

IBM DB2 Alphablox



Guide d'initiation

Version 84

IBM DB2 Alphablox



Guide d'initiation

Version 84

Remarque :

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant dans la section «Remarques», à la page 33.

Deuxième édition - mars 2006

Réf. US : GC18-9607-01

LE PRESENT DOCUMENT EST LIVRE EN L'ETAT SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. Les informations qui y sont fournies sont susceptibles d'être modifiées avant que les produits décrits ne deviennent eux-mêmes disponibles. En outre, il peut contenir des informations ou des références concernant certains produits, logiciels ou services non annoncés dans ce pays. Cela ne signifie cependant pas qu'ils y seront annoncés.

Pour plus de détails, pour toute demande d'ordre technique, ou pour obtenir des exemplaires de documents IBM, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial.

Vous pouvez également consulter les serveurs Internet suivants :

- <http://www.fr.ibm.com> (serveur IBM en France)
- <http://www.can.ibm.com> (serveur IBM au Canada)
- <http://www.ibm.com> (serveur IBM aux Etats-Unis)

*Compagnie IBM France
Direction Qualité
Tour Descartes
92066 Paris-La Défense Cedex 50*

© Copyright IBM France 2006. Tous droits réservés.

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

Table des matières

Avis aux lecteurs canadiens	v	Préparation de votre environnement de développement	21
Chapitre 1. Tutoriel : Création de votre première application	1	Installation du DB2 Alphablox Toolkit	21
Définition de votre application	1	Installation de DB2 Alphablox dans les environnements de test intégrés de WebSphere	22
Accès aux données	2	Création d'une instance de serveur WebSphere	23
Création de votre page d'accueil d'application	2	Création de variables de substitution de serveur WebSphere 5.1	24
Définition d'une page d'accueil par défaut	3	Configuration d'instances de serveur WebSphere 5.1	24
Création de votre première vue analytique	4	Ajout de l'utilisateur invité au groupe des administrateurs de DB2 Alphablox	25
Structure de votre première page de vue analytique	7	Création d'applications DB2 Alphablox	25
Création de votre deuxième vue analytique	9	Création de fichiers JSP avec du contenu DB2 Alphablox	26
Récapitulatif	10	Accès aux sources de données DB2 OLAP Server et Essbase	26
Chapitre 2. Tutoriel : Création de votre premier portlet contenant des composants Blox	11	Chapitre 4. Tutoriel : Création d'un cube DB2 Alphablox à l'aide de DB2 Cube Views.	29
Installation des exemples de portlets	11	Définition d'une source de données relationnelles DB2	29
Exécution des exemples de portlets	12	Définition d'une source de données d'adaptateur de serveur de cube Alphablox	30
Etude de la structure d'une page JSP de portlet contenant un composant Blox	12	Définition d'un cube DB2 Alphablox	31
Création de votre propre page JSP de portlet contenant des composants Blox.	15	Démarrage de votre cube DB2 Alphablox	32
Création d'un projet de type portlet afin d'utiliser des composants Blox	16	Remarques	33
Configuration d'un projet de type portlet à l'aide de Rational Application Developer	17	Marques	35
Etapes suivantes.	18	Index	37
Conseils relatifs au développement de portlets	19		
Chapitre 3. Tutoriel : Création d'applications à l'aide des outils Rational Developer	21		

Avis aux lecteurs canadiens

Le présent document a été traduit en France. Voici les principales différences et particularités dont vous devez tenir compte.

Illustrations

Les illustrations sont fournies à titre d'exemple. Certaines peuvent contenir des données propres à la France.

Terminologie

La terminologie des titres IBM peut différer d'un pays à l'autre. Reportez-vous au tableau ci-dessous, au besoin.

IBM France	IBM Canada
ingénieur commercial	représentant
agence commerciale	succursale
ingénieur technico-commercial	informaticien
inspecteur	technicien du matériel

Claviers

Les lettres sont disposées différemment : le clavier français est de type AZERTY, et le clavier français-canadien de type QWERTY.








OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

- les pages de codes 850 (multilingue) et 863 (français-canadien),
- le code pays 002,
- le code clavier CF.

Nomenclature

Les touches présentées dans le tableau d'équivalence suivant sont libellées différemment selon qu'il s'agit du clavier de la France, du clavier du Canada ou du clavier des États-Unis. Reportez-vous à ce tableau pour faire correspondre les touches françaises figurant dans le présent document aux touches de votre clavier.

France	Canada	Etats-Unis
 (Pos1)		Home
Fin	Fin	End
 (PgAr)		PgUp
 (PgAv)		PgDn
Inser	Inser	Ins
Suppr	Suppr	Del
Echap	Echap	Esc
Attn	Intrp	Break
Impr écran	ImpEc	PrtSc
Verr num	Num	Num Lock
Arrêt défil	Défil	Scroll Lock
 (Verr maj)	FixMaj	Caps Lock
AltGr	AltCar	Alt (à droite)

Brevets

Il est possible qu'IBM détienne des brevets ou qu'elle ait déposé des demandes de brevets portant sur certains sujets abordés dans ce document. Le fait qu'IBM vous fournisse le présent document ne signifie pas qu'elle vous accorde un permis d'utilisation de ces brevets. Vous pouvez envoyer, par écrit, vos demandes de renseignements relatives aux permis d'utilisation au directeur général des relations commerciales d'IBM, 3600 Steeles Avenue East, Markham, Ontario, L3R 9Z7.

Assistance téléphonique

Si vous avez besoin d'assistance ou si vous voulez commander du matériel, des logiciels et des publications IBM, contactez IBM direct au 1 800 465-1234.

Chapitre 1. Tutoriel : Création de votre première application

Ce tutoriel présente les composants Blox et décrit la procédure à suivre pour créer rapidement une application DB2 Alphablox.

Pour créer des applications DB2 Alphablox, vous allez utiliser la technologie JSP (JavaServer Pages). Toutefois, pour ce tutoriel, il n'est pas nécessaire de connaître cette technologie. Ce tutoriel décrit les aspects de JSP que vous devez maîtriser pour commencer à créer des applications analytiques. Pour parfaire vos connaissances en matière de technologie JSP, vous disposez d'un grand nombre d'ouvrages de qualité.

Prérequis

Les différentes étapes de ce tutoriel supposent que vous utilisiez DB2 Alphablox sur un serveur WebSphere Application Server. Si vous utilisez un serveur d'applications BEA WebLogic ou Apache Tomcat, vous serez peut-être obligé de modifier les procédures suivantes pour tenir compte des différences de votre serveur.

Avant de commencer à suivre ce tutoriel, vous devez installer un cube DB2 Alphablox (qcc_2003) et sa source de données relationnelle sous-jacente (qcc2003-rdb). Le répertoire `sampledata/qcc/acs` du CD d'installation de DB2 Alphablox contient des fichiers de base de données exemple et des instructions (disponibles dans les fichiers `readme.txt`). Ces dernières concernent la configuration du cube DB2 Alphablox et de la base de données relationnelle nécessaires pour ce tutoriel.

Définition de votre application

Pour créer une application à l'aide d'une méthode de développement J2EE, vous devez créer une arborescence dotée d'un répertoire WEB-INF contenant un fichier descripteur d'application (`web.xml`). Dans DB2 Alphablox, la méthode la plus simple pour cela consiste à créer une application dans la page Application des pages d'administration de DB2 Alphablox.

Pour créer une application et un dossier, procédez comme suit :

1. Créez une application, appelée MyApp, dans DB2 Alphablox en suivant la procédure décrite à la section Application Definitions du manuel *Administrator's Guide*.
 - Entrez MyApp dans la zone **Nom**.
 - Entrez My App (avec un espace) dans la zone du nom d'affichage, qui définit le libellé de l'application tel qu'il apparaît dans la liste de la page Applications.
2. Cliquez sur l'onglet **Applications** situé dans l'angle supérieur gauche de la page. La liste des applications disponibles apparaît.
3. Cliquez sur le nom de l'application nouvellement créée (My App) dans la liste des applications. Etant donné que vous n'avez créé encore aucun fichier, le répertoire des fichiers est vide.
4. Accédez au dossier de la nouvelle application situé sur votre serveur d'applications (pour WebSphere, le dossier se trouve dans le répertoire

installedApps de WebSphere). L'application comprend un répertoire WEB-INF contenant un fichier web.xml, ainsi que le répertoire tlds. Le fichier web.xml définit les informations de l'application, tandis que le répertoire tlds contient le fichier descripteur des bibliothèques de balises Blox (blox.tld), qui définit les balises Blox à utiliser pour la création de vues analytiques dans ce tutoriel. Le répertoire contient d'autres fichiers descripteurs de bibliothèques de balises (TLD) associés à d'autres bibliothèques de balises Blox (non utilisées dans ce tutoriel), notamment bloxform.tld, bloxlogic.tld, bloxreport.tld et bloxui.tld.

Vous avez créé une structure d'application DB2 Alphablox à laquelle il est possible d'ajouter des vues analytiques.

Accès aux données

Pour ce tutoriel, vous allez utiliser la source de données du cube DB2 Alphablox créée avant de commencer ce tutoriel. La base de données relationnelles et le serveur Alphablox Cube doivent être en cours d'exécution. Le cube DB2 Alphablox utilisé doit également être en cours d'exécution.

Pour vérifier que les sources de données requises sont disponibles et en cours d'exécution, procédez comme suit :

1. Ouvrez votre navigateur et affichez les pages d'administration de DB2 Alphablox.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur **Sources de données**.
4. Sur la page des sources de données, recherchez la source qcc2003-rdbdata dans le menu se trouvant sur le côté gauche de la page. Si vous ne la trouvez pas, revoyez les procédures décrites dans le fichier readme.txt se trouvant dans le répertoire sampledata/qcc/acs du CD d'installation de DB2 Alphablox.
5. Cliquez sur qcc2003-rdb dans la liste des sources de données disponibles puis sur le bouton de test de la source de données sélectionnée. Si la base de données est en cours d'exécution et que la définition de la source de données est configurée correctement, un message indiquant que l'opération a réussi s'affiche. Si un message d'erreur s'affiche, vérifiez la définition de la source de données (pour cela, sélectionnez la source de données et cliquez sur le bouton **Modifier**) ou assurez-vous que la base de données est correctement configurée avec un utilisateur disposant de droits d'accès à la base de données.
6. Sélectionnez la source de données qcc2003-acs dans la liste des sources de données puis cliquez sur le bouton de test de la source de données sélectionnée. Si la source de données du serveur Alphablox Cube est correctement configurée, un message indiquant que l'opération a réussi s'affiche. Si un message d'erreur s'affiche, vérifiez la définition de la source de données.

