

IBM DB2 Alphablox



Guide d'administration

Version 84

IBM DB2 Alphablox



Guide d'administration

Version 84

Table des matières

Avis aux lecteurs canadiens.	xi
Chapitre 1. Présentation de DB2 Alphablox	1
Présentation de DB2 Alphablox	1
DB2 Alphablox en environnement J2EE	2
Avantages de l'exécution sur un serveur d'applications	2
Configurations des serveurs d'applications WebSphere et WebLogic	3
Configuration d'Apache Tomcat	3
Composants de la plateforme DB2 Alphablox	4
DB2 Alphablox	4
Adaptateurs de données	4
Composants Blox	5
Client DHTML	7
Fonctions d'administration à distance	7
Gestion centralisée des applications	7
Architecture à n-niveaux	8
Applications DB2 Alphablox	8
Application Studio	8
Architecture DB2 Alphablox	9
Gestionnaire de services	9
Gestionnaire de requêtes	9
Gestionnaire de sessions	10
Gestionnaire d'utilisateurs	10
Gestionnaire d'applications	11
Gestionnaire de données	11
Gestionnaire de référentiels	12
Gestionnaire de fichiers	12
Gestionnaire de consoles	12
Gestionnaire de cubes DB2 Alphablox	12
Gestionnaire de clusters	12
Liste des tâches lors de la première utilisation de DB2 Alphablox	13
Chapitre 2. Applications DB2 Alphablox	15
Types d'application DB2 Alphablox	15
Modes de présentation des données	15
Conversion au format PDF	16
Personnalisation des modes de rendu à l'aide de XML	16
Application de création de rapports relationnels	16
Composants d'une application DB2 Alphablox	16
JSP (JavaServer Pages)	16
Composants Blox	17
Chapitre 3. Page d'accueil de DB2 Alphablox	19
Présentation de la page d'accueil de DB2 Alphablox	19
Onglet Applications	19
Autres applications	19
Onglet Administration	20
Autres dispositions	20
General Properties (Propriétés générales)	20
Custom Properties (Propriétés personnalisées)	20
Runtime Management (Gestion de l'exécution)	20
Console	20
Portal Theme Utility (Utilitaire de thèmes de portail)	21
Migration d'applications	21
Groups (Groupes)	21

Users (Utilisateurs)	21
Roles (Rôles)	22
Applications	22
Data Sources (Sources de données)	22
Cubes DB2 Alphablox	22
Onglet Assemblage : Application Studio	22
Plan de travail	22
Exemples	23
Modèles de données	23
Liens dans le coin supérieur droit	23
Mon profil.	23
Aide.	23

Chapitre 4. Tâches d'administration de base 25

Démarrage de DB2 Alphablox	25
Accès à DB2 Alphablox	25
Arrêt de DB2 Alphablox	25
Arrêt de DB2 Alphablox avec le script d'arrêt sur des installations Apache Tomcat	26
Sur des systèmes Windows :	26
Sous Linux et UNIX :	26
Arrêt de DB2 Alphablox à partir des services du panneau de contrôle sur des installations Apache Tomcat	26
Utilisation de l'utilitaire de bibliothèque de client Essbase	26
Autres tâches d'administration et informations	27

Chapitre 5. Administration du client 29

Administration du client DHTML	29
Configuration client DHTML	29
Remarques relatives au client DHTML	29
Feuilles de style en cascade (CSS)	29
Fenêtres en incrustation	29

Chapitre 6. Définitions d'application 31

Définition d'application DB2 Alphablox	31
Noms d'application.	32
Répertoire WEB-INF	32
Définition d'une application	32
Définition d'une application lors de l'exécution avec WebSphere.	34
Création d'une application dans DB2 Alphablox puis enregistrement dans WebSphere	34
Importation d'une application WebSphere existante dans DB2 Alphablox	36
Modification d'une définition d'application existante	36
Suppression d'une définition d'application existante.	36
Définition d'une application lors de l'utilisation de clusters WebLogic.	37
Enregistrement d'applications avec des serveurs Web externes	38
Enregistrement d'applications sur le serveur Apache HTTP Server	38
DB2 Alphablox 8.4.1 sur Apache Tomcat 5.5	38
DB2 Alphablox 8.4 sur Apache Tomcat 3.2.4	38
Arrêt et redémarrage des serveurs Apache HTTP Server	39
Enregistrement d'applications sur des serveurs Web Microsoft IIS (Internet Information Server)	39
Enregistrement d'applications sur les serveurs Web Sun iPlanet sur Alphablox 8.4.	39
Ajout du répertoire de stockage	39
Ajout des affectations appropriées au style DB2 Alphablox	40
Importation d'une application J2EE existante	40

Chapitre 7. Définitions de sources de données 43

Définition d'une source de données	43
Changement ou suppression d'une définition de source de données existante	44
Changement d'une définition de source de données existante.	44
Suppression d'une définition de source de données existante	45
Configuration de l'authentification Microsoft pour les sources de données Microsoft Analysis Services	45
Configuration des droits utilisateur Windows	45

Définition des droits pour l'utilisateur Windows	45
Définition des droits pour l'utilisateur Windows	46
Configuration du service Windows	47
Vérification que les utilisateurs sont configurés dans Microsoft Analysis Services	47
Utilisation des sources de données JDBC	48
Configuration de l'environnement pour le pilote relationnel Sybase JConnect	48
Configuration du traçage JDBC	48
Mise à jour d'un pilote JDBC pris en charge vers une version différente	49
Ajout de pilotes JDBC supplémentaires	49
Modification des paramètres de chemin de classes	50
WebSphere	50
WebLogic	50
Tomcat	50
Chapitre 8. Définitions d'utilisateurs	51
Création d'un nouvel utilisateur	51
Modification ou suppression d'un groupe ou d'un utilisateur existant	52
Modification des propriétés d'un utilisateur existant	52
Suppression d'un utilisateur	53
Modification des groupes auxquels l'utilisateur appartient	53
Chapitre 9. Définitions de groupes	55
Création d'un groupe	55
Présentation des sous-groupes	56
Modification ou suppression d'un groupe	57
Modification d'un groupe	57
Suppression d'un groupe	57
Chapitre 10. Définitions des rôles	59
Définition de nouveaux rôles	59
Modification et suppression de rôles existants	59
Modification des rôles auxquels un utilisateur ou un groupe appartient	60
Suppression d'un rôle existant	60
Chapitre 11. Sécurité et authentification	61
Modes d'authentification et de sécurité de DB2 Alphablox	61
Modèle de sécurité dans Alphablox 8.4.1	62
Module de connexion Alphablox JAAS (Java Authentication and Authorization Service)	62
Configuration du domaine JNDI dans Tomcat 5.5	62
Ajout du domaine JNDI Alphablox à Apache Tomcat 5.5	63
Admin contre droits d'utilisateur	63
Suppression des droits de connexion de l'utilisateur invité pour les applications	63
Applications et domaines de sécurité du serveur d'applications	63
Authentification du serveur Web contre authentification DB2 Alphablox	64
Utilisation des options de sécurité du serveur Web Sun iPlanet sur Alphablox 8.4 avec Apache Tomcat 3.2.4	64
Définition des options de sécurité Microsoft pour IIS NTLM	65
Installation de Microsoft IIS	65
Installation de DB2 Alphablox et sélection de Microsoft IIS comme serveur Web	65
Configuration des paramètres de sécurité dans Microsoft IIS	66
Création d'un utilisateur local Windows nommé admin pour NTLM	68
Configuration de la sécurité NTLM dans Tomcat 5.5 pour Alphablox 8.4.1	68
Connexion à DB2 Alphablox	69
Configuration de DB2 Alphablox pour utiliser la sécurité reposant sur le serveur Web	70
Configuration de la génération automatique des comptes utilisateur	70
Filtrer les adresses IP	71
Définir les droits sur les répertoires	71
Désactiver l'exploration des répertoires	71
Chapitre 12. Extension de DB2 Alphablox	73
Présentation générale	73

Extensions de calcul	73
Extensions du gestionnaire d'utilisateurs	74
Extensions de client DHTML	74
Configuration de DB2 Alphablox pour la prise en charge des classes Java personnalisées	74
Définition du chemin de classes	75

Chapitre 13. Configuration des propriétés de DB2 Alphablox 77

Tâches d'administration de DB2 Alphablox	77
Configuration des propriétés de démarrage.	77
Configuration des propriétés système.	79
Indication du port Telnet	80
Configuration de DB2 Alphablox Cube Manager	81
Définitions des propriétés personnalisées	82
Définition d'une nouvelle propriété utilisateur.	82
Modification d'une propriété utilisateur	83
Suppression d'une propriété utilisateur	83
Définition d'une nouvelle propriété d'application personnalisée	84
Modification d'une propriété d'application	84
Suppression d'une propriété d'application	85
Création et gestion de collections de commentaires	85
Accès à la boîte de dialogue Comments Management	85
Définition et accès à une source de données	86
Définition des collections de commentaires	86
Collections de commentaires utilisant des bases de données Microsoft SQL Server ou Sybase	87
Affichage des définitions des collections de commentaires	88
Suppression de collections de commentaires	88
Ajout et affichage des commentaires	88
Création d'un processeur PDF distant	88
Configuration d'un processeur PDF distant.	88
Configuration de l'administration des rapports PDF distants	89
Fichiers journaux DB2 Alphablox	90
Paramètres d'intervalle de remplacement des fichiers journaux	90
Noms des fichiers journaux	90
Gestion des fichiers journaux	91

