

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 «Remarques», à la page 47.

Troisième édition - septembre 2006

Réf. US : GC18-9607-02

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 | Création d'un projet de type portlet afin d'utiliser des composants Blox | 30 |
| Chapitre 1. Installation des bases de données exemple Alphablox | 1 | Configuration d'un projet de type portlet à l'aide de Rational Application Developer | 31 |
| Création d'un cube à partir d'une source de données relationnelles exemple | 2 | Etapes suivantes. | 33 |
| Installation des données exemple | 2 | Conseils relatifs au développement de portlets | 33 |
| Définition des sources de données | 6 | Chapitre 4. Tutoriel : Création d'applications à l'aide des outils Rational Developer | 35 |
| Création de la définition du cube | 7 | Préparation de votre environnement de développement | 35 |
| Installation de données exemple multidimensionnelles | 8 | Installation du DB2 Alphablox Toolkit | 35 |
| Installation de données exemple pour DB2 OLAP Server V8.1 et Hyperion Essbase V6.5 | 8 | Installation de DB2 Alphablox dans les environnements de test intégrés de WebSphere | 36 |
| Installation de données exemple pour DB2 OLAP Server V8.2 et Hyperion Essbase V7.1 | 9 | Création d'une instance de serveur WebSphere | 37 |
| Installation des données exemple pour Microsoft SQL Server 2000 Analysis Services. | 11 | Création de variables de substitution de serveur WebSphere 5 | 38 |
| Installation des données exemple pour Microsoft SQL Server 2005 Analysis Services. | 12 | Configuration d'instances de serveur WebSphere 5 | 38 |
| Définition de la source de données | 13 | Ajout de l'utilisateur invité au groupe des administrateurs de DB2 Alphablox | 39 |
| Chapitre 2. Tutoriel : Création de votre première application | 15 | Création d'applications DB2 Alphablox | 39 |
| Définition de votre application | 15 | Création de fichiers JSP avec du contenu DB2 Alphablox | 40 |
| Accès aux données | 16 | Accès aux sources de données DB2 OLAP Server et Essbase. | 40 |
| Création de votre page d'accueil d'application. | 16 | Chapitre 5. Tutoriel : Création d'un cube DB2 Alphablox à l'aide de DB2 Cube Views. | 43 |
| Définition d'une page d'accueil par défaut | 17 | Définition d'une source de données relationnelles DB2 | 43 |
| Création de votre première vue analytique | 18 | Définition d'une source de données d'adaptateur de serveur de cube Alphablox | 44 |
| Structure de votre première page de vue analytique | 21 | Définition d'un cube DB2 Alphablox | 45 |
| Création d'une vue analytique personnalisée | 23 | Démarrage de votre cube DB2 Alphablox | 46 |
| Récapitulatif | 24 | Remarques | 47 |
| Chapitre 3. Tutoriel : Création de votre premier portlet contenant des composants Blox | 25 | Marques | 49 |
| Installation des exemples de portlets | 25 | Index | 51 |
| Exécution des exemples de portlets | 26 | | |
| Etude de la structure d'une page JSP de portlet contenant un composant Blox | 26 | | |
| Création de votre propre page JSP de portlet contenant des composants Blox. | 29 | | |

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. Installation des bases de données exemple Alphablox

Des données exemple de la société Quality Chocolate Company (QCC) sont utilisées dans les tutoriels de DB2 Alphablox, dans les exemples d'applications Blox Sampler et dans de nombreux exemples de codes du centre de documentation de DB2 Alphablox.

Ces données exemple sont à la disposition de diverses bases de données :

Bases de données relationnelles

Pour suivre le premier tutoriel permettant de créer votre première application, vous devez installer une des versions relationnelles des données exemple QCC. Le cube QCC créé à partir des données relationnelles est utilisé dans les tutoriels et les exemples d'applications Blox Sampler.

- DB2
- Apache Derby
- Informix
- Microsoft SQL Server
- Oracle
- Sybase

Bases de données multidimensionnelles

Les exemples Blox Sampler fonctionnent également sur les bases de données multidimensionnelles suivantes. Celles-ci ne sont pas nécessaires pour les tutoriels. Toutefois, si vous utilisez l'une de ces bases de données dans votre environnement, vous devez installer les données exemple QCC pour pouvoir exploiter les exemples fournis dans les applications Blox Sampler.

- IBM DB2 OLAP Server V8.1, Hyperion Essbase V6.5
- IBM DB2 OLAP Server V8.2, Hyperion Essbase V7.1
- Microsoft SQL Server 2000 Analysis Services
- Microsoft SQL Server 2005 Analysis Services

Présentation de QCC

Quality Chocolate Company, l'entreprise que nous avons imaginée, est une chaîne de magasins de détail en pleine croissance, qui vend un petit choix de confiseries traditionnelles et de spécialités saisonnières. QCC a débuté dans une petite ville, puis a développé sa gamme de produits et multiplié le nombre de ses magasins en quelques années seulement. Cette entreprise souhaite à présent effectuer l'analyse de ses ventes, de sa marge bénéficiaire brute et de son budget. En outre, QCC compte effectuer le suivi de diverses caractéristiques de ses produits, telles que le type de chocolats (noir, au lait...), la présence de noisettes, amandes, etc., afin d'analyser les préférences de sa clientèle.

Création d'un cube à partir d'une source de données relationnelles exemple

Des tables relationnelles destinées aux données exemple de la Quality Chocolate Company (QCC) et un fichier de propriétés de la définition du cube sont fournis pour vous permettre de créer un cube Alphablox à partir de vos données relationnelles.

La création d'un cube Alphablox à l'aide des données relationnelles QCC fournies comporte trois grandes étapes :

1. Installation des tables exemple.
2. Définition des sources de données pour la source de données relationnelles sous-jacente et le cube à créer sur la source de données relationnelles, via les pages d'administration de DB2 Alphablox.
3. Création de la définition du cube en enregistrant dans votre référentiel DB2 Alphablox le fichier des propriétés de définition du cube fourni. Ce fichier de propriétés contient les informations relatives aux dimensions, mesures et niveaux du cube ainsi qu'à l'emplacement des données dans la source de données relationnelles sous-jacente.

Installation des données exemple

Pour pouvoir créer un cube Alphablox à l'aide des données exemple de la Quality Chocolate Company (QCC), vous devez préalablement installer les tables QCC sur votre serveur de base de données.

Reportez-vous aux instructions d'installation relatives à votre serveur de base de données.

Installation des données exemple pour DB2

Des données exemple sont fournies pour les sources de données DB2. Vous devez préalablement installer les tables exemple pour pouvoir les utiliser dans la création du cube Alphablox nécessaire au premier tutoriel.

Pour installer les données exemple de la Quality Chocolate Company (QCC) pour les sources de données DB2 :

1. Copiez tous les fichiers du répertoire `sampledata/qcc/db2` du CD d'installation d'Alphablox dans un répertoire du poste sur lequel le client DB2 est installé.
2. Ajoutez votre connexion DB2 au centre de contrôle.
3. A partir d'une invite de commande, accédez au répertoire dans lequel vous avez copié les fichiers et exécutez le fichier `qcc_db2.bat` avec les paramètres de nom de votre base de données DB2, de nom d'utilisateur et de mot de passe. Par exemple : `qcc_db2 test db2admin password`. Cette commande permet de vous connecter à la base de données DB2 intitulée `test`, en tant qu'utilisateur `db2admin`, avec le mot de passe `password`, et de charger les données dans la table `qcc`. Deux fichiers sont créés :
 - Un fichier `db2_command.txt` contenant le paramètre de connexion à DB2 et le fichier `db2_qcc.sql`.
 - Un fichier journal `qcc.log` pour le chargement des données QCC.
4. Consultez les messages éventuels figurant dans le fichier journal.

Les tables sont installées.

Passez à l'étape «Définition des sources de données», à la page 6 pour définir la source de données relationnelles et la source de données du cube dans DB2 Alphablox.

Installation des données exemple pour Derby

Des données exemple sont fournies pour les sources de données Derby. Vous devez préalablement installer les tables exemple pour pouvoir les utiliser dans la création du cube Alphablox nécessaire au premier tutoriel.

