

IBM DB2 Alphablox



Guide d'initiation

Version 8.3

IBM DB2 Alphablox



Guide d'initiation

Version 8.3

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 31.

Première édition - novembre 2005

Réf. US : GC18-9607-00

LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT". IBM DECLINE TOUTE RESPONSABILITE, EXPRESSE OU IMPLICITE, RELATIVE AUX INFORMATIONS QUI Y SONT CONTENUES, Y COMPRIS EN CE QUI CONCERNE LES GARANTIES DE QUALITE MARCHANDE OU D'ADAPTATION A VOS BESOINS. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

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 2005. Tous droits réservés.

© **Copyright International Business Machines Corporation 1996, 2005. All rights reserved.**

Table des matières

Avis aux lecteurs canadiens	v	Préparation de votre environnement de développement	19
Chapitre 1. Tutoriel : Création de votre première application	1	Installation du DB2 Alphablox Toolkit	19
Définition de votre application	1	Installation de DB2 Alphablox dans les environnements de test intégrés de WebSphere	20
Accès aux données	2	Création d'une instance de serveur WebSphere	21
Création de votre page d'accueil d'application	2	Création de variables de substitution de serveur WebSphere 5.1	22
Définition d'une page d'accueil par défaut	3	Configuration d'instances de serveur WebSphere 5.1	22
Création de votre première vue analytique	3	Ajout de l'utilisateur invité au groupe des administrateurs de DB2 Alphablox	23
Structure de votre première page de vue analytique	4	Création d'applications DB2 Alphablox	23
Création de votre deuxième vue analytique	6	Création de fichiers JSP avec du contenu DB2 Alphablox	24
Récapitulatif	7	Accès aux sources de données DB2 OLAP Server et Essbase	24
Chapitre 2. Tutoriel : Création de votre premier portlet contenant des composants Blox.	9	Chapitre 4. Tutoriel : Création d'un cube DB2 Alphablox à l'aide de DB2 Cube Views.	27
Installation des exemples de portlets	9	Définition d'une source de données relationnelles DB2	27
Exécution des exemples de portlets	10	Définition d'une source de données d'adaptateur de serveur de cube Alphablox	28
Étude de la structure d'une page JSP de portlet contenant un composant Blox	11	Définition d'un cube DB2 Alphablox	29
Création de votre propre page JSP de portlet contenant des composants Blox.	13	Démarrage de votre cube DB2 Alphablox	30
Création d'un projet de type portlet afin d'utiliser des composants Blox	14	Remarques	31
Configuration d'un projet de type portlet à l'aide de Rational Application Developer	15	Marques	33
Étapes suivantes.	16	Index	35
Conseils relatifs au développement de portlets	17		
Chapitre 3. Tutoriel : Création d'applications à l'aide des outils Rational Developer	19		

Avis aux lecteurs canadiens

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

Illustrations

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

Terminologie

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

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

Claviers

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

OS/2 et Windows - Paramètres canadiens

Au Canada, on utilise :

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

Nomenclature

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

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

Brevets

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

Assistance téléphonique

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

Chapitre 1. Tutoriel : Création de votre première application

Retrouvez dans ce tutoriel les principales informations de base sur les composants Blox, ainsi que les étapes à suivre pour créer rapidement une application DB2 Alphablox.

Pour créer des applications DB2 Alphablox, vous devez utiliser la technologie JSP (JavaServer Pages) ; il n'est toutefois pas nécessaire ici d'avoir des connaissances dans cette technologie. Se reposant sur vos compétences acquises auparavant dans les technologies Web, ce tutoriel expose de manière simple les aspects de la technologie JSP nécessaires à la création d'applications analytiques.

Les différentes étapes de ce tutoriel supposent que vous utilisiez DB2 Alphablox sur un serveur d'applications WebSphere. 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.