Chapitre 14. Gestionnaire d'utilisateurs et personnalisation (Alphablox 8.4.1) 93

Gestionnaire de personnalisation	93
Accès aux propriétés utilisateur et groupe JNDI	94

Chapitre 15. Gestionnaire d'utilisateurs (Alphablox 8.4). 97

Présentation du gestionnaire d'utilisateurs de DB2 Alphablox	97
Extensible User Manager	99
LDAP-Based User Manager	99
Configuration de DB2 Alphablox de façon à utiliser LDAP-based User Manager	99
Configuration des propriétés de LDAP-based User Manager.	100
Accès aux propriétés utilisateur personnalisées	101
Fonctionnement durant la phase d'exécution	101
Commande de console Telnet pour Extensible User Manager	102
Configuration du référentiel par défaut.	103
Suppression d'utilisateurs et de groupes ne se trouvant plus dans le référentiel d'utilisateurs externe	103
Interfaces Extensible User Manager	103
Implémentations personnalisées pour la sécurité.	104
Connexion unique.	105
Exemples de sécurité personnalisée	106
Exemple 1 : Configuration de DB2 Alphablox pour utiliser un gestionnaire d'utilisateurs externe	106
Exemple 2 : Configuration de DB2 Alphablox pour utiliser une classe d'utilisateur différente.	107
Exemple 3 : Configuration de DB2 Alphablox pour utiliser une classe de groupe différente	107
Références relatives aux méthodes d'interface	108
Interface IUserManager	109
findGroup()	109

findUser()	109
getExternalProperties()	110
getPrincipleUserName()	110
hasExternalEditor()	110
resume()	110
setCaseSensitiveGroups()	111
setCaseSensitiveUsers()	111
start()	111
stop()	112
suspend()	112
Interface IUser	112
authenticate()	113
authorize()	113
getEmail()	113
getFullName()	114
getName()	114
getPassword()	114
getPropertiesSubset()	115
isUserInRole()	115
refresh()	115
Interface IGroup	116
containsGroup()	116
containsUser()	116
getName()	116
getPropertiesSubset()	117
refresh()	117
Chapitre 16. Utilisation d'un référentiel de base de données	119
Présentation du référentiel DB2 Alphablox	119
Référentiel dans l'environnement DB2 Alphablox	119
Avantages du référentiel relationnel	119
Configuration du référentiel DB2 Alphablox	120
Vérification du type de référentiel	120
Utilisation de l'utilitaire de conversion du registre	120
Démarrage de l'utilitaire de conversion du registre	121
Options interactives de ligne de commande de l'utilitaire de conversion du registre	121
Conversion d'un système de fichiers en base de données	123
Conversion d'une base de données en système de fichiers	124
Configuration d'une instance afin qu'elle utilise un référentiel existant	125
Syntaxe de ligne de commande	126
Chapitre 17. Utilisation du regroupement de connexions	129
Regroupement de connexions - Présentation	129
Regroupement de connexions de bean géré par message	129
Regroupement de connexions DB2 OLAP Server et Hyperion Essbase	129
Microsoft Analysis Services et regroupement de connexions	130
Activation du pool de connexions	130
Utilisation du pool de connexions	131
Restriction du pool de connexions	131
Optimisation du pool de connexions	131
Regroupement de connexions de base de données relationnelle	133
Utilisation de DB2 Alphablox pour le regroupement de connexion de base de données relationnelle	133
Sources de données DB2 Alphablox et regroupement de connexions de base de données relationnelle	134
Référentiel DB2 Alphablox et regroupement de connexions de base de données relationnelle	134
Configuration du regroupement de connexions avec BEA WebLogic	135
Chapitre 18. Utilisation d'environnements groupés	137
Présentation des environnements groupés	137
Environnements groupés WebSphere	137
Environnements groupés WebLogic	137

Configuration et installation de DB2 Alphablox dans des environnements groupés WebLogic	138
Création d'applications dans des environnements groupés WebLogic	138
Utilisation de clusters verticaux WebLogic	138
Commandes de la console de cluster	138

Chapitre 19. Commandes de la console DB2 Alphablox 141

Accès à la console	141
Console HTML	141
Console Telnet	141
Syntaxe des commandes	142
Abréviation des commandes	142
Liste des commandes de la console	143
Commandes de console spécifiques à Essbase	146
Commandes RESOLVEALIASESTOBASEMEMBERS	147
Commande SHOW OUTLINECACHE	147
Commande DELETE OUTLINECACHE	148
Remarques relatives aux commandes de la console	148
Affichage des propriétés générales	148
Niveau des messages	149
Exécution d'un fichier texte via la console	149
Messages du journal de DB2 Alphablox	150

Chapitre 20. Administration des applications Alphablox FastForward 151

Présentation générale	151
Rôles des utilisateurs FastForward	151
Administrateurs d'application	151
Développeurs de modèles	152
Utilisateurs	152
Configuration système requise pour les applications FastForward	152
Création d'applications Alphablox FastForward	153
Modification des rôles d'administrateur	153
Administration des applications FastForward	154
Catégories et sécurité d'accès aux rapports	154
Rapports publiés	154
Rapports privés et de groupe	154
Agencement et commandes	154
Menu d'exploration	154
Gestion des rapports	155
Création de rapports	155
Modification des rapports	155
Suppression des rapports	155
Déplacement des rapports	155
Gestion des dossiers	156
Création de dossiers	156
Modification des dossiers	156
Suppression des dossiers	156
Déplacement des dossiers	156
Gestion des propriétés de l'application	157
Utilisation du journal de l'application	157

Annexe. Termes et concepts OLAP 159

Analyse bidimensionnelle	159
Tableau de ventes bidimensionnel	159
Analyse multidimensionnelle	159
Cube de données	160
Matrice multidimensionnelle des ventes	161
Termes de base de données OLAP	162

Glossaire 165

Remarques	173
Marques	175
Index	177

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. Présentation de DB2 Alphablox

DB2 Alphablox fonctionne dans un environnement de serveur d'applications J2EE et fournit des services de création d'applications Web analytiques. DB2 Alphablox peut être intégré aux principaux serveurs d'applications, notamment IBM WebSphere, BEA WebLogic et Apache Tomcat. Ce chapitre fournit une vue d'ensemble de DB2 Alphablox, explique comment DB2 Alphablox s'intègre à l'environnement J2EE et décrit l'architecture DB2 Alphablox.

Présentation de DB2 Alphablox

DB2 Alphablox facilite la création d'applications Web personnalisées, intégrées à l'infrastructure de l'entreprise et accessibles par les différents types d'utilisateurs, qu'ils soient à l'intérieur ou à l'extérieur du pare-feu de l'entreprise. Les applications créées avec la plateforme DB2 Alphablox peuvent être exécutées sur un navigateur Web standard, ce qui permet d'obtenir une analyse multidimensionnelle en temps réel et hautement personnalisable.

Les fonctions suivantes sont disponibles sur la plateforme DB2 Alphablox :

- utilisation de données stockées dans des bases de données multidimensionnelles et relationnelles
- création de rapports structurés à partir de bases de données relationnelles
- sélection parmi un large choix de graphiques de présentation des données
- création d'applications capables d'écrire des données dans la base de données (particulièrement utile dans les applications de planification financière "par simulation")
- interaction des utilisateurs avec les différents niveaux de données, grâce aux sources de données multidimensionnelles (par exemple, filtre, explorer vers le bas, etc.) afin d'afficher de façon interactive et précise la vue de données souhaitée
- l'interface utilisateur intuitive facilite et optimise l'analyse des données
- une même application peut accéder à plusieurs sources de données
- intégration à un large éventail de composants d'infrastructure d'entreprise, notamment les serveurs d'applications (IBM WebSphere et BEA WebLogic)