Une fois que le test des sources de données a abouti, vous êtes prêt à commencer à apprendre à créer quelques vues analytiques DB2 Alphablox simples.

Création de votre page d'accueil d'application

Cette procédure vous permet de créer une page d'accueil simple en ayant recours à des compétences de base en matière de programmation Web.

Pour ce tutoriel, il suffit de créer une structure d'application Web très simple. Sur la page d'accueil, créez deux liens permettant d'accéder aux deux vues analytiques créées dans ce tutoriel.

Pour créer une page d'accueil simple, procédez comme suit :

1. A l'aide d'un éditeur de texte ou de votre outil de développement préféré, attribuez à la page le nom `index.html`. Pour ce tutoriel, une page simple suffit.
2. Modifiez le fichier et insérez le titre de l'application (par exemple, "Mon application DB2 Alphablox") et deux liens pointant vers les deux vues analytiques que vous allez créer.

Copiez et collez ou entrez le code suivant dans le fichier `index.html`.

```
<html>
<head>
<title>Mon application DB2 Alphablox</title>
</head>
<body>
<h2>Mon application DB2 Alphablox</h2>
<p>
<a href="PresentBloxView.jsp">Vue PresentBlox simple</a>
</p>
<p>
<a href="ChartView.jsp">Vue graphique personnalisée</a>
</p>
</body>
</html>
```

3. Enregistrez ce fichier dans le répertoire MyApp créé lors de la définition de l'application.
4. Ouvrez votre navigateur Web et affichez la page d'accueil de DB2 Alphablox. Par défaut, le navigateur affiche l'onglet **Applications**.
5. Cliquez sur le lien **My App** pour vérifier que le fichier figure dans la liste des répertoires. Cliquez sur le lien **index.html** pour afficher votre page d'accueil. Si DB2 Alphablox est installé sur un serveur IBM WebSphere Application Server ou BEA WebLogic, les listes de répertoires doivent être activées avant de suivre cette étape. Pour les installations DB2 Alphablox sur Apache Tomcat, aucune procédure supplémentaire n'est requise. Si une erreur HTTP 403 s'affiche (les listes de répertoires ne sont pas autorisées par défaut), vous pouvez modifier le serveur afin d'autoriser les listes de répertoires ou simplement ajouter le nom du fichier `myapp.html` à la fin de l'URL (par exemple, `http://localhost:9080/MyApp/myapp.html`). La nouvelle page s'affiche lors de l'utilisation de l'adresse complète du lien.

Vous disposez maintenant d'une page d'accueil comportant des liens vers les deux vues analytiques que vous allez créer ultérieurement.

Définition d'une page d'accueil par défaut

Dans les procédures précédentes, vous accédez directement à la page d'accueil en cliquant sur le lien du fichier dans le répertoire d'application MyApp ou en indiquant le fichier de la page d'accueil, `MyApp.html`, dans l'adresse URL. Dans cette procédure, vous allez indiquer la page d'accueil par défaut de l'application.

Pour définir la page d'accueil par défaut de l'application :

1. Ouvrez votre navigateur et affichez la page d'accueil de DB2 Alphablox.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Applications**.

4. Sélectionnez **MyApp** dans la liste des applications et cliquez sur le bouton **Modifier** situé sous cette liste
5. Dans la page de modification d'application de MyApp, entrez `index.html` dans la zone de l'URL de la page d'accueil.
6. Cliquez sur le bouton **Enregistrer** pour enregistrer les modifications effectuées.
7. Cliquez sur l'onglet principal Applications, onglet du dossier situé à l'extrémité gauche, pour revenir à la page Applications.
8. Cliquez sur le nom de l'application **My App**. L'application s'ouvre désormais directement dans la page d'accueil définie.

Une fois la page d'accueil définie, le serveur sait où exécuter l'application et fait en sorte qu'elle s'ouvre dans cette page, même si l'utilisateur ne parvient pas à en indiquer l'URL. De cette manière, si un utilisateur entre l'adresse URL `http://votreNomServeur/MyApp/` dans la barre d'adresse du navigateur, l'application s'ouvrira dans la page d'accueil définie, `MyApp.html`.

Dans les rubriques suivantes, vous commencerez à créer vos premières vues analytiques DB2 Alphablox à l'aide de balises Blox simples générées à l'aide de l'application DB2 Alphablox Query Builder.

Création de votre première vue analytique

Cette procédure permet de créer une vue analytique DB2 Alphablox affichant une grille et un graphique associés dans le composant PresentBlox.

Conditions préalables : Consultez l'introduction du document *Developer's Guide* pour obtenir plus d'informations sur DB2 Alphablox et ses composants. Vous y trouverez une brève description de la fonctionnalité DB2 Alphablox et obtiendrez plus d'informations sur le mode de fonctionnement des bibliothèques de balises personnalisées Blox pouvant être utilisées lors de la génération des applications analytiques utilisant la technologie JSP.

Pour la première vue analytique à créer, vous allez utiliser l'application DB2 Alphablox Query Builder pour générer une balise personnalisée JSP définie pour un composant PresentBlox, composant Blox le plus utilisé pour la présentation de données. Un composant PresentBlox peut inclure des vues de grille et de graphique, un panneau d'agencement de données, des barres d'outils et d'autres composants imbriqués. Les vues de grille et de graphique représentent différentes vues synchronisées des mêmes données.

Pour créer votre première vue analytique à l'aide de Query Builder, procédez comme suit :

1. Ouvrez un navigateur et affichez les pages d'administration de DB2 Alphablox. Par défaut, l'onglet Applications est visible.
2. Cliquez sur DB2 Alphablox Query Builder pour ouvrir l'application.
3. Cliquez sur le bouton **Paramètres de connexion**. La fenêtre Connexion de base de données s'affiche.
4. Sélectionnez la source de données `qcc2003-acs` puis cliquez sur le bouton **Connexion**. La fenêtre se ferme et la mention "Connecté" doit s'afficher dans la section Statut de la base de données de la page Query Builder.

Remarque : Il n'est pas nécessaire de renseigner les zones **Graphique**, **Catalogue**, **Nom d'utilisateur** et **Mot de passe** car les valeurs

sont extraites automatiquement de la définition de la source de données sélectionnée. Ne sélectionnez pas la case à cocher **Exécuter la requête par défaut**.

5. Recherchez la section Requête sous la section Statut. Prenez connaissance des informations suivantes :
 - Pour ce tutoriel, vous utilisez un cube DB2 Alphablox (qcc2003-acs). Sous l'en-tête de section Requête, une liste des cubes disponibles est disponible. Le cube qcc2003-acs doit faire partie de cette liste.
 - Une chaîne de requêtes par défaut associée à cette source de données s'affiche dans la fenêtre de texte. Etant donné que la source de données est un cube DB2 Alphablox, le premier cube de la liste alphabétique est ajouté à l'instruction de requête (par exemple, "select from [nomSourcedonnees]").

Remarque : Lorsque la fonction de regroupement de connexions est activée dans le serveur DB2 Alphablox Cube, vous ne pouvez pas utiliser le bouton **Afficher la requête par défaut** pour obtenir une requête par défaut pour un cube DB2 Alphablox.

 - Lorsque des données s'affichent dans le composant PresentBlox dans la partie inférieure de la page, utilisez l'interface utilisateur standard pour permuter les axes, explorer vers le haut ou vers le bas, déplacer les dimensions entre les différents axes, etc.
 - Pour afficher les résultats de la requête, cliquez sur le bouton Exécuter la requête. L'ensemble de résultats s'affiche dans le composant PresentBlox dans la partie inférieure de la page.
6. Entrez qcc2003-acs entre crochets (par exemple, select from [qcc2003-acs]) puis cliquez sur le bouton **Exécuter la requête**.

La section PresentBlox doit afficher une vue par défaut créée en fonction de la requête.
7. Utilisez l'interface utilisateur PresentBlox pour créer une vue.

Par exemple, vous pouvez déplacer la dimension Products vers l'axe des lignes dans le panneau Agencement de données se trouvant sur la gauche. Vous pouvez ensuite déplacer la dimension Time (Calendar) vers l'axe des colonnes. Vous pouvez ensuite souhaiter déplacer la dimension Measures vers l'axe des pages. Un menu de mesures s'affiche alors dans le panneau Page, qui comporte les filtres de page, sous la zone d'affichage. Sélectionnez deux autres dimensions et ajoutez-les à l'axe des pages afin que trois filtres de page s'affichent au dessus. Si vous n'êtes pas un utilisateur averti de l'interface DB2 Alphablox, sélectionnez **Aide** dans la barre d'outils PresentBlox pour obtenir plus d'informations sur l'utilisation de l'interface.
8. Une fois la vue créée, notez l'instruction de requête qui s'affiche dans la section Requête. Si la requête n'est pas dynamiquement générée (l'option Mettre automatiquement à jour les requêtes n'est pas sélectionnée), vous pouvez cliquer sur le bouton **Afficher la requête actuelle** pour afficher la requête en cours. La zone de texte affiche l'instruction de requête requise pour le développement de la vue de données en cours. Vous pouvez copier et coller des instructions de requête provenant de la zone Requête à utiliser lors du développement d'applications DB2 Alphablox.
9. Cliquez sur le bouton **Générer l'étiquette du blox**. Une boîte de dialogue comportant la balise (et les balises imbriquées) s'affiche afin de reproduire l'agencement PresentBlox et l'ensemble de résultats.

10. Ouvrez un nouveau fichier dans votre éditeur de texte et ajoutez le texte suivant au fichier puis sauvegardez ce dernier en lui attribuant le nom `PresentBloxView.jsp`. La rubrique suivante décrit les principaux éléments du code de cette page.

```
<%@ taglib uri="bloxtld" prefix="blox" %>
//Copiez et collez les informations de balise provenant de la balise blox
//générée sous cette ligne et supprimez tous les sauts de ligne de
//l'instruction de requête. Vous pouvez également supprimer ce commentaire
//après avoir ajouté le code de DB2 Alphablox Query Builder.
<html>
<head>
<blox:header/>
</head>
<body>
<h2>Vue PresentBlox simple</h2>
<p>
<blox:display bloxRef="MyPresentBloxView"/>
</p>
</body>
</html>
```

11. Copiez le fichier enregistré (`PresentBloxView.jsp`) dans le répertoire `MyApp` sur le serveur et testez votre nouvelle vue analytique en cliquant sur le lien menant à cette page, `Vue PresentBlox simple`, à partir de la page d'accueil `MyApp`. Vous pouvez également entrer directement l'URL dans une zone d'adresse de navigateur (par exemple, `http://localhost:9080/MyApp/PresentBloxView.jsp`). La vue qui s'affiche doit être similaire au composant `PresentBlox` manipulé dans `DB2 Alphablox Query Builder`.