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 vos application et dossier :

1. Créez une application appelée MyApp dans DB2 Alphablox en suivant la procédure décrite à la section Application Definitions du manuel *Administrator's Guide*.
 - Entrez MyApp dans la zone **Name**.
 - Entrez My App (avec un espace) dans la zone **Display Name**, qui définit le libellé de l'application tel qu'il apparaît dans la liste de la page Applications.
2. Cliquez sur l'onglet **Applications** situé dans l'angle supérieur gauche de la page. La liste des applications disponibles apparaît.
3. Cliquez sur le nom de l'application nouvellement créée (My App) dans la liste des applications. Etant donné que vous n'avez créé encore aucun fichier, le répertoire des fichiers est vide.
4. Accédez au dossier de la nouvelle application situé sur votre serveur d'applications (pour WebSphere, le dossier se trouve dans le répertoire `installedApps` de WebSphere). L'application comprend un répertoire WEB-INF contenant un fichier `web.xml`, ainsi que le répertoire `tlds`. Le fichier `web.xml` définit les informations de l'application, tandis que le répertoire `tlds` contient le fichier descripteur des bibliothèques de balises Blox (`blox.tld`), qui définit les balises Blox à utiliser pour la création de vues analytiques dans ce tutoriel. Le répertoire contient d'autres fichiers descripteurs de bibliothèques de balises (TLD) associés à d'autres bibliothèques de balises Blox (non utilisées dans ce tutoriel), notamment `bloxform.tld`, `bloxlogic.tld`, `bloxreport.tld` et `bloxui.tld`.

Votre première application DB2 Alphablox est à présent définie.

Accès aux données

Pour les besoins de ce tutoriel, vous utiliserez conjointement une source de données prédéfinie et DB2 Alphablox.

Pour ouvrir la page des sources de données afin de vous assurer de la disponibilité de cette source de données :

1. Ouvrez votre navigateur et affichez la page d'accueil de DB2 Alphablox.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur **Data Sources**.
4. Sur la partie gauche de la page des sources de données, la source de données Canned s'affiche dans la sélection. La source de données Canned est une source de données prédéfinie qui utilise l'adaptateur de données intégré Canned. La source de données Canned s'avère utile pour les tests et l'apprentissage des concepts de base.

Création de votre page d'accueil d'application

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

Pour créer une page d'accueil :

1. Créez votre fichier de page d'accueil et nommez-le `myapp.html`.
2. Modifiez le fichier et insérez le titre de l'application (par exemple, "Mon application DB2 Alphablox") et deux liens pointant vers les deux vues analytiques que vous créez.

Vous pouvez copier-coller le code suivant dans votre fichier `myapp.html`.

```
<html>
<head>
<title>Mon application DB2 Alphablox</title>
</head>
<body>
<h2>Mon application DB2 Alphablox</h2>
<p>
<a href="GridandChartViews.jsp">Vues GridBlox et ChartBlox</a>
</p>
<p>
<a href="PresentBloxView.jsp">Vue PresentBlox</a>
</p>
</body>
</html>
```

3. Enregistrez ce fichier dans le répertoire MyApp créé lors de la définition de l'application.
4. Ouvrez votre navigateur Web et affichez la page d'accueil de DB2 Alphablox. Par défaut, le navigateur affiche l'onglet **Applications**.
5. Cliquez sur le lien **My App** pour vérifier que le fichier figure bien dans la liste des répertoires. Cliquez sur le lien **myapp.html** pour visualiser votre page d'accueil. Si vous utilisez BEA WebLogic et qu'une erreur HTTP 403 se produit (car les listes de répertoires ne sont pas autorisées par défaut), vous devrez ajouter le nom du fichier `myapp.html` à la fin de l'adresse URL.

Vous venez de créer une page d'accueil comportant des liens vers les deux vues analytiques que vous allez créer dans un prochain tutoriel.

Définition d'une page d'accueil par défaut

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

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

1. Affichez la page d'accueil de DB2 Alphablox dans votre navigateur.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Applications**.
4. Sélectionnez **MyApp** dans la liste des applications, puis cliquez sur le bouton **Edit** situé sous cette liste.
5. Dans la page Edit Application de MyApp, entrez `MyApp.html` dans la zone **Home URL**.
6. Cliquez sur le bouton **Save** pour enregistrer les modifications effectuées.
7. Cliquez sur l'onglet principal Applications, l'onglet du dossier situé à l'extrémité gauche, pour revenir à la page Applications.
8. Cliquez sur le nom d'application **My App**. L'application s'ouvre désormais directement dans la page d'accueil définie.

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

Dans les procédures suivantes, vous commencerez à créer vos premières vues analytiques DB2 Alphablox à l'aide de balises Blox simples.

Création de votre première vue analytique

Cette procédure vous permet de créer une vue analytique DB2 Alphablox affichant une grille et un graphique.