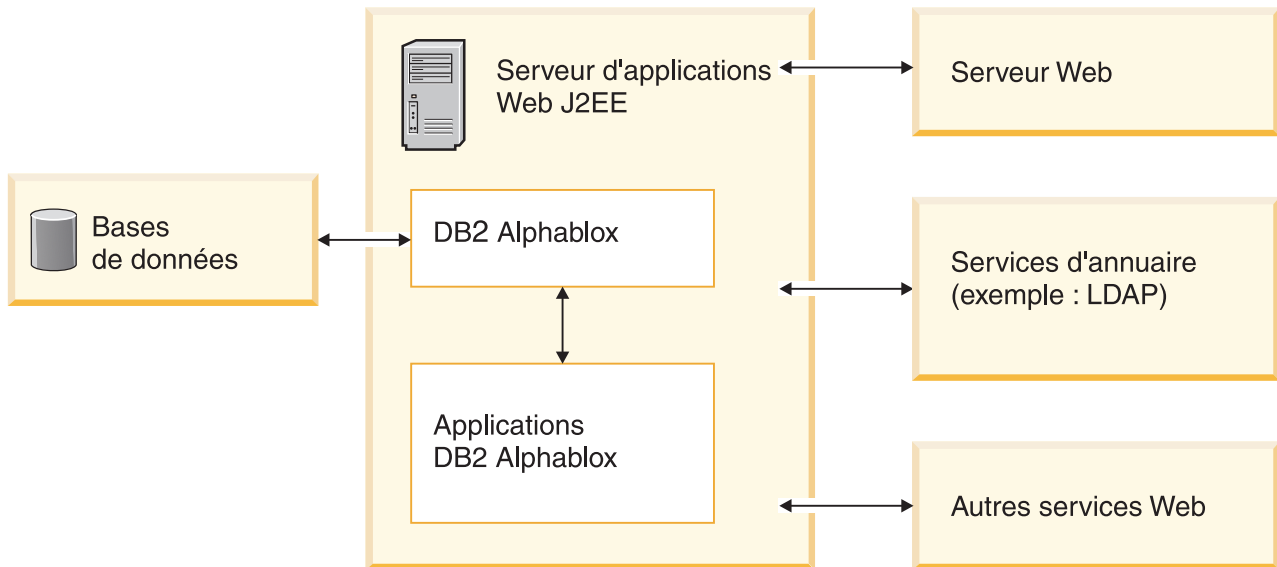
DB2 Alphablox fournit un grand choix d'API qui permettent aux développeurs de créer des applications personnalisées. Les API DB2 Alphablox sont développées en langage de programmation Java et les développeurs d'applications peuvent y accéder via Java (exécuté sur le serveur) ou via JavaScript (interprété au niveau du navigateur).

Ce chapitre décrit également comment DB2 Alphablox s'intègre à l'environnement J2EE, décrit les composants de DB2 Alphablox et l'architecture de DB2 Alphablox, le composant central de DB2 Alphablox. Pour obtenir des informations détaillées sur la création d'applications DB2 Alphablox, voir *Developer's Guide*. Pour connaître la syntaxe utilisée avec les API DB2 Alphablox et obtenir des informations de référence, voir *Developer's Reference*.

DB2 Alphablox en environnement J2EE

DB2 Alphablox est exécuté sur un serveur d'applications J2EE (IBM WebSphere ou BEA WebLogic) ou Apache Tomcat. Pour obtenir la liste des serveurs d'applications pris en charge et les versions correspondantes, voir *Guide d'installation*.

La figure suivante montre comment DB2 Alphablox s'intègre à l'environnement J2EE



Avantages de l'exécution sur un serveur d'applications

L'exécution sur un serveur d'applications présente plusieurs avantages :

- elle facilite l'intégration de composants J2EE ouverts standards à l'environnement DB2 Alphablox
- elle permet l'accès aux services fournis par les différents services Web
- elle fournit l'accès à tous les services fournis par l'environnement JRE (Java Runtime Environment), ainsi qu'à toutes les extensions Java disponibles sur un serveur d'applications particulier
- elle permet à DB2 Alphablox de centrer la plateforme sur l'activité d'analyse, en reléguant les autres aspects tels que les services Web, la sécurité, etc. aux fournisseurs spécialisés dans ces domaines
- elle fournit aux développeurs J2EE une plateforme reposant sur les technologies J2EE qu'ils connaissent bien, incluant par exemple des composants JSP (JavaServer Pages), Java, JavaBeans et XML.

Cette intégration poussée du serveur d'applications apporte à DB2 Alphablox tous les avantages tout en permettant aux développeurs d'accéder à une grande diversité d'API DB2 Alphablox.

Configurations des serveurs d'applications WebSphere et WebLogic

DB2 Alphablox peut être configuré pour exécuter les serveurs d'applications IBM WebSphere ou BEA WebLogic. Le fonctionnement standard de DB2 Alphablox est identique avec WebSphere, WebLogic ou Tomcat. Seules diffèrent quelques tâches d'administration mineures (par exemple, après avoir créé une application avec DB2 Alphablox, vous devez l'enregistrer sous WebSphere et le démarrage est effectué lors du démarrage du serveur d'applications).

Avec un serveur d'applications commercial tel que WebSphere ou WebLogic, vous avez accès à tous les outils, tous les services et toutes les possibilités d'évolutivité disponibles sur cette plateforme, ainsi qu'à tous les services disponibles sur la plateforme DB2 Alphablox.

Lors de l'installation, plusieurs options sont disponibles pour configurer WebSphere, WebLogic et Tomcat. Pour plus d'informations, voir *Guide d'installation*.

Comme les applications DB2 Alphablox sont de type J2EE, elles peuvent fonctionner dans différentes configurations. Par conséquent, vous pouvez développer et tester une application dans une configuration Apache Tomcat, puis la déployer dans la configuration du serveur d'applications de votre entreprise. Il peut y avoir quelques différences mineures au niveau de l'exécution sur différentes plateformes, il est donc nécessaire de tester une application avant de la déployer sur une autre configuration, mais si l'application n'utilise aucun service spécifique à la plateforme, la migration entre différentes configurations ne présente en général aucune difficulté.

Configuration d'Apache Tomcat

DB2 Alphablox peut utiliser le serveur d'applications Apache Tomcat. Dans la configuration Apache Tomcat, l'utilitaire d'installation de DB2 Alphablox utilise le serveur Apache Tomcat indiqué (version 3.2.4 uniquement). Pour obtenir des informations sur Apache Tomcat, voir <http://jakarta.apache.org/tomcat/> . Pour obtenir une copie de Tomcat version 3.2.4, voir <http://archive.apache.org/dist/tomcat/tomcat-3/archive/v3.2.4/>.

Si vous utilisez Apache Tomcat avec DB2 Alphablox, aucune tâche d'administration spécifique à Tomcat n'est nécessaire pour utiliser la plateforme DB2 Alphablox. La configuration Apache Tomcat est une solution évolutive, qui peut utiliser la solution de mise en cluster (environnement groupé) décrite dans le Chapitre 16, «Utilisation d'un référentiel de base de données», à la page 119.

La configuration Apache Tomcat inclut une solution de sécurité complète prenant en charge les utilisateurs, les groupes et les rôles. Vous pouvez utiliser l'un des serveurs Web externes (par exemple, Microsoft IIS, Sun iPlanet ou le serveur Web Apache) dans la configuration Apache Tomcat ou bien vous pouvez utiliser Tomcat pour gérer les requêtes HTTP.

Conseil : Le serveur HTTP inclus à la configuration Apache Tomcat convient bien pour le développement ou pour les petits systèmes de production, mais pour les systèmes de production de grande taille, il est préférable d'utiliser un serveur Web externe doté de nombreuses fonctions de mise en cache et de service de pages, afin d'améliorer les performances HTTP.

Composants de la plateforme DB2 Alphablox

Cette section décrit les différents composants de la plateforme DB2 Alphablox qui sont les suivants :

- «DB2 Alphablox», à la page 4
- «Applications DB2 Alphablox», à la page 8
- «Application Studio», à la page 8

DB2 Alphablox

DB2 Alphablox comporte une architecture robuste spécifiquement conçue pour faciliter le développement, le déploiement et l'utilisation des applications analytiques. Les principales fonctions fournies par l'architecture sont les suivantes :

- «Adaptateurs de données»
- «Composants Blox», à la page 5
- «Client DHTML», à la page 7
- «Fonctions d'administration à distance», à la page 7
- «Gestion centralisée des applications», à la page 7
- «Architecture à n-niveaux», à la page 8

Adaptateurs de données

DB2 Alphablox comporte un gestionnaire de données spécifiquement conçu pour la connexion d'une grande variété de bases de données. Le gestionnaire de données gère l'accès, l'exploration, l'interrogation et la récupération des données des bases de données relationnelles et multidimensionnelles (y compris les cubes DB2 Alphablox). La connexion à chaque base de données est implémentée via des adaptateurs (plug-in). Pour les bases de données relationnelles, les adaptateurs plug-in sont en général des pilotes JDBC. Dans chaque adaptateur sont encapsulés les informations et le traitement nécessaires à la connexion à une base de données spécifique, ce qui réduit considérablement les opérations nécessaires à la connexion aux bases de données supplémentaires.

Le gestionnaire de données et les adaptateurs de données associés prennent en charge les fonctionnalités suivantes :

- Exploration d'une collection de connexions préconfigurées à une base de données spécifique (appelée source de données).
- Accès des applications DB2 Alphablox aux bases de données disponibles de chaque source de données.
- Publication des types de requête compatibles pour une source de données spécifique.
- Balayage des métadonnées d'une base de données.
- Gestion des connexions de base de données pour les sessions utilisateur.
- Conversion des objets requête dans le langage de requête natif sous-jacent.
- Exécution des requêtes sur une base de données.
- Interrogation des données et du schéma d'un ensemble de résultats.
- Traitement d'un ensemble de résultats (affichage, permutation, développement, tri et exploration).
- Création d'applications qui permettent d'enregistrer la saisie de l'utilisateur et d'écrire les données dans la base de données sous-jacente (utilisation courante dans les applications de calcul de budget "par simulation").