Pour installer les données exemple de la Quality Chocolate Company (QCC) pour les sources de données Derby :

1. Copiez tous les fichiers du répertoire `sampledata/qcc/derby` du CD d'installation d'Alphablox dans un répertoire du poste sur lequel le serveur Derby est installé.
2. Ouvrez le fichier `qcc_dby.bat` dans un éditeur de texte.
3. Vérifiez les chemins `DERBY_INSTALL` et `JAVA_INSTALL` et modifiez-les si nécessaire. Les valeurs par défaut de `DERBY_INSTALL` et `JAVA_INSTALL` sont respectivement `C:\Program Files\IBM\Cloudscape_10.0` et `C:\Program Files\IBM\Cloudscape_10.0\ibm-jre-n142p\jre\bin`.
4. A partir d'une invite de commande, accédez au répertoire dans lequel vous avez copié les fichiers et exécutez le fichier `qcc_dby.bat` avec les paramètres de nom de votre base de données DB2, de nom d'utilisateur et de mot de passe. Ceci entraîne la création d'une base de données Blox et de tables pour l'utilisateur abx. Un fichier journal `qcc.log` est également créé pour le chargement des données de qcc.

Remarque : Si elle existe déjà, la table QCC sera supprimée.

5. Consultez les messages éventuels figurant dans le fichier journal.

Les tables sont installées.

Passez à l'étape «Définition des sources de données», à la page 6 pour définir la source de données relationnelles et la source de données du cube dans DB2 Alphablox.

Installation des données exemple pour Informix

Des données exemple sont fournies pour les sources de données Informix. Vous devez préalablement installer les tables exemple pour pouvoir les utiliser dans la création du cube Alphablox nécessaire au premier tutoriel.

Vous devez avoir créé les variables d'environnement `INFORMIXDIR` et `INFORMIXSERVER`. Veuillez consulter le manuel *DB-Access User's Manual* d'IBM Informix pour plus de détails sur l'utilisation de **DB-Access**.

Pour installer des données exemple QCC pour les sources de données Informix :

1. Sur votre serveur Informix, créez une base de données dans laquelle vous chargerez les données exemple.
2. Copiez tous les fichiers du répertoire `sampledata/qcc/informix` du CD d'installation d'Alphablox sur le poste sur lequel le serveur Derby est installé.
3. A partir d'une invite de commande, accédez au répertoire dans lequel vous avez enregistré les fichiers et tapez `dbaccess`.
4. Connectez-vous à la base de données dans laquelle vous souhaitez charger les données exemple.
 - a. Sélectionnez **DBAccess** → **Connection**.

- b. Sélectionnez **Connect**.
 - c. Sélectionnez le serveur de base de données.
 - d. Tapez le nom d'utilisateur de l'administrateur de base de données.
 - e. Tapez le mot de passe.
 - f. Sélectionnez la base de données. La base de données est connectée.
 - g. Sélectionnez **Exit**.
5. Chargez le fichier SQL pour créer les tables.
 - a. Sélectionnez **DBAccess** → **Query-language**.
 - b. Sélectionnez **Choose**.
 - c. Sélectionnez le fichier qcc_inf.sql.
 - d. Sélectionnez **run**. Les tables sont créées.
 - e. Sélectionnez **Exit** pour revenir à l'invite de commande.
 6. Chargez les données. Exécutez qcc_inf.bat à partir d'une invite de commande avec le paramètre du nom de base de données. Par exemple : qcc_info blox.

Remarque : Le fichier journal qcc.log est créé après l'exécution du fichier de commandes.

Les tables sont installées.

Passez à l'étape «Définition des sources de données», à la page 6 pour définir la source de données relationnelles et la source de données du cube dans DB2 Alphablox.

Installation des données exemple pour Microsoft SQL Server

Des données exemple sont fournies pour les sources de données Microsoft SQL Server. Vous devez préalablement installer les tables exemple pour pouvoir les utiliser dans la création du cube Alphablox nécessaire au premier tutoriel.

Pour installer les données exemple de la Quality Chocolate Company (QCC) pour les sources de données Microsoft SQL Server :

1. Copiez tous les fichiers du répertoire `sampledata/qcc/mssql` du CD d'installation d'Alphablox dans un répertoire de votre poste.
2. A partir d'une invite de commande, exécutez le fichier `qcc_sql.bat` et spécifiez les paramètres de connexion de votre serveur SQL (*serveur base de données table propriétaire utilisateur mot de passe*). Par exemple, `qcc_sql server12 blox dbo sa`. Cette commande exécute le fichier de commandes et établit une connexion à la base de données du serveur SQL, située sur le serveur `server12` : la base de données s'intitule `blox`, le propriétaire de la table est `dbo`, le nom d'utilisateur est `sa` et il n'y a pas de mot de passe. L'exécution du script `qcc_sql.bat` a pour effet de créer les fichiers journaux suivants :
 - Un fichier `qcc.log` pour la création des tables `qcc`.
 - Un fichier `qcc_fact.log` pour le chargement des données dans la table `qcc_fact`.
 - Un fichier `qcc_locations.log` pour le chargement des données dans la table `qcc_locations`.
 - Un fichier `qcc_products.log` pour le chargement des données dans la table `qcc_products`.
 - Un fichier `qcc_scenarios.log` pour le chargement des données dans la table `qcc_scenarios`.

- Un fichier qcc_time.log pour le chargement des données dans la table qcc_time.

Remarque : Si elle existe déjà, la table qcc sera supprimée.

3. Consultez les messages éventuels figurant dans les fichiers journaux.

Les tables sont installées.

Passez à l'étape «Définition des sources de données», à la page 6 pour définir la source de données relationnelles et la source de données du cube dans DB2 Alphablox.

Installation de données exemple pour Oracle

Des données exemple sont fournies pour les sources de données Oracle. Vous devez préalablement installer les tables exemple pour pouvoir les utiliser dans la création du cube Alphablox nécessaire au premier tutoriel.

Pour installer les données exemple de la Quality Chocolate Company (QCC) pour les sources de données Oracle :

1. Copiez tous les fichiers du répertoire `sampledata/qcc/oracle` du CD d'installation d'Alphablox sur le poste sur lequel le client Oracle est installé.
2. Exécutez le fichier `qcc_ora.bat` avec les paramètres de connexion d'Oracle. Par exemple, dans une invite DOS, tapez `qcc_ora server1 oraadmin tiger`. Cette commande exécute le fichier de commandes et connecte l'utilisateur `oraadmin` à la base de données Oracle du serveur `server1`, avec le mot de passe `tiger`. Les fichiers journaux suivants sont créés lors de l'exécution du fichier de commandes :
 - Un fichier `qcc.log` pour la création des tables `qcc`.
 - Un fichier `qcc_fact.log` pour le chargement des données dans la table `qcc_fact`.
 - Un fichier `qcc_locations.log` pour le chargement des données dans la table `qcc_locations`.
 - Un fichier `qcc_products.log` pour le chargement des données dans la table `qcc_products`.
 - Un fichier `qcc_scenarios.log` pour le chargement des données dans la table `qcc_scenarios`.
 - Un fichier `qcc_time.log` pour le chargement des données dans la table `qcc_time`.
3. Consultez les messages éventuels figurant dans les fichiers journaux.

Les tables sont installées.

Passez à l'étape «Définition des sources de données», à la page 6 pour définir la source de données relationnelles et la source de données du cube dans DB2 Alphablox.

Installation des données exemple pour Sybase

Des données exemple sont fournies pour les sources de données Sybase. Vous devez préalablement installer les tables exemple pour pouvoir les utiliser dans la création du cube Alphablox nécessaire au premier tutoriel.