Pour votre première vue, vous utiliserez les balises personnalisées JSP définies par DB2 Alphablox, afin de créer rapidement une page JSP comprenant deux composants Blox de présentation. Les composants GridBlox et ChartBlox que vous utiliserez permettent de créer deux vues différentes des données d'exemple : une vue de table de grille et une vue graphique.

Conditions préalables : si vous avez pris connaissance de la section d'introduction à DB2 Alphablox et ses composants, vous devez savoir ce que sont les Blox, et comment fonctionnent les pages JSP et les balises personnalisées. Si vous n'avez pas traité cette section et si vous n'êtes pas très familiarisé avec J2EE et la technologie JSP (JavaServer Pages), commencez par lire cette section d'introduction. La technologie JSP et les balises personnalisées JSP permettent d'insérer dans des pages HTML du contenu dynamique traité sur le serveur plutôt que dans le navigateur Web de l'utilisateur.

Pour créer votre première page d'analyse DB2 Alphablox, ajoutez à une page JSP deux des trois Blox de présentation les plus courants :

1. Ouvrez un nouveau fichier dans votre éditeur.

2. Créez une page de vue analytique élémentaire assortie d'un titre, d'un GridBlox et d'un ChartBlox en saisissant le code suivant :

```
<%@ taglib uri="bloxtld" prefix="blox" %>
<html>
<head>
<blox:header/>
</head>
<body>
<h2>Vues GridBlox et ChartBlox</h2>
<p>
<blox:grid id="MyGridBlox"
  height="50%"
  width="90%">
  <blox:data
    dataSourceName="Canned"/>
</blox:grid>
</p>
<p>
<blox:chart id="MyChartBlox"
  height="50%"
  width="90%"
  chartType="Bar">
  <blox:data
    dataSourceName="Canned"/>
</blox:chart>
</p>
</body>
</html>
```

3. Nommez le fichier GridandChartViews.jsp et enregistrez-le dans le répertoire MyApp.
4. Vérifiez les résultats obtenus.
 - a. Ouvrez la page d'accueil de DB2 Alphablox dans votre navigateur. Si vous ne vous trouvez pas déjà dans la page Applications, cliquez sur l'onglet **Applications**.
 - b. Cliquez sur le nom d'application **My App** pour ouvrir la page d'accueil.
 - c. Cliquez sur le lien des **Vues GridBlox et ChartBlox** pointant vers la page GridAndChartViews.jsp pour afficher votre première vue DB2 Alphablox. Si une page d'erreur apparaît, vérifiez que votre code ne contient pas d'erreurs, puis réessayez.

Structure de votre première page de vue analytique

Cette rubrique récapitule la structure de la première vue analytique créée lors de l'étape précédente.

La première ligne de la page est la suivante :

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

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

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

Dans le corps de la page, les lignes ci-après définissent le premier Blox de présentation, un GridBlox :

```
<blox:grid id="MyGridBlox"
  height="50%"
  width="90%">
  <blox:data
    dataSourceName="Canned"/>
</blox:grid>
```

La balise `<blox:grid>` indique que vous souhaitez qu'un GridBlox dont la valeur d'attribut `id` est `MyGridBlox` soit affiché ici. L'attribut `id` vous permet d'identifier ce Blox particulier en vue d'une utilisation dans des scripts. Chaque Blox défini à l'aide de balises comporte de nombreux attributs. Toutefois, à moins que vous n'ayez besoin de définir des valeurs d'attribut différentes de celles par défaut, il n'est pas nécessaire de les inclure dans les balises. Dans le cas d'une grille, vous avez également indiqué une valeur `height` de 50% et une valeur `width` de 90%. 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 `Canned`. Si aucune source de données n'est spécifiée, le message "Aucune donnée disponible" apparaît dans la grille. La source de données `Canned` est prédéfinie au cours de l'installation. Elle comprend quelques données d'exemple, ne nécessite pas l'installation ni la configuration d'une véritable base de données externe, et peut être utilisée à des fins d'apprentissage et de résolution d'incidents. Comme nous n'avons aucune requête à formuler, il n'existe aucun attribut `Query`.