Composants Blox

Les applications DB2 Alphablox utilisent des blocs élémentaires appelés composants Blox, pour accéder aux données d'entreprise en temps réel et les présenter via un navigateur Web standard. Les composants Blox sont des composants de logiciel réutilisables combinés ou *assemblés* sur une page JSP standard, qui permettent de constituer des applications interactives accessibles via un navigateur Web sur le réseau Intranet de l'entreprise ou sur Internet.

Ces composants de logiciel testés permettent d'intégrer aux applications les fonctionnalités suivantes :

- accès aux données
- analyse interactive des données
- présentation flexible des données
- accès aux données d'administration détaillées (par exemple, le nom des utilisateurs, des groupes et des applications)

Par exemple, un composant DataBlox peut alimenter une liste HTML avec les catégories de produit à partir d'une base de données sous-jacente. Ensuite, la sélection effectuée par l'utilisateur dans cette liste peut entraîner une requête sur la base de données sous-jacente. Ainsi, lorsqu'il choisira la catégorie *Voiture compacte*, il obtiendra un ensemble de données différent de celui de la catégorie *Véhicule utilitaire sport*.

Les composants Blox comportent des appels d'API étendus accessibles à l'aide de fichiers JSP. Les différents appels d'API permettent une grande souplesse de personnalisation des données accessibles, un contrôle avancé sur l'interactivité de l'utilisateur, ainsi que la personnalisation de la présentation des données aux utilisateurs.

L'assemblage de composants Blox en applications Web permet aux développeurs de fournir rapidement aux utilisateurs un accès immédiat aux informations pertinentes. L'utilisation de ces mêmes composants Blox pour différentes applications d'analyse permet non seulement d'optimiser la création, la livraison et la maintenance des applications, mais également de réduire la courbe d'apprentissage des utilisateurs. Quelles que soient les données présentées, le fonctionnement des composants Blox reste cohérent sur l'ensemble des applications et des plateformes.

De plus, les compétences requises pour assembler des applications DB2 Alphablox sont considérablement moins importantes que celles requises pour le développement d'applications traditionnelles. Le développeur d'application n'a pas besoin d'assimiler un langage de programmation complexe : il peut commencer rapidement les activités de développement en utilisant le langage HTML, JavaScript, les objets multimédia et les composants Blox. A l'aide de ces éléments, le développeur d'application peut créer des interfaces pratiques et esthétiques dotées de fonctions puissantes pour les utilisateurs.

DB2 Alphablox fournit les composants Blox suivants qui permettent d'assembler des applications :

Composant Blox	Utilisation
DataBlox	<ul style="list-style-type: none"> • Accès aux bases de données multidimensionnelles prises en charge et développement d'une présentation d'un ensemble de données pour le client. • Accès aux bases de données relationnelles prises en charge • Réception et exécution des demandes de requête • Fourniture des ensembles de résultats de requête aux composants Blox qui gèrent la présentation des données
ChartBlox	<ul style="list-style-type: none"> • Présentation d'une vue graphique des données multidimensionnelles • Permet aux utilisateurs d'organiser les données dans une grande diversité de formats de diagrammes (à secteurs, barres et courbes) • Permet aux utilisateurs d'explorer vers le bas les données hiérarchiques par séries ou de permuter les vues de données
DataLayoutBlox	<ul style="list-style-type: none"> • Présentation de listes regroupées contenant les dimensions disponibles et l'axe sur lequel elles se trouvent • Permet aux utilisateurs de naviguer entre les axes de page, ligne, colonne et "autres" (non utilisé)
GridBlox	<ul style="list-style-type: none"> • Présente des données multidimensionnelles ou relationnelles dans un format de grille avancé • Permet aux utilisateurs d'analyser et d'organiser les données multidimensionnelles • Permet aux utilisateurs d'explorer vers le bas les données hiérarchiques par séries ou de permuter les vues de données
PageBlox	<ul style="list-style-type: none"> • Présente des listes déroulantes de dimensions se trouvant sur l'axe de page (ce qui permet de filtrer les données qui apparaissent dans ChartBlox et GridBlox) • Permet aux utilisateurs de modifier les dimensions et/ou les membres sur lesquels filtrer les données
ToolbarBlox	<ul style="list-style-type: none"> • Présentation de boutons pour l'accès utilisateur aux fonctionnalités Blox, par exemple : <ul style="list-style-type: none"> – basculement entre les présentations de type graphique et grille – sélection du type de graphique – enregistrement et récupération des vues de l'application – déplacement de la barre d'outils pour faciliter l'accès – ouverture d'un Blox dans une fenêtre distincte – permutation des lignes et des colonnes • Permet à l'assembleur de sélectionner les boutons qui apparaissent sur la barre d'outils, ce qui permet d'adapter l'accès utilisateur aux fonctionnalités Blox.
PresentBlox	Combinaison des fonctionnalités des six composants Blox précédents (DataBlox, DataLayoutBlox, ChartBlox, GridBlox, PageBlox et ToolbarBlox) dans un seul composant Blox, ce qui simplifie l'assemblage d'applications et permet de conserver l'état réel de la page Web
RepositoryBlox	Fournit aux assembleurs d'applications l'accès aux objets stockés (notamment aux vues d'application enregistrées), ainsi qu'aux propriétés du serveur, des applications, des groupes et des utilisateurs

Voir *Developer's Reference* pour obtenir une description détaillée des API Blox disponibles avec DB2 Alphablox. Voir *Developer's Guide* pour consulter les instructions d'aide sur l'utilisation des composants Blox pour créer des applications.

Client DHTML

Comme DB2 Alphablox utilise les technologies DHTML standards, les développeurs d'applications peuvent créer des applications auxquelles les utilisateurs peuvent accéder à l'aide d'un navigateur Web, sans installer de plug-in supplémentaire. DB2 Alphablox affiche l'application dans le mode souhaité, sans nécessité de codage supplémentaire pour le développeur d'application.

Le **client DHTML** repose sur la technologie HTML dynamique et utilise JavaScript et CSS (feuilles de style en cascade) afin de prendre en charge la gamme complète de fonctionnalités d'analyse de données sur une interface pratique et personnalisable. Aucun plug-in ni aucun téléchargement de fichier de classes Java n'est nécessaire. Le client DHTML requiert l'utilisation des versions prises en charge de Mozilla Firefox et Microsoft Internet Explorer.

Fonctions d'administration à distance

DB2 Alphablox propose des fonctions d'administration système avancées via les pages Web d'administration, une fenêtre de commande de console ou une fenêtre Telnet. Sur ces interfaces d'administration, les administrateurs peuvent effectuer les types de tâche suivants :

- Création de journaux de DB2 Alphablox et définition des niveaux de consignation des événements et des messages.
- Surveillance de l'activité de DB2 Alphablox et des applications.
- Création d'objets DB2 Alphablox tels que les sources de données, les utilisateurs, les groupes et les applications.
- Surveillance de l'utilisation des objets DB2 Alphablox.
- Démarrage et arrêt des services DB2 Alphablox, des sessions et des applications.

Pour plus d'informations, voir Chapitre 4, «Tâches d'administration de base», à la page 25 et Chapitre 13, «Configuration des propriétés de DB2 Alphablox», à la page 77.

Gestion centralisée des applications

DB2 Alphablox comporte un gestionnaire d'applications qui gère les applications de façon centralisée. Ce gestionnaire d'applications gère et conserve la trace des éléments suivants :

- Liste des applications disponibles.
- Liste des applications actives.
- Activité au sein d'une application.
- Utilisateurs connectés aux applications.
- Composants Blox utilisés par les applications.
- Sources de données utilisées par les applications.
- Historique des instances d'application.

Architecture à n-niveaux

DB2 Alphablox est entièrement conçu sur les technologies Java les plus récentes de façon à implémenter une architecture Web à n-niveaux. Les applications comportent en général trois niveaux :

- Les données résident sur un serveur de base de données (relationnelle ou multidimensionnelle), depuis lequel elles sont récupérées.
- Les pages Web d'application se trouvent sur des applications Web sur le serveur d'applications et sont mises à disposition via DB2 Alphablox.
- La page Web s'affiche via l'interface utilisateur sur une machine client, dans le navigateur Web.

Sur demande, les composants Blox sont transférés d'un serveur Web vers les navigateurs Web via les réseaux Intranet ou Internet. Cette architecture Java permet d'éviter l'installation, la configuration et la gestion d'un logiciel d'application côté client.

Applications DB2 Alphablox

Du point de vue de l'utilisateur, une application DB2 Alphablox apparaît sous forme de collection de pages Web consultables de la même manière que les autres pages Web. Ces pages Web servent de conteneur pour les composants d'application suivants :

- Balises HTML standards et éléments de page (logos, texte, images, icônes, clips vidéo, clips audio, animations, etc.) pour rendre l'interface utilisateur plus attrayante
- Composants Blox nécessaires pour fournir la fonctionnalité d'application requise et l'interface utilisateur
- Scriptlets JavaScript ou Java pour une logique d'application et d'interface utilisateur avancée

Par exemple, une application d'analyse des ventes peut comporter une image cliquable des zones géographiques de vente. L'utilisateur peut cliquer sur une zone pour afficher le graphique de données correspondant. L'interface utilisateur permet aux utilisateurs de modifier le format du graphique, afficher les données dans un format de grille interactive et effectuer des opérations multidimensionnelles (telles que l'exploration ou la permutation). Selon l'application, l'utilisateur peut également afficher la sortie Blox pour l'imprimer ou l'exporter dans d'autres applications (par exemple, un tableur). Pour plus d'information sur les types et les composants d'application DB2 Alphablox, voir Chapitre 2, «Applications DB2 Alphablox», à la page 15.