Un exemple de code d'une version complète de la page `Vue PresentBlox simple` est présentée ci-dessous.

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

```

Dans le code exemple présenté ci-dessus, l'instruction de requête comporte des sauts de page pour une meilleure lisibilité. Pour que cette page s'exécute correctement, les sauts de ligne des valeurs d'attribut (les attributs query et selectableSlicerDimensions ci-dessus, par exemple) doivent être supprimés. Dans le code généré par DB2 Alphablox Query Builder, un grand nombre de balises présentes ne sont pas nécessaires. Toutefois, grâce à ces balises, vous pouvez facilement comprendre comment fonctionnent les balises blox en tant que balises parent et imbriquées. Vous pouvez également comprendre l'utilisation des attributs de balise ajoutés suite aux manipulations effectuées dans le composant PresentBlox de Query Builder. Pour plus d'informations sur l'utilisation des balises blox, des attributs de balise et sur le mode de développement des applications DB2 Alphablox, consultez le centre de documentation DB2 Alphablox.

Structure de votre première page de vue analytique

Cette rubrique récapitule la structure de la première vue analytique (PresentBloxView.jsp) créée à la rubrique précédente.

Vous trouverez ci-dessous une brève présentation des principaux éléments de la page créée dans une rubrique précédente. Pour une description détaillée du code, reportez-vous aux rubriques s'y rapportant dans le centre de documentation DB2 Alphablox.

La page PresentBloxView.jsp commence par la ligne suivante :

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

Cette ligne est une directive taglib de page JSP qui informe le serveur de votre intention d'utiliser la bibliothèque de balises Blox. L'identificateur uri est un pointeur désignant le fichier descripteur de bibliothèque de balises. La valeur de l'attribut prefix, blox, indique au serveur de rechercher les balises de cette page commençant par blox, puis d'en traiter le contenu à l'aide de la bibliothèque de balises Blox définie dans le fichier descripteur de bibliothèque de balises.

Sous la directive des bibliothèques de balises, le composant PresentBlox est défini à l'aide des balises suivantes et de leurs attributs :

```

<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>

```

La balise `<blox:present>` indique que vous souhaitez qu'un PresentBlox dont la valeur d'attribut d'id est `queryBuilder4_present` soit affiché ici. L'attribut `id` vous permet d'identifier ce Blox particulier en vue d'une utilisation dans des scripts. La balise `<blox:present>` inclut un attribut `visible` ayant la valeur `false`. Dans ce cas, le composant PresentBlox n'est pas représenté sur la page tant que la balise `<blox:display>` n'a pas été rencontrée par le compilateur sur cette page.

Chaque Blox défini à l'aide de balises comporte de nombreux attributs. Toutefois, à moins que vous n'ayez besoin de définir des valeurs d'attribut différentes de celles par défaut, il n'est pas nécessaire de les inclure dans les balises. Dans cet exemple, une valeur `height` de 500 et une valeur `width` de 100%. La taille et la largeur peuvent être exprimées en pixels ou en pourcentages.

Un composant DataBlox imbriqué est inclus. L'attribut `dataSourceName` prend la valeur `qc2003-acs`. Si aucune source de données n'est indiquée, le message "Aucune donnée disponible" apparaît lors de l'affichage de la page. L'attribut `query` inclut l'instruction de requête MDX générée par DB2 Alphablox Query Builder. Une fois que vous maîtrisez le langage de requête MDX utilisé avec le serveur DB2 Alphablox, vous pouvez rendre la requête plus courte, ce qui la simplifie. Cependant, l'instruction de requête présentée ici est complète et va générer la vue créée dans l'application Query Builder.

La section `<head>` de la page comporte une balise Blox spéciale permettant d'ajouter du code important dans la page, avant le rendu de cette dernière :

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


Cette balise est utilisée par DB2 pour ajouter automatiquement le code HTML, JavaScript et CSS requis dans la section d'en-tête de la page. Lors du rendu d'une page par le serveur, cette balise ajoute des liens CSS à des thèmes HTML définis, ainsi qu'une balise "meta" pour empêcher les opérations de mise en mémoire cache. N'oubliez pas d'entrer cette balise dans chaque page JSP qui utilise des composants Blox. Dans le cas contraire, le rendu de la page sera incorrect.

Dans la section <body> de la page se trouve la ligne suivante :

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

Cette balise Blox génère le composant PresentBlox défini ci-dessous et rendu ici. Vous pouvez apporter des modifications à un composant Blox ou aux composants imbriqués à un seul emplacement où vous pouvez également avoir d'autres balises Blox et scriptlets JSP contrôlant le comportement des composants Blox. Dans des fichiers JSP complexes et une fois que vous serez familiarisé avec DB2 Alphablox, vous pouvez choisir d'avoir un composant DataBlox distinct des balises PresentBlox et d'ajouter des scriptlets JSP qui modifient les composants Blox et leur comportement avant que le composant Blox ne soit rendu dans le navigateur Web.

Pour récapituler, les principales balises Blox utilisées sur cette page sont les suivantes : <blox:header/>, <blox:present> et <blox:display>. Ces trois balises et les balises imbriquées spécifient une vue analytique sans qu'il soit nécessaire de recourir à du code Java sur la page. La complexité de la logique de présentation est gérée par ces trois balises Blox. En ajoutant des Blox imbriqués et en modifiant la valeur de leurs attributs, vous pouvez personnaliser les vues en fonction de vos besoins métier.

Pour des applications et des vues plus complexes, des composants JavaBeans et du code Java supplémentaires peuvent être requis.

Création de votre deuxième vue analytique

Dans cette tâche, vous allez créer une vue analytique affichant uniquement un graphique avec des filtres de page, en fonction du code de la première vue analytique de ce tutoriel.

Dans la plupart des applications, les données sont représentées sous la forme de grille ou de graphique. Lorsque le composant PresentBlox associe un élément GridBlox et ChartBlox en tant que composants imbriqués s'affichant simultanément dans PresentBlox, vous pouvez désactiver ou masquer les composants imbriqués qui ne doivent pas être visibles. L'utilisation d'un composant PresentBlox pour la création d'une vue graphique avec des filtres de page est relativement simple. Pour cela, vous pouvez modifier les attributs de certaines balises Blox imbriquées.

Dans cette tâche, vous allez utiliser à nouveau la balise PresentBlox de la première vue analytique afin de créer une deuxième vue analytique personnalisée.

Pour créer une vue analytique affichant un graphique avec des filtres de page, procédez comme suit :

1. Dans l'éditeur, ouvrez le fichier PresentBloxView.jsp précédemment créé et enregistrez-le en lui attribuant le nom CustomizedChartBlox.jsp.
2. Dans les composants imbriqués de la balise PresentBlox présentés ci-dessous, ajoutez un attribut `visible` à chaque balise et attribuez-lui la valeur `false` : L'ajout de cet attribut fait que ces composants imbriqués ne s'affichent pas. Seule une vue de graphique comportant une barre de filtre de page s'affiche.

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

3. Modifiez la balise PageBlox (<blox:page>), en ajoutant la balise labelPlacement et en lui attribuant la valeur top. Par défaut, le libellé des filtres de page s'affiche à la gauche des menus. Le fait d'ajouter cet attribut et d'attribuer la valeur top remplace les comportement par défaut et place les libellés au-dessus des filtres de page.

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

4. Modifiez le nom de l'attribut id pour la balise <blox:present> et attribuez-lui la valeur suivante : chartview. Chaque composant Blox affiché, ou représenté, sur une page doit avoir un ID unique afin d'éviter des erreurs.
5. Enregistrez les modifications et ajoutez le fichier au répertoire MyApp sur le serveur :
6. Ouvrez votre navigateur Web et testez la nouvelle page, en cliquant sur le lien présent sur la page d'accueil ou en y accédant directement à l'aide de l'URL (par exemple, <http://hôtelocal:9080/MyApp/CustomizedChartView.jsp>)

Vous avez maintenant créé deux vues analytiques à l'aide de l'application DB2 Alphablox Query Builder. Vous avez également appris que la modification d'attributs ou l'ajout d'attributs à des balises existantes permet de contrôler l'aspect et le comportement du blox et des composants imbriqués.

Récapitulatif

Si vous avez réalisé toutes les tâches du tutoriel, vous savez comment créer une application DB2 Alphablox élémentaire à l'aide des pages d'administration DB2 Alphablox, de l'application DB2 Alphablox Query Builder et des balises Blox. Si vous avez déjà consulté le manuel *Developer's Reference*, vous savez également qu'il existe un grand nombre de propriétés et de méthodes disponibles pour la définition et la manipulation de Blox sur les pages JSP. Quelles tâches vous reste-t-il à effectuer ?

Si vous souhaitez afficher les données de votre entreprise dans une vue DB2 Alphablox, vous savez désormais créer une application de bout en bout et lui ajouter des vues analytiques. Pour afficher des données sur le champ, il vous suffit d'apporter quelques modifications simples à l'application MyApp. Pour afficher les données de vos bases de données d'entreprise plutôt que d'utiliser la source de données qcc2003-acs, vous pouvez créer des sources de données pointant vers des bases de données d'entreprise, modifier les attributs dataSourceName de telle sorte qu'ils désignent la source de données nouvellement définie puis ajouter l'attribut query approprié. A l'aide de DB2 Alphablox Query Builder, vous pouvez facilement créer des instructions de requête utilisables et des balises blox. Pour plus de détails sur la création de sources de données, reportez-vous au manuel *Administrator's Guide*. Pour savoir comment créer les requêtes appropriées et obtenir plus d'informations sur la création des applications DB2 Alphablox, consultez la section Retrieving Data du manuel *Developer's Guide* et la section DataBlox du manuel *Developer's Reference*.

Chapitre 2. Tutoriel : Création de votre premier portlet contenant des composants Blox

Ce tutoriel vous enseigne comment ajouter des composants Blox à vos portlets. Vous allez notamment apprendre à :

1. Installer un exemple de portlet préconfiguré. Cette procédure vise à vous familiariser rapidement avec la méthode d'ajout d'un composant Blox à une page JSP de portlet et le mode d'affichage dans une page de portail.
2. Ecrivez votre propre page JSP avec un GridBlox.

Ce tutoriel ne vous fournit pas de détails généraux sur le développement de portlets. Ce tutoriel présente les tâches propres à DB2 Alphablox et suppose un certain niveau de connaissances des concepts généraux d'environnement de portail et de développement de portlets.

Prérequis

- DB2 Alphablox doit être installé sur un serveur WebSphere Portal version 5.1. Pour plus d'informations, reportez-vous au *guide d'installation*.
- Votre serveur WebSphere Portal doit être démarré.
- Vous devez disposer de droits d'administration sur votre serveur WebSphere Portal.
- Vous devez bien connaître les fonctions d'administration et l'interface utilisateur de WebSphere Portal.
- Il vous faut également avoir des connaissances de base de Java et des pages JSP.
- Vous devez disposer d'un éditeur JSP.

Vous pouvez utiliser pour ce tutoriel n'importe quel éditeur JSP, voire même un éditeur de texte ; toutefois, lorsque vous développez vos propres portlets, utilisez un outil de développement recommandé par WebSphere Portal, tel que Rational Application Developer.

Ce tutoriel utilise une source de données prédéfinie ayant été installée à l'aide de DB2 Alphablox. Cette source de données peut vous servir à développer rapidement une application de base. Vous n'aurez pas à configurer de source de données personnalisée dans ce tutoriel.

Installation des exemples de portlets

La meilleure manière d'apprendre à ajouter un Blox à votre portlet consiste à installer les exemples de portlet fournis avec DB2 Alphablox, puis à les charger dans une page de portail. Cette opération vous permet d'étudier la structure de base du code JSP et d'établir des correspondances avec la sortie de votre portail.

Installez les exemples de portlets fournis dans DB2 Alphablox :

1. Ouvrez votre navigateur, puis connectez-vous à votre portail en tant qu'administrateur (son adresse URL est semblable à celle-ci : `http://<votreServeurPortail>:<port>/wps/portal`).
2. Cliquez sur le bouton **Administration**.
3. Sous la section de gestion des portlets, cliquez sur l'option des modules Web. La page de gestion des modules Web apparaît sur la droite.

4. Cliquez sur l'option d'installation. Vous êtes invité à préciser le module Web que vous souhaitez installer.
5. Cliquez sur le bouton permettant de parcourir l'arborescence, puis accédez au répertoire `installableApps` situé sous le répertoire d'installation de DB2 Alphablox.
6. Choisissez `AlphabloxSamplePortlets.war`, puis cliquez sur **Suivant**. L'application DB2 Alphablox Sample Portlets contenant un portlet appelé "DB2 Alphablox JSP Page Sample Portlet" s'affiche dans la table d'application de portlets.
7. Cliquez sur **Fin**.

L'application DB2 Alphablox Sample Portlets et les portlets qu'elle contient sont maintenant installés. Vérifiez le répertoire `installedApps` situé sous le répertoire d'installation de WebSphere Portal. Le nom du répertoire nouvellement créé commence par DB2 Alpha et se termine par un ID de portlet généré dynamiquement selon le modèle `_PA_x_x_xx.ear`.

Exécution des exemples de portlets

Pour exécuter les exemples de portlets dans la page de votre portail :

1. Accédez à la page de votre portail.
2. Créez une page ou modifiez une page existante. Vous pouvez créer une page pour tester cet exemple de portlet ou cliquer sur une page existante pour la modifier.
3. Cliquez sur l'un des boutons d'ajout de portlets dans la page d'agencement du portail.
4. Entrez DB2 dans la zone de recherche, puis cliquez sur l'option de recherche. La case à cocher **DB2 Alphablox JSP Page Sample Portlet** apparaît.
5. Sélectionnez cette case à cocher et cliquez sur **OK**.
6. Cliquez sur **Terminé**.

Après régénération de la page du portail, un `PresentBlox` apparaît dans cette page. Il comporte les éléments suivants :

- une barre de menus en haut,
- deux panneaux de barre d'outils sous la barre de menus,
- un panneau de disposition des données situé sur la gauche et permettant de déplacer les dimensions autour des différents axes,
- une grille affichant les données sous forme de tableau,
- un diagramme de Gantt 3D situé sur la droite.

Le fichier `present.jsp` constitue la page par défaut à charger, telle qu'elle est spécifiée dans le servlet `BloxJSPPagePortlet` de l'exemple. Son code source est disponible dans le répertoire `WEB-INF/src/`.

Vous avez installé les exemples de portlets et ajouté un portlet comprenant un `Blox` à la page de votre portail. L'étape suivante consiste à étudier la structure de code de cette page JSP.

Etude de la structure d'une page JSP de portlet contenant un composant Blox

Cette procédure permet de vérifier la structure de code d'une page JSP contenant un `Blox`. Toutes les pages JSP doivent disposer des mêmes éléments-clés.

Pour ouvrir un fichier JSP :

1. Accédez au répertoire `installedApps/` situé sous votre installation de WebSphere Portal, puis repérez le dossier d'application nouvellement créé et commençant par DB2 Alpha.
2. Accédez au répertoire `PA_x_x_xx.war/jsp/html/`.
3. Ouvrez le fichier `present.jsp` dans votre éditeur de pages JSP ou votre éditeur Java.
4. Consultez le code suivant et repérez les éléments-clés :