Pour installer les données exemple de la Quality Chocolate Company (QCC) pour les sources de données Sybase :

1. Copiez tous les fichiers du répertoire `sampledata/qcc/sybase` du CD d'installation d'Alphablox dans un répertoire.
2. Exécutez le fichier `qcc_syb.bat` avec les paramètres de connexion de Sybase. Par exemple, dans une invite DOS, tapez `qcc_syb server1 blox dbo sa`. Cette commande exécute le fichier de commandes et établit une connexion à Sybase, sur le serveur `server1`, la base de données s'intitule `blox`, le propriétaire de la table est `dbo`, le nom d'utilisateur est `sa` et il n'y a pas de mot de passe. Les fichiers journaux suivants sont créés lors de l'exécution du fichier de commandes :
 - `qcc.log` pour la création des tables `qcc`
 - `qcc_fact.log` pour le chargement des données dans la table `qcc_fact`
 - `qcc_locations.log` pour le chargement des données dans la table `qcc_locations`
 - `qcc_products.log` pour le chargement des données dans la table `qcc_products`
 - `qcc_time.log` pour le chargement des données dans la table `qcc_time`
3. Consultez les messages éventuels figurant dans les fichiers journaux.

Les tables sont installées.

Remarque : Si elle existe déjà, la table `qcc` sera supprimée.

Passez à l'étape «Définition des sources de données» pour définir la source de données relationnelles et la source de données du cube dans DB2 Alphablox.

Définition des sources de données

Une fois les tables de données relationnelles de la Quality Chocolate Company (QCC) installées pour votre base de données, définissez deux sources de données, l'une pour la source de données relationnelles, l'autre pour le cube que vous allez créer.

Pour définir les sources de données :

1. Créez la source de données `qcc2003-rdb`. Celle-ci est utilisée comme source de données relationnelles sous-jacente de votre cube QCC.
 - a. Ouvrez la page d'accueil de DB2 Alphablox dans une fenêtre du navigateur. Son URL est `http://votreserveur:port/AlphabloxAdmin/home`.
 - b. Sélectionnez l'onglet **Administration** > **Data Sources**.
 - c. Cliquez sur **Create** pour créer une connexion entre la source de données et les tables QCC.
 - d. Dans la zone réservée au nom de la source de données, saisissez `qcc2003-rdb`.
 - e. Spécifiez les paramètres de connexion appropriés des tables QCC.
 - f. Cliquez sur **Save**.
2. Créez la source de données `qcc2003-acs`. Cette source de données est destinée au cube QCC.
 - a. Sélectionnez **Administration** > **Data Sources** sur la page d'accueil de DB2 Alphablox.
 - b. Cliquez sur **Create** afin de créer une connexion à la source de données pour le cube DB2 Alphablox.
 - c. Dans la zone réservée au nom de la source de données, tapez `qcc2003-acs` et sélectionnez **Alphablox Cube Server Adapter** dans la liste **Adapter**.
 - d. Cliquez sur **Save**.

La dernière étape de la création d'un cube à l'aide des données exemple QCC consiste à enregistrer le fichier de la définition du cube fourni dans le référentiel de DB2 Alphablox.

- Si vous utilisez un référentiel basé sur un fichier, suivez la procédure de la section «Création de la définition du cube exemple pour un référentiel basé sur un fichier».
- Si vous utilisez une base de données pour le référentiel, suivez la procédure de la section «Création de la définition du cube exemple pour un référentiel basé sur une base de données».

Création de la définition du cube

Reportez-vous aux instructions relatives à la création de la définition du cube en fonction du référentiel que vous utilisez.

Création de la définition du cube exemple pour un référentiel basé sur un fichier

Après l'installation des données exemple de la Quality Chocolate Company (QCC) et la définition des sources de données exemple, vous devez créer la définition du cube.

Vous devez avoir installé les données exemple QCC et défini la source de données relationnelles et le cube dans DB2 Alphablox.

Pour créer la définition du cube QCC :

1. Dans le répertoire *db2alphablox_dir/repository*, créez un répertoire "cubes".
2. Dans le répertoire *db2alphablox_dir/repository/cubes*, créez un répertoire "qcc_2003".
3. Copiez *qcc_2003.properties* du répertoire *sampledata/qcc/acs* du CD vers le répertoire *alphablox_dir/repository/cubes/qcc_2003*.
4. Redémarrez le serveur d'applications sur lequel DB2 Alphablox fonctionne.

La définition du cube est désormais enregistrée dans le référentiel de DB2 Alphablox. Le cube est créé et défini dans DB2 Alphablox et utilisable avec les tutoriels et Blox Sampler.

Création de la définition du cube exemple pour un référentiel basé sur une base de données

Après l'installation des données exemple de la Quality Chocolate Company (QCC) et la définition des sources de données exemple, vous devez créer la définition du cube.

Vous devez avoir installé les données exemple QCC et défini la source de données relationnelles et le cube dans DB2 Alphablox.

Pour créer la définition du cube QCC :

1. Créez un répertoire temporaire sur votre système. Par exemple, *c:\temp*.
2. Copiez *qcc_2003.properties* du répertoire *sampledata/qcc/acs* du CD vers le répertoire temporaire créé à l'étape 1.
3. Arrêtez DB2 Alphablox en arrêtant le serveur d'applications sur lequel DB2 Alphablox fonctionne.
4. Exécutez *ConvertRepository.bat* (ou *ConvertRepository.sh*) dans le répertoire *db2alphablox_dir/bin/* pour convertir la définition du cube stockée dans le fichier *qcc_2003.properties* en référentiel de votre base de données.

5. Choisissez l'option permettant la conversion d'un référentiel à l'autre (option 3).
6. Choisissez l'option permettant la conversion d'un fichier en une base de données (option 1).
7. Lorsque vous êtes invité à indiquer la racine du référentiel, tapez le chemin du répertoire temporaire dans lequel vous avez copié le fichier qcc_2003.properties à l'étape 2.
8. Suivez les invites et fournissez les paramètres de connexion appropriés.
9. Lorsque vous êtes invité à choisir l'opération de conversion, tapez U pour mettre à jour la base de données.
10. Une fois la conversion effectuée, redémarrez DB2 Alphablox en lançant le serveur d'applications sur lequel DB2 Alphablox fonctionne.

Le cube est créé et défini dans DB2 Alphablox et utilisable avec les tutoriels et Blox Sampler.

Installation de données exemple multidimensionnelles

Des versions multidimensionnelles des données exemple de la Quality Chocolate Company (QCC) sont fournies pour être utilisées avec Blox Sampler.

Deux grandes étapes permettent d'installer les données exemple QCC et de les rendre utilisables avec Blox Sampler :

1. Installation des données exemple. Reportez-vous aux instructions d'installation relatives à votre base de données multidimensionnelle.
 - DB2 OLAP Server V8.1 et Hyperion Essbase V6.5
 - DB2 OLAP Server V8.2 et Hyperion Essbase V7.1
 - Microsoft SQL Server 2000 Analysis Services
 - Microsoft SQL Server 2005 Analysis Services
2. Définition de la source de données dans DB2 Alphablox.

Installation de données exemple pour DB2 OLAP Server V8.1 et Hyperion Essbase V6.5

Des données exemple sont fournies pour les sources de données de DB2 OLAP Server V8.1 et Hyperion Essbase V6.5. Ces données exemple de la Quality Chocolate Company (QCC) peuvent être utilisées avec Blox Sampler.

Pour installer des données exemple QCC pour les sources de données DB2 OLAP Server V8.1 et Hyperion Essbase V6.5 :

1. Connectez-vous à DB2 OLAP Server ou à Application Manager Essbase.
 - a. Démarrez DB2 OLAP Server ou Hyperion Essbase.
 - b. Ouvrez l'application Application Manager.
 - c. Sélectionnez **Server** → **Connect** et saisissez vos nom de connexion et mot de passe.
 - d. Cliquez sur **OK**.
2. Créez une application.
 - a. Sélectionnez **File** → **New** → **Application**.
 - b. Tapez le nom d'application Blox et l'emplacement Server.
 - c. Cliquez sur **OK**.
3. Créez une base de données.