Application Studio

Application Studio est à la fois un produit compagnon et un élément de DB2 Alphablox. Il permet de réduire le temps et les opérations nécessaires à la livraison d'applications analytiques complètes et personnalisées aux utilisateurs. Application Studio comporte les composants suivants :

- Un générateur de requête DHTML (DHTML Query Builder), un outil de plan de travail qui permet de développer et de tester les requêtes sur les sources de données d'application et d'accéder à différentes sources de données d'exemple pour le prototypage et la réalisation de tests
- Créateur de modèle de Blox - DHTML, contenant des exemples de code opérationnels qui mettent en valeur les fonctionnalités de DB2 Alphablox et aident les développeurs à comprendre le fonctionnement des API Blox.

Application Studio facilite le prototypage des applications. Une documentation complète fournit des instructions détaillées sur la configuration des modèles. Pour plus d'informations sur Application Studio et ses composants, voir l'aide en ligne qui fournit les différents modèles d'Application Studio.

Architecture DB2 Alphablox

Cette section décrit les composants suivants de DB2 Alphablox :

- «Gestionnaire de services»
- «Gestionnaire de requêtes»
- «Gestionnaire de sessions», à la page 10
- «Gestionnaire d'utilisateurs», à la page 10
- «Gestionnaire d'applications», à la page 11
- «Gestionnaire de données», à la page 11
- «Gestionnaire de référentiels», à la page 12
- «Gestionnaire de fichiers», à la page 12
- «Gestionnaire de consoles», à la page 12
- «Gestionnaire de cubes DB2 Alphablox», à la page 12
- «Gestionnaire de clusters», à la page 12

DB2 Alphablox comporte différents services exécutés de façon indépendante. De plus, des fonctions d'administration avancées sont fournies :

- Interface d'administration d'accès aux ressources DB2 Alphablox (telles que les applications, les sources de données, les utilisateurs et les groupes).
- Console administrative accessible via un navigateur Web, Telnet ou une fenêtre de commande.
- Possibilité de démarrer, interrompre, reprendre et arrêter les services, les sessions et les applications.
- Possibilité de surveiller l'activité de DB2 Alphablox et des applications.
- Journaux permettant de suivre les événements et les messages.

Dans le cadre de l'installation initiale, DB2 Alphablox est automatiquement configuré à l'aide d'un ensemble de propriétés par défaut requises pour son fonctionnement. Par la suite, les administrateurs peuvent modifier ou ajouter des propriétés à la configuration par défaut de DB2 Alphablox. Pour obtenir des détails, voir Chapitre 13, «Configuration des propriétés de DB2 Alphablox», à la page 77.

Gestionnaire de services

Le gestionnaire de services permet d'accéder aux autres services, de les démarrer et de les contrôler (y compris les services tiers requis). Le gestionnaire de services traite toutes les requêtes de service et facilite la distribution des composants de service.

Gestionnaire de requêtes

Le gestionnaire de requêtes contrôle toutes les requêtes des pages d'application et des services homologues. DB2 Alphablox crée un *homologue* pour chaque connexion. L'homologue conserve la trace de l'état de la connexion. Le gestionnaire de requêtes crée les moniteurs et gère les unités d'exécution de chaque requête, valide les ID de session par rapport à la liste de sessions valides et crée les objets de requête appropriés pour gérer chaque requête.

Gestionnaire de sessions

Le gestionnaire de sessions contrôle et surveille les interactions entre les utilisateurs et les applications DB2 Alphablox. Lorsqu'un utilisateur envoie une requête à une application, le gestionnaire de sessions transmet cette requête au gestionnaire d'utilisateurs pour obtenir une authentification. Une fois l'authentification de l'utilisateur effectuée, le gestionnaire de sessions crée une nouvelle session. Un utilisateur peut établir plusieurs sessions simultanées (plusieurs instances de navigateur). Le gestionnaire de sessions crée et gère les objets de session et conserve la trace des applications visitées par l'utilisateur.

Le gestionnaire de sessions termine également les sessions inactives (après la sauvegarde de l'état en cours de chaque application) et libère les ressources de session. Si un utilisateur tente de travailler dans la fenêtre du navigateur après l'expiration de la session, un message de reconnexion s'affiche. Si l'utilisateur demande une reconnexion (en cliquant sur le bouton **Rafraîchir** du navigateur dans Internet Explorer ou **Actualiser** dans Mozilla Firefox), le gestionnaire de sessions crée une nouvelle session et remet l'application dans l'état dans lequel elle se trouvait avant l'expiration de la session.

Gestionnaire d'utilisateurs

Le gestionnaire d'utilisateurs d'Alphablox contrôle tous les utilisateurs des services DB2 Alphablox, gère l'authentification des utilisateurs des applications et des administrateurs, surveille leurs ressources et gère l'accès des utilisateurs aux données et aux applications, la liste des utilisateurs actifs et les informations relatives à chaque utilisateur (par exemple moment et durée de la connexion d'un utilisateur à une application spécifique).

Dans DB2 Alphablox 8.4 et les versions précédentes, le gestionnaire d'utilisateurs est constitué à partir du moteur de personnalisation appelé Extensible User Manager. Ce gestionnaire d'utilisateurs externe repose sur Servlet 2.2, gère la sécurité et fournit la fonctionnalité de gestion des utilisateurs basé sur LDAP ou sur un référentiel, ainsi que des API de sécurité personnalisée pour les autres référentiels externes. À partir de DB2 Alphablox 8.4.1, DB2 Alphablox prend en charge la spécification Servlet 2.4. Extensible User Manager est obsolète, il est remplacé par le gestionnaire de personnalisation, lequel repose sur Servlet 2.4. Le gestionnaire de personnalisation fournit des méthodes de notification de DB2 Alphablox lors de la création, le chargement ou la suppression d'un objet utilisateur ou groupe. Vous pouvez également l'utiliser pour définir et modifier les propriétés d'un utilisateur ou d'un groupe, ou modifier les utilisateurs et les groupes contenus dans un groupe. La sécurité constitue désormais un modèle distinct. Les informations d'authentification des utilisateurs sont configurées sur le serveur d'applications sous-jacent, tandis que DB2 Alphablox fournit des API JAAS (Java Authentication and Authorization Service) pour l'authentification des utilisateurs.

Le gestionnaire d'utilisateurs détecte et gère les objets suivants qui constituent le schéma de sécurité de DB2 Alphablox :

- Un utilisateur est un utilisateur final ou bien un administrateur d'application.
- Un groupe est une collection d'utilisateurs et d'autres groupes, et fournit un mécanisme approprié aux administrateurs pour gérer plusieurs utilisateurs comme une seule entité. La structure des groupes est hiérarchique.
- Les droits d'accès déterminent quel accès (pas d'accès, lecture, lecture/écriture) un utilisateur particulier dispose dans un rôle.

- Un rôle est une liste d'utilisateurs et/ou de groupes associés à un ensemble de droits.

Remarque : Notez les informations suivantes sur les droits d'accès et les rôles : si un utilisateur apparaît plusieurs fois dans le même rôle (en tant qu'utilisateur individuel et en tant que membre de groupe), il bénéficie de la totalité des droits de toutes les entrées. Par exemple, si l'utilisateur dispose des droits de lecture/écriture et le groupe de l'utilisateur des droits de lecture, l'utilisateur aura des droits de lecture/écriture.

Gestionnaire d'applications

Le gestionnaire d'applications crée une définition d'application lorsqu'un assembleur ou un administrateur crée ou modifie une application sur la page d'accueil de DB2 Alphablox, onglet **Administration**, page **Applications**.

Le gestionnaire d'applications accepte également les requêtes utilisateur pour une application, vérifie les droits d'accès des utilisateurs et crée une instance d'application pour chaque requête. Le gestionnaire d'applications surveille toutes les instances d'application, permet à l'administrateur d'arrêter une instance, ferme automatiquement les instances inactives et gère les informations suivantes :

- Liste des applications disponibles
- Liste des applications actives
- Une activité dans une application
- Utilisateurs connectés aux applications
- Composants Blox utilisés par les applications

Gestionnaire de données

Le gestionnaire de données gère l'accès aux différentes sources de données utilisées par une application. Le gestionnaire de données permet l'exploration, l'interrogation et la récupération des données parmi une grande variété de sources de données relationnelles et multidimensionnelles, comme décrit dans le manuel *Guide d'installation* relatif à cette version.