```
<%@ 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>
```

Ce bloc de code contient six éléments-clés :

1. La première ligne indique au navigateur que la sortie est du code HTML :
2. L'extrait de code suivant spécifie les deux bibliothèques de balises JSP utilisées dans la présente page :

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

L'identificateur `uri` est un pointeur désignant l'emplacement du répertoire de stockage du fichier descripteur de bibliothèque de balises. Les valeurs des deux attributs `prefix`, `blox` et `portletAPI`, indiquent au serveur de :

- rechercher les balises de cette page commençant par `blox`, puis d'en traiter le contenu à l'aide de la bibliothèque de balises Blox définie, comme indiqué dans le fichier descripteur de bibliothèque de balises ;
 - rechercher les balises de cette page commençant par `portletAPI`, puis d'en traiter le contenu à l'aide de la bibliothèque de balises Portlet définie, comme indiqué dans le fichier descripteur de bibliothèque de balises.
3. Ensuite, une balise d'initialisation de portlet est ajoutée :

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

Cette balise permet d'accéder aux objets `PortletRequest`, `PortletResponse` et `PortletConfig`. Avec l'objet `PortletResponse`, vous pouvez appeler la méthode `encodeNamespace()` afin de vous assurer que le nom de votre composant Blox n'entre pas en conflit avec d'autres objets situés sur d'autres portlets qui s'exécutent sur la même page.

4. La balise suivante encode l'espace de nom du Blox à ajouter à la page :

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

Cette opération vous permet de créer un Blox ultérieurement et de lui attribuer ce nom unique.

5. Le bloc de code suivant ajoute la balise d'en-tête Blox nécessaire au rendu des Blox et aux communications client-serveur :

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

Cette balise est utilisée par DB2 Alphablox pour ajouter automatiquement les codes HTML, JavaScript et CSS nécessaires dans la section d'en-tête de la page. Lors du rendu d'une page par le serveur, cette balise entraîne l'insertion de liens CSS dans des thèmes HTML définis et d'une balise "meta" pour empêcher les opérations de mise en mémoire cache. Cette balise doit être ajoutée à chaque page JSP qui contient des composants Blox ; si tel n'est pas le cas, le rendu des composants sera incorrect.

6. Ajoutez un PresentBlox à l'aide des balises de la bibliothèque des balises Blox :

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

- Ce code ajoute un PresentBlox dont l'id est presentBlox et le nom bloxName est xx_x_x_xxx_presentBlox, ce qui correspond au résultat du codage de l'espace de nom.
- La largeur de ce blox est égale à 800 pixels et sa hauteur à 400 pixels (valeur par défaut).
- Un composant DataBlox imbriqué est inclus, l'attribut de balise dataSourceName étant défini sur canned.

L'attribut de balise id du PresentBlox est obligatoire. Il précise le nom de script Java à utiliser dans la page JSP. L'attribut bloxName indique le nom d'objet sur le serveur. La présence d'un bloxName encodé garantit le caractère unique de l'instance sur le serveur.

Remarque : Si aucune source de données n'est spécifiée, le message **Aucune donnée disponible** apparaît généralement dans la grille. La source de données Canned est prédéfinie lors de l'installation et comprend quelques données d'exemple. Elle ne nécessite pas d'installation ni de configuration d'une véritable base de données externe, et peut être utilisée à des fins d'apprentissage et de résolution d'incidents. Etant donné que nous n'avons pas à indiquer de requête, le code ne contient pas d'attribut query.

Cette page ne comporte pas de balise <html> de début/fin car elle est affichée dans une page de portail comportant d'autres portlets. Aucune balise <html> ou <body> supplémentaire n'est nécessaire.

Vous venez d'étudier la structure du code JSP et d'apprendre le code élémentaire à insérer dans vos pages JSP de portlet. La prochaine procédure va vous permettre de créer une nouvelle page JSP contenant un autre Blox et de spécifier certains attributs de Blox communs pour que la page s'adapte mieux à votre page de portail.

Création de votre propre page JSP de portlet contenant des composants Blox

Dans cette section, vous allez créer une page JSP comportant un composant GridBlox et définir certaines de ses propriétés. L'objectif consiste à vous familiariser avec la structure générale des balises de Blox lors de la création d'un GridBlox particulièrement bien adapté à une page de portail normale.

La taille par défaut d'un GridBlox est de 400 pixels par 400. Il comporte également une barre de menus et une barre d'outils. Vous allez définir certaines des propriétés de GridBlox les plus utilisées pour que la hauteur de la grille ne dépasse pas 100 pixels et que sa barre de menus et sa barre d'outils soient désactivées. Pour cela, définissez les propriétés de GridBlox suivantes :

- `height` : paramétrez cette propriété à 100 pixels.
- `menubarVisible` : paramétrez cette propriété à `false`.
- `toolbarVisible` : paramétrez cette propriété à `false`.

Assurez-vous que les conditions prérequis de la section Chapitre 2, «Tutoriel : Création de votre premier portlet contenant des composants Blox», à la page 11 sont remplies et que vous avez installé la collection d'exemples de portlets conformément aux instructions de la section «Installation des exemples de portlets», à la page 11.

Suivez la procédure ci-dessous :

1. Dans la fenêtre du navigateur dans laquelle le PresentBlox est affiché, cliquez sur le bouton de modification du portlet (celui avec l'icône d'un crayon). Une liste déroulante de sélection apparaît.
2. Dans cette liste déroulante, sélectionnez "Grid Blox" puis cliquez sur **OK**.
Après actualisation de la page, un GridBlox de 400 pixels par 400 est visible. Par défaut, sa barre de menus et sa barre d'outils sont activées. Vous allez faire passer sa taille à 400 pixels par 100 et désactiver sa barre de menus et sa barre d'outils.
3. Accédez au répertoire `PA_x_x_xx.war/jsp/html/` situé sous l'application DB2 Alphablox Sample Portlets que vous avez précédemment installée.
4. Ouvrez le fichier `grid.jsp` dans votre éditeur de pages JSP. Cette page est quasiment identique à la page `present.jsp` aux quelques détails près suivants :
 - La balise `<blox:present>` est maintenant remplacée par la balise `<blox:grid>` et la valeur de son id diffère :

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400">
<blox:data dataSourceName="canned" />
</blox:grid>
```
 - La valeur du `bloxName` diffère :

```
<%
String bloxName = portletResponse.encodeNamespace("gridBlox");
%>
```
5. Paramétrez la hauteur de ce GridBlox à 100 pixels en ajoutant l'attribut `height` et en lui attribuant la valeur 100 :

```
<blox:grid id="gridBlox" bloxName="<%= bloxName %>" width="400" height="100">
<blox:data dataSourceName="canned" />
</blox:grid>
```
6. Désactivez la barre de menus et la barre d'outils situées au sommet en paramétrant les attributs `menubarVisible` et `toolbarVisible` à `false` :


```

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