- a. Sélectionnez **File** → **New** → **Database**.
- b. Tapez le nom de base de données qcc.
4. Ouvrez et enregistrez la structure qcc.
 - a. Cliquez sur **Open** pour ouvrir la structure.
 - b. Sélectionnez **File** → **Open** dans la fenêtre Application Manager (Gestionnaire d'application).
 - c. Dans la zone **Location**, sélectionnez **Client**.
 - d. Cliquez sur le bouton **File System** et accédez au fichier de structure qcc.otl, situé dans le répertoire `sampledata/qcc/essbase6/` du CD de DB2 Alphablox.
 - e. Cliquez sur le bouton **Open** pour ouvrir la structure.
 - f. Fermez la fenêtre Outline initiale. Cette fenêtre ne comporte pas de hiérarchie.
 - g. Une fois la fenêtre New Outlines ouverte, sélectionnez **File** → **Save As** dans la fenêtre Application Manager.
 - h. Dans la zone **Location**, sélectionnez Server.
 - i. Dans la zone **Application**, sélectionnez blox.
 - j. Dans la zone **Database**, sélectionnez qcc.
 - k. Cliquez sur **OK**.
 - l. Fermez la fenêtre Outline.
5. Chargez les données qcc.
 - a. Depuis la fenêtre Application Manager, sélectionnez **Database** → **Clear Data** → **All**.
 - b. Sélectionnez **Database** → **Load Data**
 - c. Dans la zone **Data Files**, sélectionnez **Find**.
 - d. Dans la fenêtre **Open Server Data File Objects** qui s'ouvre, sélectionnez le bouton d'option **Client** dans la zone **Location**.
 - e. Cliquez sur le bouton **File System**.
 - f. Accédez au fichier qcc.txt, situé dans le répertoire `sampledata/qcc/essbase6/` du CD de DB2 Alphablox, et cliquez sur **Open** pour l'ajouter à la liste des fichiers de données.
 - g. Cliquez sur **OK**. Les fichiers sont chargés.
6. Calculez le cube.
 - a. Sélectionnez **Database** → **Calculate**
 - b. Une fois l'opération terminée, quittez Application Manager.

La base de données exemple est installée.

Ensuite, procédez à la définition de la source de données dans DB2 Alphablox en suivant la procédure décrite dans la section «Définition de la source de données», à la page 13.

Installation de données exemple pour DB2 OLAP Server V8.2 et Hyperion Essbase V7.1

Des données exemple sont fournies pour les sources de données DB2 OLAP Server V8.2 et Hyperion Essbase V7.1. Ces données exemple de la Quality Chocolate Company (QCC) peuvent être utilisées avec Blox Sampler.

Pour installer des données exemple QCC pour les sources de données DB2 OLAP Server V8.2 et Hyperion Essbase V7.1 :

1. Connectez-vous à DB2 OLAP Server ou aux services d'administration Essbase.
 - a. Démarrez DB2 OLAP Server ou Hyperion Essbase.
 - b. Ouvrez la console d'administration de DB2 OLAP Server ou des services d'administration Hyperion Essbase.
 - c. Vérifiez que DB2 OLAP Server ou Hyperion Essbase apparaît dans le cadre de navigation, sous **Analytic Servers**.
2. Créez une application.
 - a. Dans l'arborescence de navigation figurant sous votre serveur, cliquez avec le bouton droit de la souris sur **Applications** et sélectionnez **Create Application** → **Using Block Storage**.
 - b. Dans la fenêtre Create Application, tapez Blox dans la zone du nom de l'application.
 - c. Cliquez sur **OK**.
3. Créez une base de données.
 - a. Développez l'application Blox que vous venez de créer.
 - b. Cliquez avec le bouton droit de la souris sur **Databases**, puis sélectionnez l'option de création de bases de données.
 - c. Dans la fenêtre Create Database, tapez qcc et cliquez sur **OK**.
 - d. Développez **Databases** et vérifiez que la base de données qcc existe.
4. Ouvrez et enregistrez la structure qcc.
 - a. Dans la barre de menus de la console d'administration, sélectionnez **File** → **Open**.
 - b. Dans la fenêtre Open, cliquez sur l'onglet **File System** et accédez au répertoire `sampledata/qcc/essbase6` du CD de DB2 Alphablox.
 - c. Sélectionnez le fichier de structure qcc.otl et cliquez sur le bouton **OK**.
 - d. Vérifiez que la nouvelle structure s'affiche dans le cadre de droite.
 - e. Dans la barre de menus de la console d'administration, sélectionnez **File** → **Save As**.
 - f. Dans la fenêtre Save As, cliquez sur l'onglet **Analytic Server** et accédez au répertoire qcc.
 - g. Sélectionnez le fichier qcc.otl existant et cliquez sur **OK**.
 - h. Dans la fenêtre Confirmation, cliquez sur **Oui** pour remplacer le fichier existant.
 - i. Lorsque vous êtes invité à ouvrir la structure, cliquez sur **Yes** pour vérifier que la nouvelle structure s'affiche.
5. Chargez les données qcc.
 - a. Dans la console d'administration, cliquez avec le bouton droit de la souris sur la liste où figure la base de données qcc et sélectionnez l'option de chargement des données.
 - b. Dans la fenêtre Data Load, cliquez sur le bouton **Find Data File**. La fenêtre Open s'affiche.
 - c. Cliquez sur l'onglet du système de fichiers et accédez au répertoire `sampledata/qcc/essbase6` du CD de DB2 Alphablox.
 - d. Sélectionnez le fichier de données qcc.txt et cliquez sur **OK**.
 - e. Dans la fenêtre Data Load, vérifiez que le fichier de données apparaît et cliquez sur **OK**. La fenêtre Data Load Completed apparaît, indiquant que le fichier de données a été chargé sans incidents.

- f. Cliquez sur **Close**.
6. Calculez le cube.
 - a. Dans la console d'administration, cliquez à nouveau avec le bouton droit de la souris sur la liste où figure la base de données qcc, mais sélectionnez l'option **Execute Calculation**.
 - b. Dans la fenêtre Execute Database Calculation, vérifiez que le calcul par défaut est mis en évidence, puis cliquez sur **OK**. A la fin du script de calcul, un message indiquant la bonne exécution du script de calcul devrait apparaître dans la fenêtre Messages.
 - c. Fermez la console d'administration.

La base de données exemple est installée.

Ensuite, procédez à la définition de la source de données dans DB2 Alphablox en suivant la procédure décrite dans la section «Définition de la source de données», à la page 13.

Installation des données exemple pour Microsoft SQL Server 2000 Analysis Services

Des données exemple sont fournies pour les sources de données Microsoft SQL Server 2000 Analysis Services. Ces données exemple de la Quality Chocolate Company (QCC) peuvent être utilisées avec Blox Sampler.

Pour installer des données exemple QCC pour Microsoft SQL Server 2000 Analysis Services :

1. Copiez les fichiers Blox.CAB et qcc.mdb du répertoire sampledata/qcc/msas du CD d'installation de DB2 Alphablox vers un répertoire de votre système (par exemple, C:\Blox).
2. Cliquez avec le bouton droit de la souris sur le fichier qcc.mdb et assurez-vous que l'option **Read Only** est désélectionnée.
3. Dans Analysis Manager, restaurez le cube :
 - a. Cliquez avec le bouton droit de la souris sur le serveur de base de données, puis sélectionnez **Restore Database**.
 - b. Accédez au fichier Blox.CAB, copié à l'étape 1.
 - c. Sélectionnez Blox.CAB et cliquez sur **Open**. Si vous êtes invité à écraser la base de données existante, cliquez sur **Yes**.
 - d. Dans la fenêtre Restore Database, cliquez sur **Restore**.
 - e. Lorsqu'un message indiquant que la base de données a été restaurée s'affiche à l'extrémité de la fenêtre, cliquez sur **Close**.
4. Si votre fichier qcc.mdb n'est pas enregistré dans C:\Blox, ce qui est prévu par le fichier CAB, modifiez la source de données comme suit :
 - a. Développez la base de données Blox.
 - b. Développez **Data Source**.
 - c. Cliquez avec le bouton droit de la souris sur l'option "C blox qcc mdb" et sélectionnez **Edit**.
 - d. Sélectionnez l'onglet **Connection**.
 - e. Modifiez le chemin de votre fichier qcc.mdb dans la zone **Select** ou tapez un nom de base de données dans la zone de modification.
 - f. Cliquez sur **Test Connection** pour vérifier que vous pouvez vous connecter à la source de données.