Les lignes suivantes définissent un ChartBlox à l'aide de la balise `<blox:chart>` :

```
<blox:chart id="MyChartBlox"
  height="50%"
  width="90%"
  chartType="Bar">
  <blox:data
    dataSourceName="Canned"/>
</blox:chart>
```

Comme indiqué dans le précédent GridBlox, les attributs spécifiés sont les suivants : `id`, `height` et `width`. Outre ces attributs communs, un attribut `chartType` dont la valeur est `Bar` (diagramme de Gantt normal) a été défini pour remplacer la valeur par défaut `3D Bar` (diagramme de Gantt tridimensionnel). De même que pour le GridBlox, un DataBlox imbriqué est spécifié à l'aide de la balise `<blox:data>`. Pour pouvoir rendre disponibles les ensembles de résultats, le composant Blox a besoin d'une source de données.

Pour récapituler, trois balises Blox ont été employées sur cette page : `<blox:header/>`, `<blox:grid>` et `<blox:chart>`. Ces trois balises spécifient une vue analytique, sans qu'il soit nécessaire de recourir à du code Java dans la page. La complexité de la logique de présentation est gérée par ces trois simples 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.

Création de votre deuxième vue analytique

Cette procédure vous permet de créer une vue analytique PresentBlox, combinant à la fois une grille et un graphique dans un même composant Blox.

Jusqu'ici, nous avons étudié les composants GridBlox et ChartBlox pour la création de vues analytiques qui ne sont pas synchronisées pour afficher les mêmes données. De nombreuses applications requièrent une vue des données n'utilisant qu'une grille ou un graphique. Dans d'autres circonstances, vous pouvez également afficher ces données dans une vue analytique permettant aux utilisateurs de passer d'une vue sous forme de grille à une vue sous forme de graphique, ou inversement. Vous pouvez aussi permettre aux utilisateurs de visualiser simultanément une grille et un graphique. Le composant PresentBlox combine un composant GridBlox et un composant ChartBlox en tant que composants imbriqués s'affichant simultanément au sein du composant PresentBlox. Dans la procédure suivante, vous allez définir un PresentBlox dans une page JSP et utiliser les propriétés de ce PresentBlox afin de créer une vue qui, au départ, affiche côte à côte une grille et un graphique. A l'aide du contrôle d'interface utilisateur, un utilisateur peut également afficher uniquement une grille ou un graphique. L'association de ces deux vues dans un Blox vous permet non seulement d'obtenir des vues synchronisées des données, mais également d'exploiter au mieux l'espace limité d'une page.

Pour créer votre deuxième vue analytique :

1. Ouvrez un nouveau fichier dans votre éditeur HTML.
2. Créez une page de vue analytique élémentaire assortie d'un titre et d'un unique composant PresentBlox en saisissant le code suivant :

```
<%@ taglib uri="bloxtld" prefix="blox" %>
<html>
<head>
<blox:header/>
</head>
<body>
<h2>Vue PresentBlox</h2>
<p>
<blox:present id="MonPresentBlox"
  height="50%"
  width="90%">
  <blox:data
    dataSourceName="Canned"/>
  <blox:grid
    bandingEnabled="true"/>
  <blox:chart
    chartType="Bar"/>
</blox:present>
</p>
</body>
</html>
```

3. Nommez le fichier PresentBloxView.jsp et enregistrez-le dans le répertoire MyApp.
4. Vérifiez les résultats obtenus :
 - a. Ouvrez la page d'accueil de DB2 Alphablox dans votre navigateur. Si vous ne vous trouvez pas déjà dans la page Applications, cliquez sur l'onglet **Applications**.
 - b. Cliquez sur le nom d'application **My App** pour ouvrir la page d'accueil.

- c. Pour afficher votre nouvelle vue analytique, cliquez sur le lien **PresentBlox View** pointant vers le fichier `PresentBloxView.jsp`. Si une page d'erreur apparaît, vérifiez que votre code ne contient pas d'erreurs, puis réessayez.

Récapitulatif

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

Si vous souhaitez afficher les données de votre entreprise dans une vue DB2 Alphablox, vous savez désormais créer une application de bout en bout et lui ajouter des vues analytiques. Pour afficher des données sur le champ, il vous suffit d'apporter quelques modifications simples à l'application MyApp. Pour afficher les données de vos bases de données d'entreprise plutôt que d'utiliser la source de données Canned, vous devez créer une source de données pointant vers une source de données d'entreprise, modifier les attributs `dataSourceName` pour qu'ils pointent vers cette source de données nouvellement définie, puis ajouter l'attribut `query` approprié. Pour plus de détails sur la création de sources de données, reportez-vous au manuel *Administrator's Guide*. Pour savoir comment créer les requêtes appropriées, consultez la section *Retrieving Data* du *Developer's Guide* et la section *DataBlox* du *Developer's Reference*.