```

Vérifiez que vous avez correctement saisi les noms d'attributs, en particulier la lettre majuscule "V" figurant dans ces deux noms). Vérifiez également que les attributs ont été ajoutés avant le crochet angulaire de fin (">").

7. Modifiez l'espace de nom en myFirstGrid :

```

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

```

La modification de l'espace de nom du GridBlox permet de s'assurer que les modifications que vous venez d'effectuer seront visibles lors du chargement de cette page JSP dans votre portail. Comme vous l'avez précédemment chargée, une instance de ce GridBlox est déjà en cours d'exécution sur le serveur pour cette session. Si vous ne modifiez pas l'espace de nom, les modifications effectuées ne seront pas visibles, même si vous actualisez la page. La modification de l'espace de nom constitue une méthode rapide pour le test des modifications dans un environnement de développement. Une autre solution consiste à ouvrir une nouvelle fenêtre de navigateur afin de créer un nouvel objet sur le serveur pour cette nouvelle session.

8. Enregistrez le fichier.

Vous êtes désormais prêt à tester ce fichier sur WebSphere Portal.

Pour tester les modifications que vous venez d'effectuer :

1. Revenez à la page du portail.
2. Cliquez sur le bouton d'actualisation du navigateur pour recharger la page.

Un GridBlox de 400 pixels par 100 sans barre de menus, ni barre d'outils est normalement visible.

Remarque : La liste déroulante de sélection JSP n'est pas une fonctionnalité intégrée de WebSphere Portal. Elle est créée par cet exemple de portlet. Vérifiez le fichier edit.jsp et les fichiers source Java du répertoire WEB-INF/src/.

Création d'un projet de type portlet afin d'utiliser des composants Blox

Lors de la création de votre projet de type portlet dans votre outil de développement, assurez-vous que le mappage de servlet AlphabloxServer et les références de bibliothèque de balises nécessaires aux composants Blox sont ajoutés au fichier web.xml de votre projet et que les fichiers .tld des bibliothèques de balises de DB2 Alphablox sont copiés dans votre projet.

1. Modifiez le fichier web.xml de votre projet pour qu'il comprenne les lignes suivantes :

- Pour la définition et le mappage de servlets :

```

<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>

```

- Pour les références de bibliothèque de balises :

```

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

```

2. Copiez les fichiers .tld des bibliothèques de balises de DB2 Alphablox dans le répertoire WEB-INF/tlds/ de votre projet. Ces fichiers sont situés dans le répertoire

```
<rép_db2alphablox>/bin/
```

où <rép_db2alphablox> correspond au répertoire d'installation de DB2 Alphablox.

Configuration d'un projet de type portlet à l'aide de Rational Application Developer

Avec WebSphere Portal, nous vous recommandons d'utiliser Rational Application Developer pour le développement de portlets. En particulier, Rational Application Developer version 6.0 est requis pour WebSphere Portal version 5.1. Rational Application Developer intègre un assistant d'aide à la configuration et à la création d'un projet de type portlet. En fonction de votre sélection, il configure la structure et le fichier descripteur de déploiement appropriés, puis crée automatiquement les classes Java nécessaires au contrôleur et aux pages JSP de votre vue de portlet. Au fur et à mesure du déroulement de l'assistant de nouveau projet de portlet, assurez-vous que les éléments suivants sont correctement configurés :

- Choisissez le "module de remplacement WebSphere Portal v5.1" comme serveur cible. Cette option est définie sur le premier écran de l'assistant en cliquant sur le bouton **Afficher les options avancées >>**.
- Ajoutez le mappage de servlet de DB2 Alphablox au fichier web.xml créé sous WebContent/WEB-INF/. Pour cela, saisissez la définition de servlet et le code de mappage suivants dans le fichier web.xml de votre projet :

```

<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>

```

- Ajoutez les bibliothèques de balises Alphablox au fichier web.xml de votre projet. Suivez la procédure ci-dessous :
 1. Une fois le fichier web.xml ouvert, cliquez sur l'onglet **Variables**.
 2. Accédez à la section consacrée aux références des bibliothèques de balises et située au bas de l'onglet.
 3. Cliquez sur **Ajouter**.
 4. Pour les zones **URL** et **Location (Emplacement)**, saisissez les valeurs en vous aidant du tableau suivant :

URL	Emplacement
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

Pour chaque paire URL-Emplacement, cliquez sur **Fin** pour ajouter la référence de bibliothèque de balises, puis sur **Ajouter** pour ajouter la paire suivante.

Remarque : Pour plus de détails sur les différentes bibliothèques de balises et leur utilisation, reportez-vous à la section Using JavaServer Pages and Blox.

Enfin, n'oubliez pas de copier les fichiers .tld des bibliothèques de balises de DB2 Alphablox dans le répertoire WEB-INF/tlds/ de votre projet. Ces fichiers sont situés dans le répertoire

```
<rép_db2alphablox>/bin/
```

où <rép_db2alphablox> correspond au répertoire d'installation de DB2 Alphablox.

Etapes suivantes

Vous avez appris à créer un portlet avec des composants Blox, ainsi que la structure fondamentale du code. Pour afficher des données de vos bases de données, vous devez :

1. définir une nouvelle source de données DB2 Alphablox pointant vers votre source de données ;
2. modifier l'attribut dataSourceName de DataBlox pour le faire pointer vers la source de données définie ;
3. ajouter un attribut query avec votre chaîne de requête au DataBlox.

Pour plus de détails sur la création de sources de données, reportez-vous au manuel *Administrator's Guide*. Pour savoir comment créer les requêtes appropriées, consultez la section Retrieving Data du *Developer's Guide* et la section DataBlox du *Developer's Reference*.

Une fois que vos données s'affichent dans les vues Blox, vous pouvez également commencer à étudier les nombreuses propriétés définissables à l'aide des balises Blox.

Conseils relatifs au développement de portlets

Lorsque vous consultez les informations figurant dans la documentation DB2 Alphablox et que vous poursuivez simultanément le développement de votre portlet, vous devez tenir compte d'un certain nombre d'instructions générales, de notions et de sujets relatifs aux portlets. La liste ci-dessous regroupe ces instructions générales de développement, ainsi que des renvois intéressants vers des sections particulières de la documentation.

- Définissez toujours le nom de votre Blox à l'aide de l'espace de nom du portlet. Un espace de nom garantit le caractère unique d'un nom de Blox pour la session J2EE en cours.
- Définissez toujours la largeur et la hauteur d'un Blox en pixels. En dehors de l'environnement du portail, vous pouvez les exprimer en pourcentages, "50%" ou "100%" par exemple. Les valeurs de pourcentage ne fonctionnent pas dans l'environnement de portail en raison de la coexistence de plusieurs portlets sur la page.
- N'utilisez pas d'URL relatives pour appeler des ressources au sein d'un portlet. Les URL doivent être encodées à l'aide de la méthode `encodeURL()` de la classe `PortletResponse`. Lorsque des passages de cette documentation font référence à l'utilisation d'adresses URL relatives, veillez à toujours encoder ces dernières.
- Pour vous assurer que votre portlet Blox utilise un thème semblable à celui de votre portail, servez-vous de l'utilitaire de thèmes de portail. Cet utilitaire est disponible à partir de deux sources :
 - La page d'accueil de DB2 Alphablox.
Dans l'onglet **Administration**, cliquez sur le lien **General**. Cet utilitaire figure sous la section **Portal**. Reportez-vous au manuel *Administrator's Guide* et à l'aide en ligne pour plus de détails.
 - Le fichier `AlphabloxAdminPortlets.war`.
Il s'agit d'une version de type portlet de l'utilitaire de thèmes de portail. Vous pouvez installer ce portlet de manière à l'exécuter via votre portail sans avoir à établir une connexion distincte avec la page d'accueil de DB2 Alphablox.

Cet utilitaire combine les styles de votre environnement de portail avec ceux de DB2 Alphablox de manière à ce que les Blox s'affichent dans des couleurs et des polices semblables à celles des autres portlets de la page.

- Consultez la rubrique **Blox Portlet Tag Library** du *Developer's Guide* pour connaître les problèmes dont vous devrez tenir compte lors de la phase de planification.
- DB2 Alphablox possède un puissant modèle d'identification unique des Blox. Ce modèle comprend un objet `ClientLink` qui permet de charger une URL spécifiée lorsque l'utilisateur clique sur un composant Blox de la page. Pour le développement de portlets, utilisez la bibliothèque de balises de portlets de Blox pour la création de ce lien. Ces balises gèrent dynamiquement la formation des adresses URL de manière à ce que les liens soient mis à jour lors de l'actualisation de la page. Pour plus de détails, reportez-vous à la rubrique **Planning for portlet development** du *Developer's Guide*.
- Pour les identificateurs URI des actions et des portlets, utilisez la bibliothèque de balises de portlets de Blox pour la création de liens d'action ou de portlet. Vous

pouvez ensuite recourir à l'API de portlet du portail pour le traitement du lien de portlet ou d'action. Pour plus de détails, reportez-vous à la rubrique Blox Portlet Tag Library du *Developer's Guide*.