- g. Cliquez sur **OK**.
- 5. Vérifiez que la fonctionnalité d'écriture différée est activée pour le cube qccwb.
 - a. Développez la base de données Blox.
 - b. Développez le dossier **Cubes**.
 - c. Cliquez avec le bouton droit de la souris sur le cube qccwb et sélectionnez **Write-Enable** dans le menu. Si **Writeback Options** apparaît dans le menu contextuel, la fonctionnalité d'écriture différée est activée.
 - d. Cliquez sur **OK**
 - e. Si la fonctionnalité d'écriture différée n'est pas activée, cliquez de nouveau sur le cube qccwb avec le bouton droit de la souris. L'option d'écriture différée doit alors apparaître.

La base de données exemple est installée.

Ensuite, procédez à la définition de la source de données dans DB2 Alphablox en suivant la procédure décrite dans la section «Définition de la source de données», à la page 13.

Installation des données exemple pour Microsoft SQL Server 2005 Analysis Services

Des données exemple sont fournies pour les sources de données Microsoft SQL Server 2005 Analysis Services. Ces données exemple de la Quality Chocolate Company (QCC) peuvent être utilisées avec Blox Sampler.

Pour installer des données exemple QCC pour Microsoft SQL Server 2005 Analysis Services :

1. Copiez les fichiers qcc_ssas.abf et qcc_ssas.mdb du répertoire `sampledata/qcc/ssas2005` du CD d'installation de DB2 Alphablox vers un répertoire de votre système (par exemple, `C:\sampledata\qcc\ssas2005`).
2. Cliquez avec le bouton droit de la souris sur le fichier qcc_ssas.mdb et assurez-vous que l'option **Read Only** est désélectionnée.
3. Dans SQL Server Management Studio, restaurez le cube :
 - a. Cliquez avec le bouton droit de la souris sur le dossier **Databases**, sous votre serveur de base de données et sélectionnez **Restore**.
 - b. Sélectionnez l'option **Restore target -> From backup file**.
 - c. Dans la zone **Selected path**, indiquez le chemin d'accès au fichier qcc_ssas.abf.
 - d. Tapez le nom de fichier qcc_ssas et cliquez sur **OK**.
 - e. Tapez password dans la zone **Password encryption** (Chiffrement par mot de passe) et cliquez sur **OK**. Ce mot de passe est nécessaire car le fichier de sauvegarde est chiffré.
 - f. Une fois la restauration terminée, cliquez avec le bouton droit de la souris sur le dossier **Databases** et sélectionnez **Refresh** pour afficher la base de données MyBlox nouvellement créée.
 - g. Cliquez avec le bouton droit de la souris sur le dossier et attribuez le nom Blox à la base de données intitulée MyBlox.
4. Si votre fichier qcc_ssas.mdb n'est pas enregistré dans `C:\sampledata\qcc\msas\ssas2005`, modifiez la source de données comme suit :
 - a. Développez la base de données Blox.
 - b. Développez **Data Source**.

- c. Cliquez avec le bouton droit de la souris sur le nom de la vue et sélectionnez **Propriétés**.
 - d. Sélectionnez la zone de chaîne de connexion dans la table pour afficher la boîte de dialogue **Connection Manager** (Gestionnaire de connexions).
 - e. Dans la zone réservée au **nom du fichier base de données**, entrez le chemin et le nom du fichier, puis cliquez sur **OK**.
5. Vérifiez si la fonctionnalité d'écriture différée est activée pour le cube qccwb.
 - a. Développez la base de données Blox.
 - b. Développez le dossier **Cubes**.
 - c. Développez le cube qccwb.
 - d. Développez les **groupes de mesures (Measure Groups)**.
 - e. Développez le cube qccwb.
 - f. Cliquez avec le bouton droit de la souris sur le dossier **Writeback** et sélectionnez l'option **Enable Writeback**.
 6. Cliquez avec le bouton droit de la souris sur le dossier Blox, sélectionnez **Process**, puis cliquez sur **OK**.

La base de données exemple est installée.

Ensuite, procédez à la définition de la source de données dans DB2 Alphablox en suivant la procédure décrite dans la section «Définition de la source de données».

Définition de la source de données

Une fois les données d'exemple de la Quality Chocolate Company (QCC) installées pour votre base de données, définissez la source de données dans DB2 Alphablox.

Avant de continuer, vérifiez que les données exemple de la QCC sont installées.

Pour définir la source de données :

1. Accédez à la page d'accueil de DB2 Alphablox. Son URL est `http://votre_serveur:port/AlphabloxAdmin/home`.
2. Sélectionnez **Administration > Data Sources**.
3. Cliquez sur **Create**.
4. Entrez les valeurs suivantes pour définir cette source de données :

Tableau 1. Valeurs de la définition de la source de données QCC

| Zone | DB2 OLAP Server/Hyperion Essbase | Microsoft SQL Server Analysis Services |
|------------------------------|--|--|
| Nom de la source de données | QCC-Essbase | QCC-MSAS |
| Carte | IBM DB2 OLAP Server ou Hyperion Essbase | OLE DB pour OLAP |
| Nom du serveur | <i>nom de votre serveur de données IBM DB2 OLAP Server ou Hyperion Essbase</i> | <i>nom de votre serveur de données Microsoft Analysis Services</i> |
| Application | Blox | |
| Base de données | qcc | Blox |
| Nom d'utilisateur par défaut | <i>nomd'utilisateur</i> | <i>nom de connexion</i> |
| Mot de passe par défaut | <i>motdepasse</i> | <i>motdepasse</i> |

Tableau 1. Valeurs de la définition de la source de données QCC (suite)

| Zone | DB2 OLAP Server/Hyperion Essbase | Microsoft SQL Server Analysis Services |
|---|----------------------------------|--|
| Utiliser le nom d'utilisateur et le mot de passe de DB2 Alphablox | Non | Non |
| Nombre maximal de lignes | 1 000 | 1 000 |
| Nombre maximal de colonnes | 1 000 | 1 000 |

5. Cliquez sur **Save**.

Il est désormais possible d'utiliser la source de données avec Blox Sampler.

Chapitre 2. Tutoriel : Création de votre première application

Ce tutoriel décrit d'importants concepts de base relatifs aux composants Blox lors de la création d'une application analytique simple 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 DB2 Alphablox.

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.

Prérequis

DB2 Alphablox doit être installé et accessible. Vous devez disposer d'un nom de connexion Alphablox doté de droits d'administration. Les bases de données exemple Alphablox doivent également être installées, comme décrit dans le Chapitre 1, «Installation des bases de données exemple Alphablox», à la page 1.

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.
 - a. Connectez-vous aux pages d'administration de DB2 Alphablox en tant qu'utilisateur membre du groupe d'administrateurs.
 - b. Cliquez sur l'onglet **Administration**.
 - c. Cliquez sur le lien **Applications**.
 - d. Cliquez sur le bouton **Create**. La page **Create Application** s'affiche.
 - e. Tapez MyApp dans la zone **Name**.
 - f. Tapez My App (avec un espace) dans la zone du nom d'affichage, ce qui définit le libellé de l'application tel qu'il apparaît dans la liste de la page Applications.
 - g. Cliquez sur le bouton **Save** pour définir la nouvelle application et revenir à la page **Applications**.

Remarque : Si vous utilisez une configuration Apache Tomcat avec un serveur Web externe (Apache ou IIS), vous devez suivre les étapes nécessaires à votre configuration pour que l'application soit disponible. Pour plus de détails, reportez-vous au *Guide d'administration*.

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 est vide. Le dossier de la nouvelle application se trouve sur votre serveur d'applications (pour le serveur WebSphere, le dossier se trouve dans le répertoire installedApps de WebSphere).

Vous avez créé une structure d'application DB2 Alphablox. Vous pourrez ensuite lui ajouter des vues analytiques.

Accès aux données

Vous devez vérifier que votre base de données relationnelle et le serveur Alphablox Cube fonctionnent pour pouvoir accéder à la source de données exemple du tutoriel. Le cube DB2 Alphablox doit également être en cours de fonctionnement.

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 le lien **Data Sources**.
4. Sur la page des sources de données, recherchez la source qcc2003-rdb dans le menu se trouvant sur le côté gauche de la page.
5. Cliquez sur qcc2003-rdb dans la liste des sources de données disponibles, puis sur le bouton **Test Selected Data Source**. 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 (en sélectionnant la source de données et en cliquant sur le bouton **Edit**) ou assurez-vous que la base de données est correctement configurée et que son utilisateur dispose 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 **Test Selected Data Source**. 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.