Le gestionnaire de données fonctionne de façon spécifique avec un adaptateur de base de données pour les fonctions suivantes :

- Exploration d'une collection de sources de données préconfigurées spécifiques
- Présentation des cubes disponibles de chaque source de données (le cas échéant)
- Publication des types de requête compatibles pour une source de données spécifique
- Balayage des métadonnées d'une base de données
- Gestion des connexions de base de données pour les sessions utilisateur
- Conversion des objets requête dans le langage de requête natif sous-jacent
- Exécution des requêtes sur une base de données
- Interrogation des données et du schéma d'un ensemble de résultats
- Modification d'un ensemble de résultats (permutation, développement et exploration)
- Création d'applications qui permettent d'enregistrer la saisie de l'utilisateur dans la base de données pour l'analyse par simulation

Gestionnaire de référentiels

Le gestionnaire de référentiels contrôle le référentiel DB2 Alphablox et son contenu, ce qui inclut les vues d'application enregistrées et toutes les autres propriétés d'application, de groupe et d'utilisateur. Le référentiel DB2 Alphablox peut être stocké sur des fichiers de système d'exploitation ou dans une base de données relationnelle. Pour obtenir des détails, voir Chapitre 16, «Utilisation d'un référentiel de base de données», à la page 119.

Gestionnaire de fichiers

Le gestionnaire de fichiers gère les fichiers temporaires et les fichiers utilisés dans les pages d'administration de DB2 Alphablox.

Gestionnaire de consoles

Le gestionnaire de consoles fournit plusieurs interfaces (navigateur Web, Telnet et fenêtre de commande) qui permettent à l'administrateur système de surveiller et gérer l'environnement DB2 Alphablox. Pour plus d'informations, voir Chapitre 19, «Commandes de la console DB2 Alphablox», à la page 141 et «Console», à la page 20.

Gestionnaire de cubes DB2 Alphablox

Gestionnaire de cubes DB2 Alphablox fournit des interfaces permettant d'arrêter et de démarrer les cubes DB2 Alphablox et d'effectuer d'autres tâches d'administration. Pour obtenir des informations sur DB2 Alphablox Cube Server, voir *DB2 Alphablox Cube Server - Guide d'administration*.

Gestionnaire de clusters

Le gestionnaire de clusters gère la communication entre différents noeuds de DB2 Alphablox lorsqu'il est exécuté dans un environnement groupé. Une configuration en cluster permet d'augmenter l'évolutivité de votre système DB2 Alphablox pour prendre en charge un grand nombre d'utilisateurs. Pour obtenir des informations sur la configuration de DB2 Alphablox pour l'exécution dans un environnement groupé, voir Chapitre 18, «Utilisation d'environnements groupés», à la page 137.

Liste des tâches lors de la première utilisation de DB2 Alphablox

La liste suivante présente les principales tâches à effectuer lors de la première utilisation de DB2 Alphablox. Pour plus d'informations sur chaque tâche, voir le manuel ou la section indiquée dans la colonne *Référence*.

Tâche	Référence
1. Comprendre le fonctionnement de l'environnement dans lequel est exécuté DB2 Alphablox afin de choisir la configuration (par exemple, IBM WebSphere, BEA WebLogic ou Apache Tomcat).	Chapitre 1, «Présentation de DB2 Alphablox», à la page 1
2. Installer et configurer DB2 Alphablox.	<i>Guide d'installation</i>
3. Connaître le contenu des pages d'accueil de DB2 Alphablox qui permettent d'effectuer des tâches d'administration, telles que la création d'applications, la création d'utilisateurs et la configuration des règles de sécurité.	Chapitre 3, «Page d'accueil de DB2 Alphablox», à la page 19
4. Comprendre l'environnement de base de données dans lequel se trouvent les données.	Votre environnement de base de données et vos agents de gestion des connaissances
5. Comprendre l'environnement de développement pour la création des applications DB2 Alphablox.	Chapitre 2, «Applications DB2 Alphablox», à la page 15 et <i>Developer's Guide</i> .
6. Créer une définition d'application.	Chapitre 6, «Définitions d'application», à la page 31
7. Développer et tester l'application à l'aide de la technologie JSP, des bibliothèques de balises personnalisées Blox et les API de DB2 Alphablox.	Voir <i>Developer's Guide</i> et <i>Developer's Reference</i>
8. Déployer l'application de votre entreprise.	Vos utilisateurs, le réseau Intranet de l'entreprise et Internet.

Chapitre 2. Applications DB2 Alphablox

Vous pouvez créer des applications DB2 Alphablox permettant d'accéder aux données pour analyse via un navigateur, dans un environnement intranet d'une entreprise ou sur Internet. Etant donné que les applications DB2 Alphablox peuvent être personnalisées et qu'elles peuvent être intégrées à un grand nombre d'entreprises, il existe une gamme étendue d'applications pouvant être appelées applications "DB2 Alphablox." Le présent chapitre décrit brièvement ces types d'application ainsi que les composants d'une application DB2 Alphablox. Pour plus d'informations sur les applications DB2 Alphablox, voir *Developer's Guide*.

Types d'application DB2 Alphablox

Les applications DB2 Alphablox sont des applications Web compatibles J2EE qui s'exécutent dans l'environnement DB2 Alphablox. Etant donné que les applications DB2 Alphablox s'exécutent dans un navigateur Web et qu'elles peuvent être associées à d'autres technologies d'application Web, l'aspect et le champ d'action de ces applications peuvent être très variés. La fonction de base des applications DB2 Alphablox permet d'afficher de manière interactive des données analytiques sur une page Web afin que les utilisateurs puissent consulter ces données et ainsi s'informer sur le secteur d'activité sous-jacent aux données. Les données se trouvent dans différentes bases de données d'une entreprise.

Pour les composants DB2 Alphablox dans ces applications Web, cette section décrit brièvement les catégories suivantes des applications DB2 Alphablox :

- «Modes de présentation des données»
- «Application de création de rapports relationnels», à la page 16

Vous pouvez également associer divers éléments DB2 Alphablox dans une seule application. Par exemple, vous pouvez afficher un graphique et une feuille de calcul dans la même application. Pour obtenir plus de détails sur les différentes possibilités et sur le mode de développement de ces applications, consultez les documents *Developer's Guide* et *Relational Reporting Developer's Guide*.

Modes de présentation des données

Il existe plusieurs moyens de présenter les données dans les applications DB2 Alphablox. Vous pouvez afficher les données dans une grille et/ou dans un graphique. Vous pouvez également utiliser les données d'une base de données pour charger les éléments de vos applications (listes déroulantes, menus, etc.).

Le client DHTML DB2 Alphablox utilise la dernière technologie Dynamic HTML, incluant HTML, JavaScript et CSS (Feuilles de style en cascade), pour prendre en charge plusieurs fonctionnalités d'analyse de données avec une interface utilisateur graphique personnalisable et facile d'utilisation, comportant des menus contextuels. L'extensibilité disponible lors de l'utilisation des API blox avec le modèle d'interface utilisateur de blox constitue un des principaux avantages du client DHTML. Le mode client DHTML peut être utilisé uniquement avec les navigateurs Microsoft Internet Explorer les plus récents. Aucun plug-in de navigateur Web, installation de machine client ou téléchargement de fichiers de classe Java n'est requis.

Conversion au format PDF

Vous pouvez créer des applications DB2 Alphablox qui permettent d'imprimer des rapports dans des fichiers PDF. Vous obtenez ainsi une sortie de très haute qualité. La fonction d'impression au format PDF utilise un modèle PDF qui peut contenir tous les éléments requis (logo de l'entreprise, texte modèle, etc.) puis extrait vos données actives et crée un fichier PDF contenant un rapport de qualité supérieure. Vous pouvez alors imprimer, transmettre par messagerie électronique ou placer le fichier PDF sur un site Web pour distribution. La fonctionnalité PDF peut être utilisée avec l'ensemble des modes de rendu du client

Personnalisation des modes de rendu à l'aide de XML

DB2 Alphablox fournit des API qui permettent d'accéder aux données au format XML. Ainsi, les développeurs peuvent créer des couches de données personnalisées ou manipuler les données en ayant recours aux techniques personnalisées souhaitées. Ces API sont particulièrement utiles lorsque vous avez des besoins précis que les autres méthodes de présentation ne gèrent pas.

Application de création de rapports relationnels

Vous pouvez utiliser des composants ReportBlox pour créer des applications DB2 Alphablox qui fournissent des rapports provenant de bases de données relationnelles. Ces rapports sont personnalisables et sont affichés dans des tables HTML, à l'aide de la technologie CSS (Feuilles de style en cascade) qui permet une personnalisation facile et une présentation flexible.

Les rapports créés à l'aide des composants ReportBlox peuvent être utilisés en mode interactif afin de mettre à disposition une interface utilisateur qui permet de les personnaliser, ainsi qu'une API complète qui permet de les changer à l'aide d'un programme. Pour plus de détails sur les composants ReportBlox, voir le document *Relational Reporting Developer's Guide*.

Composants d'une application DB2 Alphablox

Etant donné que les applications DB2 Alphablox sont des applications J2EE, elles peuvent comporter tous les composants disponibles pour les applications J2EE. La présente section décrit les composants de base suivants des applications DB2 Alphablox :

- «JSP (JavaServer Pages)»
- «Composants Blox», à la page 17

Pour obtenir une description de l'arborescence des applications DB2 Alphablox, voir «Définition d'application DB2 Alphablox», à la page 31.