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

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

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

Ce tutoriel ne vous fournit pas de détails généraux sur le développement de portlets. Il s'attache principalement aux tâches propres à DB2 Alphablox et suppose un certain niveau de connaissances dans les 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 de détails sur l'installation, 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 portlets 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 Portlet Management, cliquez sur **Web Modules**. La page Manage Web Modules apparaît sur la droite.
4. Cliquez sur **Install**. Vous êtes invité à préciser le module Web que vous souhaitez installer.
5. Cliquez sur le bouton **Browse**, 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 **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 vise à vérifier la structure de code d'une page JSP contenant un Blox. Toutes les pages JSP doivent disposer des mêmes éléments-clés.

Pour ouvrir un fichier JSP :

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

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

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

<portletAPI:init/>

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

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

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

Ce bloc de code contient six éléments-clés :

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

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

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

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

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

Cette balise permet d'accéder aux objets `PortletRequest`, `PortletResponse` et `PortletConfig`. Avec l'objet `PortletResponse`, vous pouvez appeler la méthode

encodeNamespace() afin de vous assurer que le nom de votre composant Blox n'entre pas en conflit avec d'autres objets situés sur d'autres portlets qui s'exécutent sur la même page.

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

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

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

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

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

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

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

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

- Ce code ajoute un PresentBlox dont l'id est presentBlox et le nom bloxName xx_x_x_xxx_presentBlox ; c'est là le résultat du codage de l'espace de nom.
- Sa largeur est égale à 800 pixels et sa hauteur à 400 pixels (valeurs 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, un message de type **Aucune donnée disponible** apparaît généralement dans la grille. La source de données Canned est prédéfinie lors de l'installation et comprend quelques données d'exemple. Elle ne nécessite pas d'installation ni de configuration d'une véritable base de données externe, et peut être utilisée à des fins d'apprentissage et de résolution d'incidents. Etant donné que nous n'avons pas à indiquer de requête, le code ne contient pas d'attribut query.

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

Vous venez d'étudier la structure du code JSP et d'apprendre le code élémentaire à insérer dans vos pages JSP de portlet. La prochaine procédure va vous permettre

de créer une nouvelle page JSP contenant un autre Blox et de spécifier certains attributs de Blox communs pour que la page s'adapte mieux à votre page de portail.

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

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

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

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

Assurez-vous que les conditions prérequis de la section Chapitre 2, «Tutoriel : Création de votre premier portlet contenant des composants Blox», à la page 9 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 9.

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

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

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

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

6. Désactivez la barre de menus et la barre d'outils situées au sommet en paramétrant les attributs `menubarVisible` et `toolbarVisible` à `false` :

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

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

7. Modifiez l'espace de nom en `myFirstGrid` :

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

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

8. Enregistrez le fichier.

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

Pour tester les modifications que vous venez d'effectuer :

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

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

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

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

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

1. Modifiez le fichier `web.xml` de votre projet pour qu'il comprenne les lignes suivantes :
 - Pour la définition et le mappage de servlets :

```

<servlet>
  <servlet-name>AlphabloxServer</servlet-name>
  <servlet-class>com.alphablox.server.webapps.server.AlphabloxServer
  </servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>AlphabloxServer</servlet-name>
  <url-pattern>/abx/*</url-pattern>
</servlet-mapping>

```

- Pour les références de bibliothèque de balises :

```

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

```

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

```
<rép_db2alphablox>/bin/
```

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

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

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

- Choisissez "WebSphere Portal v5.1 stub" comme serveur cible. Cette option est spécifiée sur le premier écran de l'assistant en cliquant sur le bouton **Show Advanced >>**.