Chapitre 3. Tutoriel : Création d'applications à l'aide des outils Rational Developer

Vous pouvez utiliser les outils Rational Developer en combinaison avec DB2 Alphablox Toolkit, et développer ainsi plus rapidement des applications utilisant des composants DB2 Alphablox et des API DB2 Alphablox Java.

Les outils Rational Developer (Rational Application Developer ou Rational Web Developer) peuvent être utilisés en combinaison avec DB2 Alphablox Toolkit, un ensemble de plug-ins (modules d'extension) Eclipse, pour le développement et le test d'applications créées à l'aide de la technologie DB2 Alphablox. Ce tutoriel va vous guider tout au long de la configuration de votre environnement de développement intégré Rational, qui permet l'utilisation de méthodes et de balises, ainsi que des améliorations personnalisées ajoutées par DB2 Alphablox Toolkit.

Commencez la configuration de votre outil Rational Developer avec DB2 Alphablox Toolkit, en vue de développer et de tester des applications DB2 Alphablox.

Préparation de votre environnement de développement

Avant de pouvoir installer DB2 Alphablox Toolkit et de commencer à l'utiliser, préparez votre environnement de développement en vous assurant de disposer de tous les logiciels requis.

Pour préparer votre environnement de développement :

1. Installez un outil Rational Developer (Rational Application Developer ou Rational Web Developer) sur votre poste de travail.
2. Installez toutes les mises à jour requises afin de vous assurer que vous disposez au moins de Rational Application Developer ou de Rational Web Developer version 6.0.0.1.
3. Si vous envisagez d'utiliser les environnements de test intégrés WebSphere 5.1, assurez-vous d'avoir mis à jour votre version de Rational Developer de manière à ce qu'elle intègre cette installation facultative.

Remarque : Les installations par défaut des outils de Rational Developer n'incluent pas les environnements de test intégrés WebSphere 5.1.

Vous êtes maintenant prêt pour effectuer les tâches de ce tutoriel.

Installation du DB2 Alphablox Toolkit

L'utilisation de DB2 Alphablox Toolkit en combinaison avec Rational Developer facilite le développement d'applications Web comportant du contenu DB2 Alphablox.

DB2 Alphablox Toolkit nécessite l'utilisation de Rational Application Developer ou de Rational Web Developer version 6.0.0.1.

L'installation de DB2 Alphablox Toolkit dans un outil Rational Developer permet de supprimer les nombreuses tâches manuelles nécessaires à l'utilisation de la fonction de fourniture de contenu pour les méthodes Java et les balises

personnalisées JSP. En outre, les assistants de personnalisation et les aide-mémoire peuvent vous aider pour la création d'applications comportant du contenu DB2 Alphablox ainsi que pour la définition des instances du serveur WebSphere.

Pour installer DB2 Alphablox Toolkit dans Rational Application Developer ou Rational Web Developer :

1. Insérez le CD d'installation de DB2 Alphablox dans le lecteur de CD de votre poste de travail.
2. Dans le répertoire plugin, repérez le sous-répertoire nommé UpdateSite.
3. Copiez le répertoire UpdateSite à un emplacement adéquat de votre disque dur. Par exemple, copiez-le dans C:\DB2Alphablox\UpdateSite.
4. Démarrez l'outil Rational Developer.
5. Dans la barre de menus, sélectionnez **Aide** puis l'**option de mises à jour des logiciels** et enfin l'**option de recherche et d'installation**.
6. Dans la fenêtre d'installation qui s'affiche, sélectionnez l'option de recherche des nouvelles fonctions à installer puis cliquez sur **Suivant**.
7. Dans la fenêtre de mise à jour des sites à visiter, cliquez sur le bouton du nouveau site local puis accédez à l'emplacement du répertoire UpdateSite.
8. Cliquez sur **Suivant**, sélectionnez la fonction **DB2 Alphablox Toolkit** puis cliquez une nouvelle fois sur **Suivant**.
9. Dans les fenêtres suivantes, acceptez l'accord de licence puis sélectionnez l'emplacement d'installation de la fonctionnalité DB2 Alphablox Toolkit.
10. Une fois l'installation terminée, redémarrez l'outil Rational Developer.

Les fonctionnalités de DB2 Alphablox Toolkit seront disponibles après le redémarrage de l'outil Rational Developer.

Installation de DB2 Alphablox dans les environnements de test intégrés de WebSphere

Cette procédure vous permet d'installer DB2 Alphablox dans les environnements de test intégrés de WebSphere situés dans un outil Rational Developer.

Conditions préalables : l'environnement de test intégré de WebSphere doit être installé dans un outil Rational Developer. Vous devez accéder au CD d'installation de DB2 Alphablox. Prenez connaissance de toutes les étapes de pré-installation requises décrites dans la section relative aux tâches de pré-installation du guide d'installation *DB2 Alphablox Installation Guide* et effectuez-les.

DB2 Alphablox peut être installé sur des serveurs d'applications WebSphere autonomes ou dans les environnements de test intégrés de WebSphere disponibles dans l'outil Rational Developer. Le récapitulatif de la procédure présentée ci-dessous explique les différences de manipulation à effectuer pour l'installation de DB2 Alphablox dans les environnements d'exécution de WebSphere disponibles dans l'outil Rational Developer.

Pour installer DB2 Alphablox dans les environnements de test intégrés de WebSphere :

Installez DB2 Alphablox en suivant les étapes décrites dans la section d'installation du guide *DB2 Alphablox Installation Guide*. Tenez compte des exceptions suivantes :

1. Dans la fenêtre de configuration de WebSphere du programme d'installation, saisissez dans la zone **Répertoire racine WebSphere**, l'environnement d'exécution de WebSphere à utiliser comme serveur d'environnement de test intégré.

Par exemple, pour installer DB2 Alphablox dans l'environnement de test intégré de WebSphere 6 sur l'installation par défaut, sélectionnez le chemin d'accès au répertoire base_v6. Pour une installation standard de Rational Application Developer, ce chemin est semblable à ce qui suit :

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

2. Dans la fenêtre de paramétrage de WebSphere, indiquez un nom et un mot de passe pour l'administrateur WebSphere. Les valeurs saisies ne sont pas utilisées dans l'environnement de test intégré de WebSphere, mais elles sont requises par le programme d'installation de DB2 Alphablox.

Important : N'effectuez pas la procédure de post-installation décrite dans le manuel *DB2 Alphablox Installation Guide*.

Après avoir installé DB2 Alphablox dans l'environnement d'exécution de WebSphere, vous pouvez créer des instances de serveur WebSphere afin de tester les applications DB2 Alphablox dans l'outil Rational Developer.

Création d'une instance de serveur WebSphere

Pour exécuter des applications ou des fichiers JSP comportant du contenu DB2 Alphablox, vous devez créer une instance de serveur WebSphere capable d'accéder aux services DB2 Alphablox et aux classes Java requis.

Conditions préalables : configurez l'outil Rational Developer. Installez DB2 Alphablox Toolkit. Installez DB2 Alphablox dans les environnements de test intégrés de WebSphere.

Pour créer une instance de serveur WebSphere afin de tester les applications et les fichiers JSP comportant du contenu DB2 Alphablox :

1. Ouvrez l'outil Rational Developer.
2. Cliquez sur l'onglet **Serveurs** (s'il n'est pas visible, sélectionnez **Fenêtre** puis **l'option d'affichage de la vue** et enfin **Serveurs**).
3. Cliquez à l'aide du bouton droit de la souris dans la fenêtre de la vue et sélectionnez **Nouveau > Serveur**. La fenêtre du nouveau serveur apparaît.
4. Entrez `hôte\local` comme nom d'hôte et sélectionnez le type de serveur. Cliquez sur **Suivant**.
5. Saisissez le numéro de port du serveur puis cliquez sur **Suivant**.
6. Ajoutez à la liste du projet configuré les projets disponibles que vous voulez exécuter dans cette instance de serveur. Cliquez sur **Fin**. La nouvelle instance de serveur apparaît dans la vue **Serveurs**.

Si vous avez créé une instance de serveur WebSphere 5.1, vous devez effectuer l'étape suivante, "Création de variables de substitution de serveur WebSphere 5.1.". Pour les instances de serveur WebSphere 6, passez à l'étape "Ajout de l'utilisateur invité au groupe des administrateurs DB2 Alphablox."

Création de variables de substitution de serveur WebSphere 5.1

Dans cette procédure, vous allez modifier les variables de substitution de serveur pour les instances du serveur WebSphere 5.1 afin d'exécuter correctement DB2 Alphablox dans les environnements de test intégrés WebSphere.

1. Ouvrez la vue **Serveurs** puis cliquez deux fois sur l'instance de serveur WebSphere 5.1 à modifier.
2. Cliquez sur l'onglet **Configuration** de l'instance de serveur.
3. Sélectionnez l'option d'activation de la console d'administration puis désélectionnez l'option d'activation du client de test universel.
4. Cliquez sur l'onglet **Variables** de l'instance du serveur. La fenêtre **Substitution Variables** apparaît.
5. Ajoutez deux nouvelles variables à l'aide du bouton **Ajouter** (situé en regard de la liste des paramètres de noeud).
 - a. Ajoutez une variable nommée `WS_EAR_AlphabloxPlatform` puis paramétrez sa valeur à `$(RACINE_INSTALL_APP)/hôtelocal/AlphabloxPlatform.ear`.
 - b. Ajoutez une seconde variable nommée `WS_EAR_AlphabloxStudio` puis paramétrez sa valeur à `$(RACINE_INSTALL_APP)/hôtelocal/ApplicationStudio.ear`
6. Enregistrez vos modifications.

Pour les instances de serveur WebSphere 5.1, vous devez suivre la procédure de la section "Configuration d'instances de serveur WebSphere 5.1" pour mener à terme la modification de l'instance de serveur.

Configuration d'instances de serveur WebSphere 5.1

Vous devez configurer les instances de serveur WebSphere 5.1 pour l'exécution des fichiers et des applications DB2 Alphablox.

Conditions préalables : créez les variables de substitution de serveur WebSphere 5.1 nécessaires.