JSP (JavaServer Pages)

La technologie JSP (JavaServer Pages) est le mécanisme à partir duquel un développeur d'applications fournit la fonctionnalité DB2 Alphablox à une application. Les pages Web interactives sont développés à l'aide de JSP, des bibliothèques de balises personnalisées Blox, de HTML et tout autre code Java ou JavaScript requis par votre application. Les pages JSP sont compilées lors de l'exécution sur le serveur d'applications (pour des raisons de performances, le serveur d'applications compile uniquement la page lors du premier accès ; les requêtes de pages suivantes utilisent la page précompilée dans la plupart des configurations de serveur d'applications), puis DB2 Alphablox traite toutes les requêtes DB2 Alphablox et pour finir le serveur Web renvoie le contenu dynamique à l'utilisateur.

Composants Blox

Ces composants sont des composants DB2 Alphablox permettant par exemple d'accéder aux sources de données, d'afficher des données dans des grilles et des graphiques, etc. Pour ajouter des composants Blox à une page JSP, vous utilisez les bibliothèques de balises personnalisées Blox. Les bibliothèques de balises Blox permettent de contrôler un grand nombre d'aspects du blox. Par exemple, vous pouvez changer un attribut de ChartBlox afin d'indiquer le type de graphique affiché.

Vous pouvez également manipuler des composants blox à l'aide d'un programme, en utilisant soit Java, soit JavaScript. Lorsque vous utilisez JavaScript pour accéder aux API de blox, le code est interprété sur le navigateur. Lorsque vous utilisez Java pour accéder aux API de blox, le code est compilé par le moteur JSP (le serveur d'applications) puis exécuté sur le serveur.

Pour plus d'informations sur les types d'utilisation du blox, voir «Client DHTML», à la page 7. Pour plus de détails sur l'utilisation des composants Blox dans des applications DB2 Alphablox et sur les types de composant disponibles lors de l'utilisation du client DHTML, voir les documents *Developer's Guide* et *Developer's Reference*.

Chapitre 3. Page d'accueil de DB2 Alphablox

Ce chapitre décrit la page d'accueil de DB2 Alphablox. Pour plus d'informations sur l'interface de ligne de commande, voir Chapitre 19, «Commandes de la console DB2 Alphablox», à la page 141.

Présentation de la page d'accueil de DB2 Alphablox

DB2 Alphablox est automatiquement configuré durant l'installation initiale. Après l'installation, les administrateurs peuvent définir ou modifier la configuration, les propriétés et les ports de démarrage de DB2 Alphablox. Pour administrer DB2 Alphablox, utilisez l'onglet **Administration** de la page d'accueil de DB2 Alphablox ou l'interface de ligne de commande de la console. Pour accéder aux pages d'administration de DB2 Alphablox, entrez l'une des adresses URL suivantes dans la fenêtre de votre navigateur :

```
http://<nom_serveur>/AlphabloxAdmin/home/  
http://<nom_serveur>/AlphabloxAdmin/  
http://<nom_serveur>/AlphabloxAdmin (sauf iPlanet)
```

où <nom_serveur> représente le nom du serveur et le numéro de port sur lequel DB2 Alphablox s'exécute.

Onglet Applications

Lors de l'ouverture de la page d'accueil de DB2 Alphablox, l'onglet Application est sélectionné par défaut.

L'onglet **Applications** s'affiche par défaut lorsqu'un utilisateur accède à la page d'accueil de DB2 Alphablox. L'onglet **Applications** répertorie les applications et les états des applications enregistrées disponibles auxquelles l'utilisateur a accès. Certains utilisateurs ne verront peut-être que l'onglet **Applications** parce qu'ils n'ont pas les privilèges nécessaires pour afficher les autres onglets ou pour y accéder.

Pour lancer une application, cliquez sur le lien de votre choix. Les applications sont lancées en mode DHTML par défaut. Le fait de sélectionner **Etat de l'application par défaut** dans la liste déroulante charge l'application à l'état défini par défaut dans la définition de l'application.

Autres applications

Toutes les autres applications définies dans DB2 Alphablox apparaissent également sous l'onglet **Applications**. Pour lancer une application, cliquez sur son lien. Les applications sont lancées en mode DHTML par défaut. Le fait de sélectionner **Etat de l'application par défaut** dans la liste déroulante charge l'application à l'état défini par défaut dans la définition de l'application.

Onglet Administration

L'onglet **Administration** apparaît sur la page d'accueil de DB2 Alphablox uniquement si l'utilisateur authentifié dispose des privilèges d'administration. Le fait de cliquer sur cet onglet ouvre la page **General** sur laquelle les administrateurs peuvent définir ou modifier les aspects de DB2 Alphablox. Les sections suivantes se trouvent sous l'onglet **Administration** :

- General
- «Groups (Groupes)», à la page 21
- «Users (Utilisateurs)», à la page 21
- «Roles (Rôles)», à la page 22
- «Applications», à la page 22
- «Data Sources (Sources de données)», à la page 22
- «Cubes DB2 Alphablox», à la page 22

Autres dispositions

Le lien **General** sous l'onglet **Administration** fournit l'interface permettant aux administrateurs de configurer les fonctions principales de DB2 Alphablox.

General Properties (Propriétés générales)

La section **General Properties** vous permet de configurer les paramètres de démarrage de DB2 Alphablox et de définir le nombre maximal de cubes DB2 Alphablox autorisés pour le DB2 Alphablox Cube Server. Pour plus de détails sur l'exécution de ces tâches, voir «Tâches d'administration de DB2 Alphablox», à la page 77. Pour plus d'informations sur le DB2 Alphablox Cube Server, consultez le document *DB2 Alphablox Cube Server - Guide d'administration*.

Custom Properties (Propriétés personnalisées)

Pour plus de détails sur la définition des propriétés personnalisées, voir «Définitions des propriétés personnalisées», à la page 82.

Runtime Management (Gestion de l'exécution)

Utilisez la section **Runtime Management** pour gérer votre licence DB2 Alphablox, les sessions d'applications, les collections de commentaires, les rapports PDF (DHTML uniquement) et les cubes DB2 Alphablox. Pour plus d'informations sur la création et la maintenance des cubes DB2 Alphablox, consultez le document *DB2 Alphablox Cube Server - Guide d'administration*.

Console

Le fait de cliquer sur le lien **Start Console Session** ouvre la fenêtre console, dans laquelle les administrateurs peuvent entrer des commandes textuelles pour effectuer la plupart des fonctions administratives, normalement effectuées à travers l'interface utilisateur. Pour plus d'informations sur l'utilisation de la console, voir Chapitre 19, «Commandes de la console DB2 Alphablox», à la page 141.

Remarque : L'utilisateur de la console Telnet (telle que définie dans la page d'accueil de DB2 Alphablox, onglet **Administration**, lien **General**, lien **Telnet Console**) peut également accéder à la console à travers tout logiciel de terminal Telnet. Vous pouvez accéder au logiciel Telnet sur la plupart des systèmes exécutant un système d'exploitation Microsoft Windows, à partir du menu Windows **Démarrer**, dossier **Tous les programmes**, groupe **Accessoires**, raccourci **Telnet** ou ouvrir une fenêtre Telnet à partir du menu **Démarrer**, raccourci **Exécuter**, puis en tapant **telnet**.

Portal Theme Utility (Utilitaire de thèmes de portail)

Remarque : L'option Portal Theme Utility s'affiche dans le menu uniquement lorsque DB2 Alphablox est installé sur un WebSphere Portal Server.

Cliquez sur le lien **Portal Theme Utility** pour créer les thèmes de portail à utiliser dans les applications de portail WebSphere contenant des composants Blox DB2 Alphablox. Avec cet utilitaire, un nouveau thème de portail DB2 Alphablox est créé en fusionnant les propriétés de style du thème WebSphere Portal Server spécifié avec les propriétés de style du thème DB2 Alphablox spécifié. Le nouveau thème peut être utilisé pour afficher les composants Blox de façon optimale (par exemple, un GridBlox ou ChartBlox) dans les applications de portail.

Une version portlet en option de l'utilitaire de thèmes de portail est disponible pour une utilisation dans les applications de portail. Ce portlet se trouve dans le fichier `AlphabloxAdminPortlets.war`, dans le répertoire `installableApps` de votre installation de DB2 Alphablox.

Pour obtenir de l'aide sur l'utilitaire de thèmes de portail, consultez l'aide en ligne, disponible en cliquant sur le bouton Aide de cet utilitaire. Sur la version portlet de l'utilitaire de thèmes de portail, le fait de sélectionner le mode aide ouvre l'aide en ligne.

Migration d'applications

Remarque : L'option Application Migration Utility s'affiche dans le menu uniquement lorsque DB2 Alphablox est installé sur un serveur Apache Tomcat.