- Ajoutez le mappage de servlet de DB2 Alphablox au fichier web.xml créé sous WebContent/WEB-INF/. Pour cela, saisissez la définition de servlet et le code de mappage suivants dans le fichier web.xml de votre projet :


```
<servlet>
  <servlet-name>AlphabloxServer</servlet-name>
  <servlet-class>com.alphablox.server.webapps.server.AlphabloxServer
    </servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>AlphabloxServer</servlet-name>
  <url-pattern>/abx/*</url-pattern>
</servlet-mapping>
```
- Ajoutez les bibliothèques de balises Alphablox au fichier web.xml de votre projet. Suivez la procédure ci-dessous :
 1. Une fois le fichier web.xml ouvert, cliquez sur l'onglet **Variables**.
 2. Accédez à la section consacrée aux références des bibliothèques de balises et située au bas de l'onglet.
 3. Cliquez sur **Add**.
 4. 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.

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

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

<rép_db2alphablox>/bin/

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

Etapes suivantes

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

1. définir une nouvelle source de données DB2 Alphablox pointant vers votre source de données ;

2. modifier l'attribut `dataSourceName` de `DataBlox` pour le faire pointer vers la source de données définie ;
3. ajouter un attribut `query` avec votre chaîne de requête au `DataBlox`.

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

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

Conseils relatifs au développement de portlets

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

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

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

- Consultez la rubrique `Blox Portlet Tag Library` du *Developer's Guide* pour connaître les problèmes dont vous devrez tenir compte lors de la phase de planification.
- `DB2 Alphablox` possède un puissant modèle d'identification unique des `Blox`. Ce modèle comprend un objet `ClientLink` qui permet de charger une URL spécifiée

lorsque l'utilisateur clique sur un composant Blox de la page. Pour le développement de portlets, utilisez la bibliothèque de balises de portlets de Blox pour la création de ce lien. Ces balises gèrent dynamiquement la formation des adresses URL de manière à ce que les liens soient mis à jour lors de l'actualisation de la page. Pour plus de détails, reportez-vous à la rubrique *Planning for portlet development* du *Developer's Guide*.

- Pour les identificateurs URI des actions et des portlets, utilisez la bibliothèque de balises de portlets de Blox pour la création de liens d'action ou de portlet. Vous pouvez ensuite recourir à l'API de portlet du portail pour le traitement du lien de portlet ou d'action. Pour plus de détails, reportez-vous à la rubrique *Blox Portlet Tag Library* du *Developer's Guide*.

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

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

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 la fonction d'ajout de méthodes et de balises de fin, ainsi que les 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 nécessaires ; vous devez disposer au moins de Rational Application Developer ou Rational Web Developer version 6.0.0.1.
3. Si vous envisagez d'utiliser les environnements de test intégrés de WebSphere 5.1, assurez-vous d'avoir mis à jour votre version de Rational Developer de manière à ce qu'elle intègre cette installation facultative.

Remarque : Les installations par défaut des outils de Rational Developer ne comprennent pas les environnements de test intégrés WebSphere 5.1.

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

Installation du DB2 Alphablox Toolkit

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

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

L'installation de DB2 Alphablox Toolkit dans un outil Rational Developer vous épargne de 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 et 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 l'unité de CD-ROM de votre poste de travail.
2. Dans le répertoire plugin, repérez le sous-répertoire nommé UpdateSite.
3. Copiez le répertoire UpdateSite à un emplacement adéquat de votre disque dur. Par exemple, copiez-le dans C:\DB2Alphablox\UpdateSite.
4. Démarrez l'outil Rational Developer.
5. Dans la barre de menus, sélectionnez **Help > Software Updates > Find and Install**.
6. Dans la fenêtre d'installation qui apparaît, sélectionnez l'option **Search for new features to install**, puis cliquez sur **Next**.
7. Dans la fenêtre Update sites to visit, 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 de nouveau sur **Next**.
9. Dans les fenêtres suivantes, acceptez l'accord de licence, puis sélectionnez l'emplacement dans lequel installer la fonctionnalité DB2 Alphablox Toolkit.
10. Une fois l'installation terminée, redémarrez l'outil Rational Developer.

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

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

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

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

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

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

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

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

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

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

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

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

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

Création d'une instance de serveur WebSphere

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

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

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

1. Ouvrez l'outil Rational Developer.
2. Cliquez sur l'onglet **Servers** (s'il n'est pas visible, sélectionnez **Window > Show View > Servers**).
3. Cliquez à l'aide du bouton droit de la souris à l'aide de la fenêtre de vue, puis sélectionnez **New > Server**. La fenêtre New Server apparaît.
4. Entrez localhost comme nom d'hôte et sélectionnez le type de serveur. Cliquez sur **Next**.
5. Saisissez le numéro de port, 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 5.1, vous devez effectuer l'étape suivante, "Création de variables de substitution de serveur WebSphere 5.1". Pour les instances de serveur WebSphere 6, passez à l'étape "Ajout de l'utilisateur invité au groupe des administrateurs DB2 Alphablox".