Pour configurer les instances de serveur WebSphere 5.1 :

1. Ouvrez la vue **Serveurs** dans l'outil Rational Developer puis démarrez l'instance de serveur à configurer.
2. Cliquez avec le bouton droit de la souris sur l'instance de serveur puis sélectionnez l'option d'exécution de la console d'administration.
3. N'entrez pas de valeur dans la zone **ID**. Cliquez sur **OK**.
4. Cliquez sur le bouton **OK** situé sous la table de configuration.
5. Ouvrez la vue de navigation **Applications d'entreprise** puis cliquez sur le nom d'application **ApplicationStudio**.
6. Attribuez la valeur `$(WS_EAR_ApplicationStudio)` aux fichiers binaires des applications.
7. Sélectionnez l'option d'utilisation des métadonnées des fichiers binaires.
8. Cliquez sur le bouton **OK** situé sous la table de configuration.
9. Cliquez sur le bouton **Enregistrer** de la console d'administration WebSphere.
10. Dans la fenêtre de sauvegarde dans la configuration maîtresse, cliquez sur **Enregistrer**.

11. Démarrez les applications AlphabloxPlatform et ApplicationStudio dans la console d'administration de WebSphere sous **Applications > Applications d'entreprise**.

L'instance de serveur est configurée. Vous devez ensuite suivre la procédure "Ajout de l'utilisateur invité au groupe des administrateurs de DB2 Alphablox".

Ajout de l'utilisateur invité au groupe des administrateurs de DB2 Alphablox

Si vous ajoutez l'utilisateur invité au groupe des administrateurs de DB2 Alphablox, vous pouvez accéder aux pages d'administration de DB2 Alphablox lors de l'utilisation de Rational Developer.

Conditions préalables : créez l'instance de serveur WebSphere. Configurez l'instance de serveur WebSphere.

Pour ajouter l'utilisateur invité au groupe des administrateurs DB2 Alphablox :

Important : N'accordez pas à l'utilisateur invité de droits d'administration sur vos serveurs de production WebSphere.

1. Connectez-vous à la console DB2 Alphablox à l'aide de la commande telnet suivante : `telnet localhost numéroPort` . où *Numéro_port* correspond au port spécifié lors de l'installation de DB2 Alphablox.
2. A l'invite de la console Telnet, entrez la commande DB2 Alphablox suivante : `set Administrators guest` puis appuyez sur Entrée.
3. Dans la console Telnet, entrez `save` puis appuyez sur Entrée.
4. Fermez la session Telnet.

Votre instance de serveur WebSphere est désormais prête à l'emploi.

Création d'applications DB2 Alphablox

Lorsque vous créez des applications dans Rational Developer, vous devez ajouter le contenu DB2 Alphablox nécessaire à leur bon fonctionnement.

Conditions préalables : installez DB2 Alphablox Toolkit sur l'outil Rational Developer.

1. Dans la barre de menus de l'outil Rational Developer, sélectionnez **Fichier > Nouveau projet**. L'assistant **Nouveau projet** s'ouvre.
2. Développez l'option **Web**, sélectionnez l'option **Projet Web dynamique**, puis cliquez sur **Suivant**.
3. Saisissez le nom de votre projet puis cliquez sur le bouton **Afficher les options avancées**. Des options supplémentaires apparaissent.
4. Sélectionnez le serveur cible et la version de servlet appropriés.
5. Cliquez sur **Suivant**. La fenêtre des options des fonctions apparaît.
6. Sélectionnez l'option correspondant au contenu DB2 Alphablox puis cliquez sur **Fin**.

Votre nouveau projet d'application Web est maintenant prêt pour DB2 Alphablox. Les bibliothèques de balises de DB2 Alphablox et les API Java de Blox sont disponibles. En outre, le fichier descripteur de l'application (web.xml) a été modifié

pour l'inclusion des informations DB2 Alphablox requises. Vous pouvez maintenant commencer à ajouter des fichiers JSP comportant du contenu DB2 Alphablox à votre projet.

Création de fichiers JSP avec du contenu DB2 Alphablox

Dans cette procédure, vous allez créer des fichiers JSP permettant d'accéder aux bibliothèques de balises Blox, à l'aide de l'outil Rational Developer et de DB2 Alphablox Toolkit.

Conditions préalables : installez DB2 Alphablox Toolkit sur l'outil Rational Developer.

1. Dans la vue de l'explorateur de projets de Rational Developer, sélectionnez **Fichier > Nouveau > Fichier JSP**.
2. Entrez un nom de fichier dans la zone **Nom de fichier**.
3. Cliquez sur l'option permettant de configurer les options avancées puis sur **Suivant**.
4. Ajoutez les bibliothèques de balises DB2 Alphablox à utiliser dans votre fichier JSP.
 - a. Cliquez sur le bouton **Ajouter** pour ouvrir la fenêtre d'ajout de bibliothèques de balises.
 - b. Sélectionnez les bibliothèques de balises DB2 Alphablox que vous envisagez d'utiliser dans votre fichier JSP.
 - c. Cliquez sur **Suivant**.
5. Sélectionnez **ISO 10646/Unicode(UTF-8)** dans la liste **Codage**. Pour que les applications DB2 Alphablox s'exécutent correctement, vous devez utiliser le codage UTF-8.
6. Cliquez sur **Fin**. Le nouveau fichier JSP apparaît dans la liste du projet.
7. Cliquez deux fois sur son nom pour l'ouvrir dans la fenêtre de l'éditeur de pages JSP.
8. Placez le curseur à l'intérieur de la balise HTML <head>, en veillant, cependant, à le placer à la suite des directives taglib JSP de DB2 Alphablox que vous avez ajoutées précédemment.
9. Saisissez la balise d'en-tête Blox en entrant le code suivant sur la nouvelle ligne : <blox:header/> Cette balise est nécessaire pour l'ajout de fichiers CSS et JavaScript de DB2 Alphablox lors de l'exécution du fichier JSP.

Le nouveau fichier JSP est désormais prêt à accéder aux bibliothèques de balises de DB2 Alphablox sélectionnées. Grâce à la fonction Content Assist de Rational Developer, vous pouvez insérer des balises Blox et des attributs de balise pour les bibliothèques de balises sélectionnées.

Accès aux sources de données DB2 OLAP Server et Essbase

Pour pouvoir accéder aux sources de données DB2 OLAP Server ou Essbase dans l'outil Rational Developer lors de l'utilisation d'instances de serveur WebSphere 5.1, vous devez créer un fichier de commandes de démarrage permettant de charger les bibliothèques client nécessaires.

Conditions préalables : Installez DB2 Alphablox Toolkit. Installez DB2 Alphablox dans les environnements de test intégrés WebSphere 5.1. Configurez les instances du serveur WebSphere 5.1. Installez IBM DB2 OLAP Server ou Hyperion Essbase sur votre poste de développement.

En raison des limites inhérentes aux environnements de test intégrés WebSphere 5.1 de l'outil Rational Developer, vous devez créer un fichier de commandes de démarrage pour que Rational Developer puisse accéder aux sources de données DB2 OLAP Server ou Hyperion Essbase. Lors du démarrage de l'outil Rational Developer à l'aide du fichier de commandes, les bibliothèques client Essbase requises sont ajoutées dans l'emplacement de bibliothèque Java.

Pour créer un fichier de commandes permettant d'accéder à DB2 OLAP Server ou Essbase via les instances de serveur WebSphere 5.1 :

1. A l'aide d'un éditeur de texte, créez un document.
2. Ajoutez une ligne de code pour appeler le fichier aassetup.bat de DB2 Alphablox, situé dans votre répertoire d'installation DB2 Alphablox. Par exemple, le code suivant exécutera le fichier aassetup.bat situé dans le répertoire d'installation DB2 Alphablox :

```
call C:\alphablox\analytics\bin\aassetup.bat
```
3. Ajoutez une deuxième ligne afin d'appeler le fichier rationalsdp.exe RAD qui démarrera RAD. Par exemple, le code suivant exécutera l'outil Rational Developer :

```
call C:\Program Files\IBM\Rational\SDP\6.0\rationalsdp.exe
```
4. Enregistrez ce fichier sous le nom startRAD.bat sur le bureau de votre poste de travail (ou à un autre emplacement, selon votre convenance).

Lorsque vous cliquez deux fois sur le fichier startRAD.bat, le fichier aassetup.bat de DB2 Alphablox s'exécute et définit les chemins d'accès et les variables d'environnement nécessaires ; Rational Application Developer est ensuite lancé. La version des bibliothèques client Essbase utilisées avec DB2 Alphablox doit correspondre à celle du serveur DB2 OLAP Server (ou d'Hyperion Essbase) utilisé comme source de données. Pour modifier les bibliothèques Essbase utilisées avec DB2 Alphablox, exécutez l'utilitaire DB2 OLAP Server / Essbase Client Library (ChangeEssbase.bat), situé dans le répertoire *db2_alphablox\analytics\bin*, où *db2_alphablox* correspond au répertoire racine de l'installation de DB2 Alphablox. Cet utilitaire modifie l'un des fichiers de commandes exécutés lorsque vous démarrez RAD à l'aide du fichier de commandes startRAD que vous avez créé.

Retrouvez ci-dessous un exemple complet de fichier startRAD.bat (tel que décrit ci-dessus) :

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

A chaque fois que vous essayez d'accéder aux sources de données DB2 OLAP Server ou Essbase, vous devez utiliser le fichier startupRAD.bat pour lancer l'outil Rational Developer.

Chapitre 4. Tutoriel : Création d'un cube DB2 Alphablox à l'aide de DB2 Cube Views

Ce tutoriel vous guide tout au long de la création d'un cube DB2 Alphablox à l'aide d'une base de données d'exemple DB2 Cube Views.

Aucun détail concernant la création de cubes DB2 Alphablox personnalisés n'est fourni ici. Ce tutoriel s'attache à vous expliquer comment créer rapidement un cube DB2 Alphablox qui peut être utilisé pour explorer les fonctions du serveur de cubes DB2 Alphablox. Les sources de données qui en résultent peuvent également servir à tester et à créer des applications DB2 Alphablox.

Conditions préalables :

- Installez DB2 Alphablox. Pour plus d'informations, reportez-vous au *guide d'installation*.
- Obtenez des droits d'accès vers une implémentation DB2 Cubes Views prise en charge sur laquelle est installée la base de données d'exemple DB2 Cube Views CVSAMPLE. Pour plus de détails sur les versions de DB2 Cube Views prises en charge, reportez-vous au *guide d'installation*.

Dans ce tutoriel, vous apprendrez à créer un cube DB2 Alphablox reposant sur la base de données d'exemple DB2 Cube Views, CVSAMPLE. Vous saurez ainsi effectuer les tâches suivantes :

Définition d'une source de données relationnelles DB2

Cette procédure permet d'établir une définition de source de données dans DB2 Alphablox pour une base de données DB2.

