 IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이 예제에는 가능한 완벽하게 개념을 설명하기 위해 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연히 일치하는 것입니다.

이 정보에는 여러 가지 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스에 부합하는 응용프로그램을 개발, 사용, 마케팅 및 배포하기 위한 목적으로 IBM에 추가 비용없이 어떤 형태로든 이러한 샘플 프로그램을 복사, 수정 및 배포할 수 있습니다. 이러한 예제는 모든 조건하에서 철저히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 암시하지 않습니다. 귀하는 IBM의 응용프로그램 프로그래밍 인터페이스 (API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용없이 이러한 샘플 응용프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다.

상표

다음 용어는 미국 또는 기타 국가에서 사용되는 IBM Corporation의 상표 또는 등록 상표입니다.

DB2
IBM

DB2 OLAP Server
WebSphere

DB2 Universal Database

Alphablox 및 Blox는 미국 또는 기타 국가에서 사용되는 Alphablox Corporation의 상표 또는 등록상표입니다.

Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, inc.의 상표입니다.

Linux[®]는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

색인

[카]

쿼리 빌더, DHTML

사용 4

자습서

응용프로그램 개발 1

쿼리

쿼리 빌더를 사용한 생성 4

쿼리 빌더

사용 4

[파]

포틀릿

JSP 구조 15

샘플 13



프로그램 번호: 5724-L14

GA30-3005-01