Création de variables de substitution de serveur WebSphere 5.1

Dans cette procédure, vous allez modifier les variables de substitution de serveur des instances de serveur WebSphere 5.1 pour exécuter correctement DB2 Alphablox au sein des environnements de test intégrés WebSphere.

1. Ouvrez la vue **Servers**, puis cliquez deux fois sur l'instance de serveur WebSphere 5.1 que vous voulez modifier.
2. Cliquez sur l'onglet **Configuration** de l'instance de serveur.
3. Sélectionnez **Enable administration console**, puis désélectionnez **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 en regard de la liste **Node Settings**).
 - a. Ajoutez une variable nommée `WS_EAR_AlphabloxPlatform`, puis paramétrez sa valeur à `$(APP_INSTALL_ROOT)/localhost/AlphabloxPlatform.ear`.
 - b. Ajoutez une seconde variable nommée `WS_EAR_AlphabloxStudio`, puis paramétrez sa valeur à `$(APP_INSTALL_ROOT)/localhost/ApplicationStudio.ear`
6. Enregistrez vos modifications.

Pour les instances de serveur WebSphere 5.1, vous devez suivre la procédure de la section "Configuration d'instances de serveur WebSphere 5.1" pour terminer les modifications de votre instance de serveur.

Configuration d'instances de serveur WebSphere 5.1

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

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

Pour configurer les instances de serveur WebSphere 5.1 :

1. Ouvrez la vue **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 **Run Administrative Console**.
3. Ne saisissez aucune 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. Paramétrez **Application Binaries** sur `$(WS_EAR_ApplicationStudio)`.
7. Sélectionnez **Use Metadata from Binaries**.
8. Cliquez sur le bouton **OK** situé sous la table de configuration.
9. Cliquez sur le bouton **Save** de la console d'administration de WebSphere.
10. Dans la fenêtre Save to Master Configuration, 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 de DB2 Alphablox à l'aide de la commande Telnet suivante : `telnet Système_hôte_local Numéro_port`, où *Numéro_port* correspond au port spécifié lors de l'installation de DB2 Alphablox.
2. A l'invite de la console Telnet, tapez la commande DB2 Alphablox suivante : `set Administrators guest` et appuyez sur Entrée.
3. Toujours dans la console Telnet, tapez `save` et 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, puis cliquez sur le bouton **Show Advanced**. Des options supplémentaires apparaissent.
4. Sélectionnez le serveur cible et la version de servlet appropriés.
5. Cliquez sur **Next**. La fenêtre Features options 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 désormais 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 **Project Explorer** de Rational Developer, sélectionnez **File > New > JSP File**.
2. Saisissez un nom de fichier dans la zone **File Name**.
3. Cliquez sur **Configure advanced options**, 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 Add Tag Libraries.
 - 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. Positionnez le curseur à l'intérieur de la balise HTML <head> en veillant, cependant, à le placer à la suite des directives taglib JSP de DB2 Alphablox que vous avez ajoutées précédemment.
9. Saisissez la balise d'en-tête Blox en tapant le code suivant sur la nouvelle ligne : <blox:header/> Cette balise est nécessaire pour l'ajout des fichiers CSS et JavaScript de DB2 Alphablox lors de l'exécution du fichier JSP.

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

Accès aux sources de données DB2 OLAP Server et Essbase

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

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

En raison des limites inhérentes aux environnements de test intégrés WebSphere 5.1 de l'outil Rational Developer, vous devez créer un fichier de commandes de démarrage pour que Rational Developer puisse accéder aux sources de données DB2 OLAP Server ou Hyperion Essbase. Lors du démarrage de l'outil Rational

Developper à l'aide du fichier de commandes, les bibliothèques client Essbase requises sont ajoutées dans l'emplacement de bibliothèque Java.