Conditions préalables : les pilotes JDBC DB2 doivent être accessibles dans DB2 Alphablox.

Un cube DB2 Alphablox nécessite que la source de données relationnelles soit prédéfinie en tant que source de données DB2 Alphablox. Les cubes DB2 Alphablox sont générés à l'aide des métadonnées et des données disponibles dans les bases de données relationnelles.

Pour définir une source de données DB2 Alphablox pour une base de données DB2 :

1. Connectez-vous aux pages d'administration de DB2 Alphablox en tant qu'administrateur (ou utilisateur disposant de droits d'administration).
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Sources de données**.
4. Cliquez sur le bouton **Créer**.

5. Dans le menu **Carte**, sélectionnez le pilote JDBC IBM DB2 approprié pour votre serveur de base de données.

Sélectionnez **IBM DB2 JDBC Type 4 Driver** ou **IBM DB2 UDB on iSeries Driver**.

6. Dans la zone du nom de la source de données, entrez CVSAMPLE comme nom à utiliser pour la source de données.

7. Entrez les valeurs appropriées pour le nom du serveur, le numéro du port et le nom de la base de données (en l'occurrence CVSAMPLE).

Remarque : Si vous avez besoin d'aide pour déterminer les valeurs correctes à saisir dans ces zones, contactez votre administrateur de base de données.

8. Entrez un nom d'utilisateur par défaut et un mot de passe par défaut .
Ils doivent être valides dans la base de données relationnelles. Le nom d'utilisateur et le mot de passe par défaut sont toujours utilisés lorsqu'un cube DB2 Alphablox accède à une base de données relationnelles. L'utilisateur de base de données utilisé doit posséder les droits d'accès en lecture à la base de données.

Remarque : La valeur de la zone concernant l'utilisation du nom d'utilisateur et du mot de passe DB2 Alphablox est ignorée lorsque la source de données relationnelles spécifiée est utilisée pour le remplissage d'un cube DB2 Alphablox. Vous pouvez utiliser les listes de contrôle d'accès pour restreindre l'accès aux cubes DB2 Alphablox. Pour plus d'informations sur les listes de contrôle d'accès, reportez-vous au manuel *Administrator's Guide*.

9. Les valeurs **Nombre maximal de lignes** et **Nombre maximal de colonnes** sont ignorées lorsque la source de données est utilisée pour le remplissage d'un cube DB2 Alphablox. Vous pouvez toujours entrer des valeurs. Néanmoins, elles seront utilisées lorsque d'autres applications utilisent la source de données.
10. Attribuez la valeur **Non** à l'option indiquant que le traçage JDBC est activé, sauf si vous voulez consigner les informations de journalisation dans le fichier journal de DB2 Alphablox. Activez le traçage JDBC uniquement si des incidents surviennent et que vous devez effectuer des opérations de débogage.
11. Cliquez sur le bouton **Enregistrer** pour enregistrer la définition de la source de données.

Vous venez de définir la source de données DB2 Alphablox pour CVSAMPLE. Vous pouvez maintenant créer une définition de cube DB2 Alphablox pour l'accès aux métadonnées des cubes Cube Views de cette source de données DB2.

Définition d'une source de données d'adaptateur de serveur de cube Alphablox

Dans cette section, vous allez établir une définition de source de données DB2 Alphablox utilisant l'adaptateur de serveur de cube Alphablox.

Conditions préalables : créez la définition de source de données DB2 Alphablox pour votre base de données CVSAMPLE DB2.

Pour définir une source de données d'adaptateur de serveur de cube DB2 Alphablox :

1. Connectez-vous aux pages d'administration de DB2 Alphablox en tant qu'administrateur (ou utilisateur disposant de droits d'administration).
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Sources de données**.
4. Cliquez sur le bouton **Créer**.

5. Dans le menu **Carte**, sélectionnez l'option de la carte serveur du cube Alphablox.
6. Dans la zone du nom de la source de données, entrez DB2AlphabloxCubes comme nom à utiliser pour la source de données.
7. Cliquez sur le bouton **Enregistrer** pour enregistrer la définition de la source de données.

Vous venez de définir une source de données DB2 Alphablox qui peut être utilisée pour accéder aux cubes DB2 Alphablox. Il vous faut ensuite définir un cube DB2 Alphablox auquel vous pouvez accéder.

Définition d'un cube DB2 Alphablox

Dans cette section, vous allez définir un cube DB2 Alphablox sur la base des métadonnées d'un cube CVSAMPLE DB2 Cube Views.

Conditions préalables : définissez la source de données relationnelles DB2. Définissez une source de données d'adaptateur de serveur de cube DB2 Alphablox.

Pour définir les propriétés générales d'un cube DB2 Alphablox :

1. Connectez-vous aux pages d'administration de DB2 Alphablox en tant qu'administrateur (ou utilisateur disposant de droits d'administration).
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Cubes**.
4. Cliquez sur le bouton **Créer**. La fenêtre Cube Administration s'ouvre.
5. Définissez le nouveau cube :
 - a. Dans la zone du nom de cube DB2 Alphablox, entrez CVSales.
 - b. Sélectionnez l'option **Activé**, située en regard du nom de cube DB2 Alphablox . Si vous sélectionnez cette option, votre cube démarre automatiquement lors du redémarrage de votre serveur.
 - c. Dans le menu de la source de données relationnelles, sélectionnez MyDB2, source de données relationnelles créée pour les besoins du présent tutoriel.
 - d. Veillez à ce que l'option du rôle de sécurité ne soit pas sélectionnée. Cette option sert à limiter le nombre d'utilisateurs susceptibles d'accéder à un cube particulier.
6. Activez les paramètres de DB2 Cube Views et indiquez les métadonnées à utiliser :
 - a. Sélectionnez l'option d'activation des paramètres DB2 Cube Views.
 - b. Dans le menu du modèle de cube, sélectionnez **CVSAMPLE.Sales**.
 - c. Dans le menu **Cube**, sélectionnez le cube **General Sales**
 - d. Sélectionnez le bouton d'option d'utilisation des noms métier pour indiquer les noms. Si vous sélectionnez cette option, les noms de membre généralement plus significatifs et plus lisibles sont utilisés.
 - e. Cliquez sur le bouton d'importation des définitions de cube. L'utilisation de cette option vous permet d'importer une définition de cube et de préremplir les mesures et les dimensions de votre cube DB2 Alphablox. La définition de cube importée est le reflet le plus fidèle possible du cube DB2 Cube Views que DB2 Alphablox est en mesure de générer en fonction des métadonnées DB2 Cube Views. Lorsque vous aurez acquis une meilleure maîtrise des cubes, vous pourrez adapter les mesures et les dimensions à vos besoins.

- f. Cliquez sur le bouton d'affichage du journal d'importation pour afficher le journal des messages d'information et de débogage liés à l'opération d'importation. Pour les besoins de ce tutoriel, cette étape est simplement incluse pour vous aider à vous familiariser avec cette fonction.
 - g. Sélectionnez l'option d'importation de définition de cube lors du démarrage, de la reconstruction et de l'édition. Si vous sélectionnez cette option, votre cube DB2 Alphablox utilise la dernière définition des vues de cube DB2 dès que le cube DB2 Alphablox est démarré, reconstitué ou ouvert pour être modifié. Lorsque vous aurez acquis une meilleure maîtrise de DB2 Alphablox et de DB2 Cube Views, vous pourrez importer la définition de cube et la personnaliser.
7. Cliquez sur le bouton **Enregistrer** pour enregistrer la définition de cube DB2 Alphablox.

Vous venez de créer une définition de cube DB2 Alphablox. Vous pouvez à présent lancer votre nouveau cube CVSales.

Démarrage de votre cube DB2 Alphablox

Cette procédure vous permet de lancer le cube CVSales via les pages d'administration de DB2 Alphablox.

Conditions préalables : définissez la source de données relationnelles DB2. Définissez une source de données d'adaptateur de serveur de cube DB2 Alphablox. Définissez le cube DB2 Alphablox.

Pour démarrer le cube CVSales :

1. Connectez-vous aux pages d'administration de DB2 Alphablox en tant qu'administrateur (ou utilisateur disposant de droits d'administration).
2. Cliquez sur l'onglet **Administration**.
3. Sous la section de gestion de l'exécution, cliquez sur le lien **Cubes**.
4. Dans la liste des cubes DB2 Alphablox, sélectionnez le cube DB2 Alphablox à démarrer.
5. Cliquez sur le bouton de démarrage. Lorsque le cube démarre, la zone d'état affiche **Exécution**.

Vous disposez à présent d'un cube DB2 Alphablox d'exemple en cours d'exécution. Vous pouvez maintenant commencer à créer des applications à l'aide du cube DB2 Alphablox que vous venez de créer. Pour procéder à une vérification rapide de votre nouveau cube, vous pouvez utiliser le générateur de requêtes : sélectionnez votre cube comme source de données et exécutez des requêtes rDX sur ce cube.

Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM EMEA Director of Licensing
IBM Europe Middle-East Africa
Tour Descartes
La Défense 5
2, avenue Gambetta
92066 - Paris-La Défense CEDEX
France

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada

Les informations sur les licences concernant les produits utilisant un jeu de caractères double octet peuvent être obtenues par écrit à l'adresse suivante :

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

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales. LE PRESENT DOCUMENT EST LIVRE EN L'ETAT. IBM DECLINE TOUTE RESPONSABILITE, EXPLICITE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE NON-CONTREFACON ET D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Il est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut modifier sans préavis les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

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

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation IBM.

Marques

Les termes qui suivent sont des marques d'International Business Machines Corporation aux Etats-Unis et/ou dans certains autres pays :

DB2	DB2 OLAP Server	DB2 Universal Database
IBM	WebSphere	

Alphablox et Blox sont des marques d'Alphablox Corporation aux Etats-Unis et/ou dans certains autres pays.

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques de Sun Microsystems, Inc. aux Etats-Unis et/ou dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

D'autres sociétés sont propriétaires des autres marques, noms de produits ou logos qui pourraient apparaître dans ce document.

Index

P

portlet
 sample 11
 structure JSP 12

Q

Query Builder
 utilisation 4
Query Builder, DHTML
 utilisation 4

R

requêtes
 génération à l'aide de Query
 Builder 4

T

tutoriels
 développement d'applications 1



Numéro de programme : 5724-L14

GC11-2396-01