Vous pouvez à présent apprendre à créer des vues analytiques simples DB2 Alphablox.

Création de votre page d'accueil d'application

Ce cours vous permet de créer une page d'accueil simple en ayant recours à des compétences de base en matière de programmation Web. La page d'accueil comportera des liens vers des vues analytiques que vous créerez dans une prochaine étape de ce tutoriel.

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, créez un fichier intitulé `index.html` pour votre page d'accueil. Pour ce tutoriel, ce fichier peut être très simple.
2. Modifiez le fichier afin d'insérer le titre de l'application (par exemple, "Mon application DB2 Alphablox") et deux liens pointant vers les deux vues analytiques que vous créerez plus loin.

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="ChartBloxView.jsp">Vue ChartBlox 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 la page **Applications**.
5. Si DB2 Alphablox est installé sur un serveur d'applications IBM WebSphere ou un serveur BEA WebLogic, les listes de répertoires doivent être activées avant de passer à l'étape suivante. 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://hôtelocal:9080/MyApp/myapp.html`). La nouvelle page doit s'afficher.
6. Cliquez sur le lien **My App** pour vérifier que le fichier apparaît dans la liste des répertoires. Cliquez sur le lien **index.html** pour afficher votre page d'accueil.

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 cours précédents, 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, `index.html`, dans l'adresse URL. Dans ce cours, 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 Web 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 **Edit**, situé sous cette liste.

5. Dans la page de modification d'application de MyApp, tapez `index.html` dans la zone de l'URL de la page d'accueil.
6. Cliquez sur le bouton **Save** pour enregistrer les modifications effectuées.
7. Cliquez sur l'onglet principal **Applications**, onglet du dossier situé dans la partie supérieure 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 fait en sorte que l'application s'ouvre dans cette page, même si l'utilisateur n'indique pas cette dernière dans l'adresse URL. 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, `index.html`.

Dans les prochains cours, vous commencerez à créer des 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

Ce cours permet de créer une vue analytique DB2 Alphablox qui affiche une grille et un graphique dans un composant PresentBlox. Pour la première vue analytique à créer, vous allez utiliser l'application DB2 Alphablox Query Builder afin de générer une balise personnalisée JSP qui définit un composant PresentBlox.

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 DB2 Alphablox 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. Connectez-vous à la base de données.
 - a. Cliquez sur le bouton **Connection Settings**. La fenêtre de connexion de base de données (Database Connection) s'ouvre.
 - b. Sélectionnez la source de données `qcc2003-acs`. Cliquez sur le bouton **Connect**. La fenêtre se ferme et la mention "Connected" doit s'afficher dans la section Database Status de la base de données de la page Query Builder.
- Remarque :** Il n'est pas nécessaire de renseigner les zones **Schema**, **Catalog**, **Username** et **Password** car leurs valeurs sont extraites automatiquement de la définition de la source de données sélectionnée.
- c. Veillez à ce que la case à cocher **Execute Default Query** soit désélectionnée.
4. Spécifiez la requête.
 - a. Recherchez la section de la requête (Query). Elle se trouve plus bas, dans la page. Prenez connaissance des informations suivantes :
 - Pour ce tutoriel, vous utilisez un cube DB2 Alphablox (`qcc2003-acs`). La section de la requête contient une liste de cubes disponibles (en cours d'exécution). 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 [nomSourcededonnées]").

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

- Si des données s’affichent dans le composant PresentBlox, en bas de page, vous pouvez 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 **Execute Query**. L’ensemble de résultats s’affiche dans le composant PresentBlox, en bas de page.

- b. Entrez qcc2003-acsc entre crochets (par exemple, select from [qcc2003-acsc]) puis cliquez sur le bouton **Execute Query**.

La section PresentBlox doit afficher une vue par défaut créée en fonction de la requête.

5. Utilisez l’interface utilisateur PresentBlox pour créer une vue et une requête.
 - a. Créez une vue.

Par exemple, vous pouvez déplacer la dimension Products vers l’axe des lignes dans le panneau de présentation des données se trouvant sur la gauche. Vous pouvez ensuite déplacer la dimension Time (Calendar) vers l’axe des colonnes. Vous aurez ensuite la possibilité de déplacer la dimension Measures vers l’axe de la page pour qu’un menu de mesures s’affiche dans le panneau Page. Celui-ci affiche des filtres de page au-dessus du panneau d’affichage des grilles et des graphiques. Sélectionnez deux autres dimensions et ajoutez-les à l’axe des pages afin que trois filtres de page s’affichent au dessus.

- b. Notez l’instruction de requête qui s’affiche dans la section Query. Si la requête n’est pas générée dynamiquement (l’option **Automatically Update Queries** n’est pas sélectionnée), vous pouvez cliquer sur le bouton **Get Current Query** 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 **Query** à utiliser lors du développement d’applications DB2 Alphablox.

6. Cliquez sur le bouton **Generate Blox Tag**. Une fenêtre comportant la balise (et les balises imbriquées) nécessaire à la reproduction de l’agencement PresentBlox et de l’ensemble de résultats s’affiche.

7. Copiez la nouvelle balise dans un nouveau fichier.

- a. Ouvrez un nouveau fichier dans votre éditeur de texte.
- b. Insérez le texte suivant dans le fichier.

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

```
//Copiez et collez les informations de balise provenant de la balise blox
//générée sous cette ligne.
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>

```

- c. Remplacez les lignes commentées par le code affiché à l'étape 6.
 - d. Enregistrez le fichier en lui attribuant le nom PresentBloxView.jsp.
8. Copiez le fichier enregistré (PresentBloxView.jsp) dans le répertoire MyApp du serveur et testez votre nouvelle vue analytique en cliquant sur le lien de la Vue PresentBlox simple, dans 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'affiche correctement, vous devez supprimer les sauts de ligne des valeurs d'attribut (les attributs query et selectableSlicerDimensions ci-dessus, par exemple).

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.

La rubrique suivante décrit les principaux éléments du code de cette page.

Structure de votre première page de vue analytique

Cette rubrique récapitule la structure de la première vue analytique (PresentBloxView.jsp) créée lors du précédent cours.

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.