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

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

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

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

Lorsque vous cliquez deux fois sur le fichier startRAD.bat, le fichier aassetup.bat DB2 Alphablox s'exécute et définit les chemins d'accès et les variables d'environnement nécessaires ; Rational Application Developer est ensuite lancé. Les installations de DB2 Alphablox intègrent plusieurs versions des bibliothèques client Essbase pouvant être utilisées avec les sources de données de DB2 OLAP Server. 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 client Essbase utilisées avec DB2 Alphablox, exécutez l'utilitaire DB2 OLAP Server / Essbase Client Library (ChangeEssbase.bat), situé dans le répertoire *db2_alphablox\analytics\bin* (*db2_alphablox* correspondant au répertoire principal de votre installation 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 startRAD.bat pour lancer l'outil Rational Developer.

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

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

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

Conditions préalables :

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

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

Définition d'une source de données relationnelles DB2

Cette procédure vous 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 des 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 **Data Source Name**, entrez CVSAMPLE comme nom de source de données.

7. Entrez les valeurs appropriées dans les zones **Server Name**, **Port Number** et **Database Name** (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. Saisissez un **nom d'utilisateur par défaut** (Default Username) et un **mot de passe par défaut** (Default Password).

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** est ignorée lorsque la source de données relationnelles spécifiée est utilisée pour le remplissage d'un cube DB2 Alphablox. Vous pouvez utiliser les listes de contrôle d'accès pour restreindre l'accès aux cubes DB2 Alphablox. Pour plus d'informations sur les listes de contrôle d'accès, reportez-vous au manuel *Administrator's Guide*.

9. Les valeurs des zones **Maximum Rows** et **Maximum Columns** sont ignorées lorsque la source de données est utilisée pour le remplissage d'un cube DB2 Alphablox. Vous pouvez toujours saisir des valeurs. Néanmoins, elles seront utilisées lorsque d'autres applications exploiteront cette source de données.
10. Paramétrez l'option **JDBC Tracing Enabled** à **No**, sauf si vous voulez consigner les informations de journalisation JDBC dans le fichier journal de DB2 Alphablox. N'activez le traçage JDBC que si vous rencontrez des problèmes et que vous devez effectuer des opérations de débogage pour en déterminer l'origine.
11. Cliquez sur le bouton **Save** pour enregistrer la définition de source de données.

Vous venez de définir la source de données DB2 Alphablox pour CVSAMPLE. Vous pouvez à présent créer une définition de cube DB2 Alphablox afin d'accéder aux métadonnées des cubes Cube Views contenus dans 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 des 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 **Data Source Name**, saisissez CubesDB2A1phab1ox comme nom de 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

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

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

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

1. Connectez-vous aux pages d'administration de DB2 Alphablox en tant qu'administrateur (ou utilisateur disposant des 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 **DB2 Alphablox Cube Name**, entrez CVSa1es.
 - b. Sélectionnez l'option **Enabled**, située en regard de la zone **DB2 Alphablox Cube Name**. Si vous sélectionnez cette option, votre cube démarre automatiquement lors du redémarrage de votre serveur.
 - c. Dans le menu **Relational Data Source**, sélectionnez MyDB2, la source de données relationnelles que vous avez créée pour les besoins du présent tutoriel.
 - d. Veillez à ce que l'option **Security Role** ne soit pas sélectionnée. Cette option sert à limiter le nombre d'utilisateurs susceptibles d'accéder à un cube particulier.
6. Activez les paramètres DB2 Cube Views et indiquez les métadonnées à utiliser :
 - a. Sélectionnez l'option **Enable DB2 Cube Views Settings**.
 - b. Dans le menu **Cube Model**, sélectionnez CVSAMPLE.Sales.
 - c. Dans le menu **Cube**, sélectionnez le cube **General Sales**.
 - d. Sélectionnez le bouton d'option **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 utilise la dernière définition DB2 Cube Views chaque fois qu'il est démarré, reconstitué ou ouvert pour être modifié. Lorsque vous aurez acquis une meilleure maîtrise de DB2 Alphablox et de DB2 Cube Views, vous pourrez importer la définition de cube et la personnaliser.
7. Cliquez sur le bouton **Save** pour enregistrer la définition de cube DB2 Alphablox.

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

Démarrage de votre cube DB2 Alphablox

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

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

Pour démarrer le cube CVSales :

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

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

DB2	DB2 OLAP Server	DB2 Universal Database
IBM	WebSphere	

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

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

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

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

Index

P

portlet
 exemple 9
 structure de page JSP 11

T

tutoriels
 développement d'application 1



Numéro de programme : 5724-L14

SC11-2396-00