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

Le fichier 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 blox de l'attribut prefix indique au serveur de traiter le contenu des balises de cette page commençant par blox, en utilisant 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-ac"
    onErrorClearResultSet="true"
    query="SELECT DISTINCT({[qcc_2003].[Time (Calendar)].[All Time
(Calendar)], [qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2000],
[qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2001],
[qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2002],
[qcc_2003].[Time (Calendar)].[All Time (Calendar)].[2003]} )
ON AXIS(0), DISTINCT( {[qcc_2003].[Products].[All Products],
[qcc_2003].[Products].[All Products].[100 Truffles],
[qcc_2003].[Products].[All Products].[200 Chocolate Blocks],
[qcc_2003].[Products].[All Products].[300 Chocolate Nuts],
[qcc_2003].[Products].[All Products].[400 Specialties]} )
ON AXIS(1) FROM [qcc_2003] WHERE ([qcc_2003].[Measures].[Sales],
[qcc_2003].[Time (Fiscal)].[All Time (Fiscal)],
[qcc_2003].[Date Opened].[All Date Opened],
[qcc_2003].[Has Nuts].[All Has Nuts],
[qcc_2003].[Chocolate Type].[All Chocolate Type],
[qcc_2003].[Ounces Per Package].[All Ounces Per Package],
[qcc_2003].[Pieces Per Package].[All Pieces Per Package],
[qcc_2003].[Date Introduced].[All Date Introduced],
[qcc_2003].[Seasonal].[All Seasonal],
[qcc_2003].[Scenario].[Actual],
[qcc_2003].[Locations].[All Locations]} )"
    selectableSlicerDimensions="[qcc_2003].[Measures],
[qcc_2003].[Locations],[qcc_2003].[Chocolate Type]"
    useAliases="true"/>
  <blox:toolbar/>
  <blox:dataLayout/>
  <bloxui:calculationEditor />
</blox:present>
```

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 peut comporter de nombreux attributs. Toutefois, vous ne devez les inclure dans les balises que si vous avez besoin de définir des valeurs d'attribut différentes de celles par défaut. 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 qcc2003-ac. 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 Cube, vous pouvez simplifier la requête pour la rendre

plus courte. 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 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 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.

La section <body> de la page contient la ligne suivante :

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

Cette balise Blox génère le composant PresentBlox défini ci-dessus et rendu ici. La balise <blox:display> permet d'apporter des modifications à un composant Blox ou aux composants imbriqués à un seul emplacement. D'autres balises Blox et scriptlets JSP contrôlant le comportement des composants Blox peuvent aussi figurer à cet endroit. Vous pouvez choisir d'avoir un composant DataBlox distinct du composant 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 des applications et des vues plus complexes, des composants JavaBeans et du code Java supplémentaires peuvent être requis.

Création d'une vue analytique personnalisée

Ce cours vous permet de créer une vue analytique personnalisée contenant un seul graphique avec filtres de page.

Dans la plupart des applications, les données sont représentées sous la forme d'une grille ou d'un graphique. Le composant PresentBlox associe un élément GridBlox et ChartBlox en tant que composants imbriqués s'affichant simultanément dans le composant PresentBlox, mais vous pouvez désactiver ou masquer les composants imbriqués qui ne doivent pas être visibles. Pour utiliser un composant PresentBlox pour la création d'une vue graphique avec des filtres de page, modifiez 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 :

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

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

3. Modifiez la balise PageBlox (<blox:page>), en ajoutant l'attribut 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 comporter un identificateur unique afin d'éviter d'éventuelles erreurs.
5. Enregistrez les modifications et ajoutez le fichier au répertoire MyApp, sur le serveur.
6. Ouvrez votre navigateur et testez la nouvelle page, en cliquant sur le lien présent sur la page d'accueil ou en accédant directement à la page à l'aide de l'adresse URL (par exemple, <http://hôtelocal:9080/MyApp/ChartBloxView.jsp>)

Vous venez de créer 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 de ses composants imbriqués.

Récapitulatif

Si vous avez suivi tous les cours du tutoriel, vous savez comment créer une application DB2 Alphablox élémentaire à l'aide des pages d'administration DB2, de l'application DB2 Alphablox Query Builder et de la bibliothèque de balises Blox. Un grand nombre de propriétés et de méthodes sont disponibles pour la définition et la manipulation de composants 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 rapidement créer des instructions de requête utilisables et des balises Blox.

Chapitre 3. 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. Ecrire 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 **Next**. 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 **Finish**.

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 **Add Portlets** dans la page d'agencement du portail.
4. Entrez DB2 dans la zone de recherche, puis cliquez sur l'option **Search**. La case à cocher **DB2 Alphablox JSP Page Sample Portlet** apparaît.
5. Sélectionnez cette case à cocher et cliquez sur **OK**.
6. Cliquez sur **Done**.

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 :

```
<%@ page contentType="text/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 3, «Tutoriel : Création de votre premier portlet contenant des composants Blox», à la page 25 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 25.

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 tenant compte de la casse (lettre majuscule "V" figurant dans ces deux noms). Vérifiez également que les attributs ont été ajoutés avant le chevron 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 descripteurs des bibliothèques de balises de DB2 Alphablox sont recopiés 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 (cela s'applique uniquement à DB2 Alphablox version 8.4, les utilisateurs de la version 8.4.1 doivent passer à l'étape 2) :

Important : Etant donné que la version 8.4.1 prend en charge Servlet 2.4 au lieu de Servlet 2.2, votre serveur d'applications reconnaîtra automatiquement les fichiers descripteurs des bibliothèques de balises. Si vous utilisez la version 8.4.1, vous ne devez pas référencer explicitement les fichiers descripteurs de la bibliothèque de balises dans le fichier web.xml de votre projet.

```

<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 descripteurs des bibliothèques de balises DB2 Alphablox dans votre projet.
 - Pour la version 8.4, copiez les fichiers descripteurs des bibliothèques de balises DB2 Alphablox (fichiers TLD) de *db2alphablox_dir/bin* vers le répertoire WEB-INF/tlds de votre projet.
 - Pour la version 8.4.1, copiez le fichier aastaglibs.jar de la bibliothèque de balises DB2 Alphablox de *db2alphablox_dir/lib* vers le répertoire WEB-INF/lib de votre projet.

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 :

1. 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 **Show Advanced >>** .
2. 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>
```

3. Pour DB2 Alphablox version 8.4 uniquement (les utilisateurs de la version 8.4.1 doivent passer à l'étape 4), ajoutez les bibliothèques de balises Alphablox au fichier web.xml de votre projet.

Important : Etant donné que la version 8.4.1 prend en charge Servlet 2.4 au lieu de Servlet 2.2, votre serveur d'applications reconnaîtra automatiquement les fichiers descripteurs des bibliothèques de balises. Si vous utilisez la version 8.4.1, vous ne devez pas référencer explicitement les fichiers descripteurs des bibliothèques de balises dans le fichier de votre projet.

- a. Une fois le fichier web.xml ouvert, cliquez sur l'onglet **Variables**.
- b. Accédez à la section consacrée aux références des bibliothèques de balises et située au bas de l'onglet.
- c. Cliquez sur le bouton d'ajout (**Add**).
- d. Pour les zones **URL** et **Location**, 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 **Finish** pour ajouter la référence de bibliothèque de balises, puis sur **Add** pour ajouter la paire suivante.

4. Copiez les fichiers descripteurs des bibliothèques de balises DB2 Alphablox dans votre projet.
 - Pour la version 8.4, copiez les fichiers descripteurs des bibliothèques de balises DB2 Alphablox (fichiers TLD) de *db2alphablox_dir/bin* vers le répertoire WEB-INF/tlds de votre projet.
 - Pour la version 8.4.1, copiez le fichier aastaglibs.jar des bibliothèques de balises DB2 Alphablox de *db2alphablox_dir/lib* vers le répertoire WEB-INF/lib de votre projet.

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 *Guide d'administration*. 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, par exemple "50%" ou "100%". 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 *Guide d'administration* 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 4. 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, 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.

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 intitulé UpdateSite.
3. Copiez le répertoire UpdateSite dans 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 **Help > Software Updates > Find and Install**.
6. Dans la fenêtre d'installation qui s'affiche, sélectionnez l'option **Search for new features to install**, puis cliquez sur **Next**.
7. Dans la fenêtre de mise à jour des sites à visiter, cliquez sur le bouton **New Local Site**, puis accédez à l'emplacement du répertoire UpdateSite.
8. Cliquez sur **Next**, sélectionnez la fonction **DB2 Alphablox Toolkit**, puis cliquez une nouvelle fois sur **Next**.
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 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 Installation de la documentation *DB2 Alphablox - Guide d'installation*. Tenez compte des exceptions suivantes :

1. Dans la fenêtre de configuration de WebSphere du programme d'installation, spécifiez dans la zone **WebSphere Root Directory** l'emplacement de 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 - Guide d'installation*.

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 **Servers** (s'il n'est pas visible, sélectionnez **Window > Show View > Servers**).
3. Cliquez à l'aide du bouton droit de la souris dans la fenêtre de la vue et sélectionnez **New > Server**. La fenêtre du nouveau serveur apparaît.
4. Entrez `hôte\local` comme nom d'hôte et sélectionnez le type de serveur. Cliquez sur **Next**.
5. Saisissez le numéro de port du serveur, puis cliquez sur **Next**.
6. Ajoutez à la liste du projet configuré les projets disponibles que vous voulez exécuter dans cette instance de serveur. Cliquez sur **Finish**. La nouvelle instance de serveur apparaît dans la vue **Servers**.

Si vous avez créé une instance de serveur WebSphere 5, vous devez effectuer l'étape suivante, "Création de variables de substitution de serveur WebSphere 5.". 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

Dans cette procédure, vous allez modifier les variables de substitution de serveur pour les instances du serveur WebSphere 5 afin d'exécuter correctement DB2 Alphablox dans les environnements de test intégrés WebSphere.

1. Ouvrez la vue **Servers**, puis cliquez deux fois sur l'instance de serveur WebSphere 5 à modifier.
2. Cliquez sur l'onglet **Configuration** de l'instance de serveur.
3. Sélectionnez l'option **Enable administration console**, puis désélectionnez l'option **Enable universal test client**.
4. Cliquez sur l'onglet **Variables** de l'instance de serveur. La fenêtre **Substitution Variables** apparaît.
5. Ajoutez deux nouvelles variables à l'aide du bouton **Add** (situé à proximité de la liste des paramètres de noeud).
 - a. Ajoutez une variable intitulée `WS_EAR_AlphabloxPlatform` et attribuez-lui la valeur `$(RACINE_INSTALL_APP)/hôtelocal/AlphabloxPlatform.ear`.
 - b. Ajoutez une deuxième variable intitulée `WS_EAR_AlphabloxStudio` et attribuez-lui la valeur `$(RACINE_INSTALL_APP)/hôtelocal/ApplicationStudio.ear`
6. Enregistrez vos modifications.

Pour les instances de serveur WebSphere 5, vous devez suivre les étapes de la section "Configuration d'instances de serveur WebSphere 5" pour mener à terme la modification de l'instance de serveur.

Configuration d'instances de serveur WebSphere 5

Vous devez configurer les instances de serveur WebSphere 5 pour l'exécution des fichiers et des applications DB2 Alphablox.

Conditions préalables : créez les variables de substitution de serveur WebSphere 5 nécessaires.

Pour configurer les instances de serveur WebSphere 5 :

1. Ouvrez la vue **Servers** 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 **Run Administrative Console**.
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 **Enterprise Applications**, puis cliquez sur le nom d'application **ApplicationStudio**.
6. Attribuez la valeur `$(WS_EAR_ApplicationStudio)` aux fichiers binaires de l'application.
7. Sélectionnez **Use Metadata from Binaries** (Utiliser les métadonnées des fichiers binaires).
8. Cliquez sur le bouton **OK** situé sous la table de configuration.
9. Cliquez sur le bouton **Save** de la console d'administration WebSphere.
10. Dans la fenêtre de sauvegarde dans la configuration maîtresse, cliquez sur **Save**.

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

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. Sur la console telnet, tapez `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 **File > New project**. L'assistant **New project** s'ouvre.
2. Développez l'option **Web**, sélectionnez **Dynamic Web Project**, puis cliquez sur **Next**.
3. Saisissez le nom de votre projet et cliquez sur le bouton **Show Advanced**. Des options supplémentaires apparaissent.
4. Sélectionnez la version de servlet et le serveur cible appropriés.
5. Cliquez sur **Next**. La fenêtre des options des fonctions apparaît.
6. Sélectionnez l'option **DB2 Alphablox Content**, puis cliquez sur **Finish**.

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 **File > New > JSP File**.
2. Tapez un nom de fichier dans la zone **File Name**.
3. Cliquez sur l'option de configuration des options avancées, puis sur **Next**.
4. Ajoutez les bibliothèques de balises DB2 Alphablox à utiliser dans votre fichier JSP.
 - a. Cliquez sur le bouton **Add** 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 **Next**.
5. Sélectionnez **ISO 10646/Unicode(UTF-8)** dans la liste **Encoding**. Pour que les applications DB2 Alphablox s'exécutent correctement, vous devez utiliser le codage UTF-8.
6. Cliquez sur **Finish**. 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 à ce qu'il figure après les 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 des fichiers JavaScript et CSS 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, 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. Configurez les instances du serveur WebSphere 5. 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 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. A l'aide d'un éditeur de texte, créez un document texte.
2. Ajoutez une ligne de code pour appeler le fichier aassetup.bat de DB2 Alphablox, situé dans le répertoire d'installation de 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 RAD rationalsdp.exe 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 5. 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 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, utilisable pour explorer les fonctions du serveur de cubes DB2 Alphablox. La source de données qui en résulte peut 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 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 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 **Data Sources**.
4. Cliquez sur le bouton **Create**.
5. Dans le menu **Adapter**, 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, saisissez 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 **Use DB2 Alphablox Username and Password** (Utiliser le nom d'utilisateur et le 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 *Guide d'administration*.

9. Les valeurs correspondant au **Nombre maximal de lignes** et au **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 **JDBC Tracing Enabled (Traçage JDBC activé)**, sauf si vous voulez consigner les données de journalisation JDBC 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 **Save** 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 **Data Sources**.
4. Cliquez sur le bouton **Create**.

5. Dans le menu **Adapter**, sélectionnez l'option **Alphablox Cube Server Adapter**.
6. Dans la zone du nom de la source de données, saisissez DB2AlphabloxCubes comme nom à utiliser pour la source de données.
7. Cliquez sur le bouton **Save** pour enregistrer la définition de votre 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

Cette procédure permet de définir un cube DB2 Alphablox à partir des métadonnées d'un cube CVSAMPLE DB2 Cube Views.

Conditions préalables : définissez la source de données relationnelles DB2. Définissez une source de données de carte 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 **Create**. La fenêtre Cube Administration s'ouvre.
5. Définissez le nouveau cube :
 - a. Dans la zone du nom du cube DB2 Alphablox, saisissez CVSales.
 - b. Sélectionnez l'option **Enabled**, située à proximité de la zone du nom du cube DB2 Alphablox. Si vous sélectionnez cette option, votre cube démarre automatiquement lors du redémarrage de votre serveur.
 - c. Dans le menu **Relational Data Source**, 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 **Enable DB2 Cube Views Settings**.
 - b. Dans le menu du modèle de cube, sélectionnez **CVSAMPLE.Sales**.
 - c. Dans le menu **Cube**, sélectionnez le cube **General Sales (Ventes générales)**.
 - d. Sélectionnez le bouton **Use Business Names** pour spécifier les noms. Si vous sélectionnez cette option, les noms de membre généralement plus significatifs et plus lisibles sont utilisés.
 - e. Cliquez sur le bouton **Import Cube Definition**. 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 **Show Import Log** 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 **Import cube definition on start, rebuild, and edit**. Si vous sélectionnez cette option, votre cube DB2 Alphablox utilisera la dernière définition des vues de cube DB2 à chaque démarrage, reconstruction ou ouverture pour modification du cube DB2 Alphablox. 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 **Save** pour enregistrer la définition du 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 à l'aide des pages d'administration de DB2 Alphablox.

Conditions préalables : définissez la source de données relationnelles DB2. Définissez une source de données de carte 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 **Start**. Lorsque le cube démarre, la zone d'état affiche **Running**.

Vous disposez à présent d'un cube exempleDB2 Alphablox 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 MDX 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-CONTREFAÇON 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 :

| | | |
|----------------------------|----------|-----------------|
| Cube Views | DB2 | DB2 OLAP Server |
| DB2 Universal Database IBM | Informix | Rational |
| 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.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Index

D

- données exemple
 - création de la définition du cube QCC, référentiel basé sur un fichier 7
 - création de la définition du cube QCC, référentiel de base de données 7
 - définition de la source de données QCC 13
- installation
 - DB2 2
 - Derby 3
 - Hyperion Essbase 6.5 et DB2 OLAP Server 8.1 8
 - Hyperion Essbase 7.1 et DB2 OLAP Server 8.2 10
 - Informix 3
 - Microsoft SQL Server 4
 - Microsoft SQL Server 2000 Analysis Services 11
 - Microsoft SQL Server 2005 Analysis Services 12
 - Oracle 5
 - Sybase 5
- installation des tables QCC 2
- multidimensionnelles 8
- présentation des données exemple de QCC 1

P

- portlet
 - exemple 25
 - structure JSP 26

I

- installation
 - données exemple
 - création de la définition du cube QCC, référentiel basé sur un fichier 7
 - création de la définition du cube QCC, référentiel de base de données 7
 - DB2 2
 - définition de la source de données QCC 13
 - Derby 3
 - Hyperion Essbase 6.5 et DB2 OLAP Server 8.1 8
 - Hyperion Essbase 7.1 et DB2 OLAP Server 8.2 10
 - Informix 3
 - installation des tables QCC 2
 - Microsoft SQL Server 4
 - Microsoft SQL Server 2000 Analysis Services 11
 - Microsoft SQL Server 2005 Analysis Services 12
 - multidimensionnelles 8
 - Oracle 5
 - Sybase 5



Numéro de programme : 5724-L14

GC11-2396-02