Cliquez sur le lien **Application Migration** pour accéder à l'Utilitaire de migration d'applications Alphablox pour Tomcat 5.5. Cet utilitaire vous aide à migrer les applications DB2 Alphablox sélectionnées d'Apache Tomcat 3.2.4 vers Apache 5.5. Pour chaque application migrée, l'utilitaire :

- Copie l'application (si nécessaire) vers le répertoire *webapps* de l'installation Alphablox actuelle.
- Déploie l'application vers Apache Tomcat 5.5.
- Importe l'application vers Alphablox (si nécessaire).
- Met à jour les fichiers TLD (Tag Library Descriptor, descripteur de bibliothèque de balises) Alphablox pour l'édition actuelle.
- Met à jour le fichier *web.xml* de l'application sur la spécification Servlet 2.4.

Les applications migrées doivent fonctionner correctement sur les installations Apache Tomcat 5.5 avec DB2 Alphablox.

Groups (Groupes)

Utilisez le lien **Groups** sous l'onglet **Administration** pour définir les groupes, attribuer des utilisateurs aux groupes, etc. Pour plus de détails sur la configuration des groupes, voir Chapitre 9, «Définitions de groupes», à la page 55.

Users (Utilisateurs)

Utilisez le lien **Users** sous l'onglet **Administration** pour définir les utilisateurs, attribuer des appartenances de groupes aux utilisateurs, etc. Pour plus de détails sur la configuration des utilisateurs, voir Chapitre 8, «Définitions d'utilisateurs», à la page 51.

Roles (Rôles)

Utilisez le lien **Roles** sous l'onglet **Administration** pour créer et modifier les rôles. Les rôles vous permettent de contrôler quels utilisateurs et/ou groupes ont accès à une application particulière. Pour plus de détails sur la configuration des rôles, voir Chapitre 10, «Définitions des rôles», à la page 59.

Applications

Utilisez le lien **Applications** sous l'onglet **Administration** pour créer ou modifier l'état des applications qui résident sur le DB2 Alphablox. Vous pouvez définir des états par défaut, définir des rôles et définir une image s'affichant sur l'onglet Applications. Pour plus de détails sur les tâches nécessaires pour définir une application, voir Chapitre 6, «Définitions d'application», à la page 31.

Data Sources (Sources de données)

Utilisez le lien **Data Sources** sous l'onglet **Administration** pour définir et modifier les sources de données disponibles pour les applications DB2 Alphablox s'exécutant sur DB2 Alphablox. Vous pouvez définir des sources de données pour accéder aux bases de données OLAP (par exemple, IBM DB2 OLAP Server, Hyperion Essbase et Microsoft Analysis Services), aux bases de données relationnelles (par exemple, IBM DB2 UDB, Oracle, Sybase et Microsoft SQL Server) et aux cubes DB2 Alphablox. Pour plus de détails sur la définition et la modification d'une source de données, voir Chapitre 7, «Définitions de sources de données», à la page 43.

Cubes DB2 Alphablox

Utilisez le lien **Cubes** sous l'onglet **Administration** pour définir ou modifier les cubes DB2 Alphablox. Les cubes DB2 Alphablox présentent les données stockées dans une base de données relationnelle dans un format multidimensionnel. Pour plus d'informations sur les cubes DB2 Alphablox et le DB2 Alphablox Cube Server, consultez le document *DB2 Alphablox Cube Server - Guide d'administration*.

Onglet Assemblage : Application Studio

Le fait de cliquer sur l'onglet **Assemblage** de la page d'accueil de DB2 Alphablox ouvre DB2 Alphablox Application Studio. La page d'ouverture d'Application Studio contient des liens vers des pages contenant des exemples et des outils. Ces pages sont conçues pour aider les développeurs à créer rapidement des applications DB2 Alphablox. Application Studio est un ensemble d'outils de plan de travail conviviaux et d'exemples de codes de travail qui facilite la création d'applications. Pour plus d'informations sur Application Studio, consultez le document *Developer's Guide*.

Plan de travail

La page **Plan de travail** sous l'onglet **Assemblage** contient un lien vers **Query Builder**. Query Builder est une application qui permet aux développeurs d'entrer des requêtes de façon interactive sur diverses sources de données et de consulter les résultats dans une grille et/ou un graphique.

Exemples

La page **Exemples** sous l'onglet **Assemblage** contient des liens vers des exemples d'applications JSP, des échantillons de code et autres exemples. Dans les exemples, vous trouverez le *Blox Sampler*, qui contient de nombreux exemples de code de travail. Ces échantillons de code sont très utiles pour apprendre à développer des applications avec DB2 Alphablox et bon nombre d'entre eux peuvent être directement coupés et collés dans vos propres applications personnalisées.

Modèles de données

Des modèles de données à utiliser dans l'application *Blox Sampler* sont fournis avec DB2 Alphablox. Les modèles de données sont regroupés dans une base de données nommée Quality Chocolate Company (QCC) et il existe des versions de la base de données disponibles pour plusieurs bases de données, multidimensionnelle et relationnelle. Pour plus de détails sur l'installation des modèles de données, consultez "Loading the Sample Data" (Chargement des modèles de données) dans le *Guide d'installation*.

Liens dans le coin supérieur droit

Chaque onglet de la page d'accueil de DB2 Alphablox présente deux liens dans le coin supérieur droit de la page :

- «Mon profil»
- «Aide»

Mon profil

Le fait de cliquer sur le lien **Mon profil** ouvre la page **Profil**. Les utilisateurs peuvent modifier leur description de profil, leur mot de passe et leur adresse e-mail et installer ou désinstaller les fichiers (JAR) de classe locale à partir de cette page.

Aide

Le menu **Aide** fournit les liens suivants :

- Le lien **Aide sur** fournit une aide en ligne d'après contexte pour la page sur laquelle le bouton est cliqué. L'aide en ligne fournit des informations sur l'utilisation de l'interface pour chaque page.
- Le lien **Documentation en ligne** ouvre la documentation en ligne de DB2 Alphablox dans une nouvelle fenêtre de navigateur. La documentation en ligne contient les versions Web (HTML) et PDF de la documentation de DB2 Alphablox.
- Le lien **Référence d'API côté serveur** ouvre le Javadoc pour les API Java côté serveur.
- Le lien **A propos de DB2 Alphablox** fournit les informations de copyright relatives à DB2 Alphablox.

Remarque : Les utilisateurs qui ne sont pas membres du rôle AlphabloxAdministrator ne voient pas tous ces liens.

Chapitre 4. Tâches d'administration de base

DB2 Alphablox inclut un ensemble complet de fonctions permettant de gérer l'environnement du serveur. La page d'accueil DB2 Alphablox fournit un point unique permettant d'administrer tous les aspects du serveur. Cette section décrit les procédures permettant de démarrer, d'accéder à, de configurer et de démarrer DB2 Alphablox.

Démarrage de DB2 Alphablox

Lorsque vous exécutez DB2 Alphablox avec un serveur d'applications commercial tel IBM WebSphere ou BEA WebLogic, DB2 Alphablox est intégré au serveur d'applications. Lorsque vous démarrez le serveur d'applications, DB2 Alphablox est automatiquement démarré. Pour IBM WebSphere Portal Server, DB2 Alphablox est démarré lorsque vous démarrez le serveur de portail. Pour plus d'informations sur le démarrage de WebLogic ou de WebSphere, consultez la documentation relative à ces produits.

Accès à DB2 Alphablox

DB2 Alphablox est géré via l'onglet **Administration** de la page d'accueil DB2 Alphablox.

Pour accéder à DB2 Alphablox via un navigateur, entrez l'URL suivante :

```
http://<nomServeur>/AlphabloxAdmin/home/
```

où <nomServeur> représente le nom du serveur et le numéro de port sur lequel DB2 Alphablox s'exécute.

Pour accéder à l'onglet **Administration** de la page d'accueil DB2 Alphablox à partir du menu **Démarrer** de Windows :

1. Cliquez sur le menu **Démarrer** de Windows.
2. Sélectionnez le dossier **Tous les programmes**.
3. Sélectionnez le groupe de programmes **DB2 Alphablox**.
4. Sélectionnez le nom de l'instance de DB2 Alphablox (la valeur par défaut est **AlphabloxAnalytics**).
5. Sélectionnez le raccourci **Page d'accueil DB2 Alphablox**.

Arrêt de DB2 Alphablox

Sur des serveurs d'applications WebSphere, vous pouvez arrêter DB2 Alphablox en arrêtant le serveur d'applications. Pour WebSphere Portal Server, l'arrêt du serveur de portail arrête DB2 Alphablox. Sur des installations Apache Tomcat, vous pouvez arrêter DB2 Alphablox en utilisant le script d'arrêt lors de l'exécution dans une fenêtre de console ou lors de l'exécution en tant que service Windows, dans la section des services du Panneau de configuration.

Arrêt de DB2 Alphablox avec le script d'arrêt sur des installations Apache Tomcat

Pour arrêter DB2 Alphablox lorsque DB2 Alphablox est en cours d'exécution dans une fenêtre de console (sous Windows : la fenêtre qui s'ouvre lorsque vous démarrez DB2 Alphablox en sélectionnant le menu **Démarrer** Windows, le dossier **Tous les programmes**, le groupe **DB2 Alphablox**, le nom d'instance (la valeur par défaut est **AlphabloxAnalytics**) et le raccourci d'arrêt de DB2 Alphablox ; sous Sun Solaris : la fenêtre dans laquelle vous avez démarré le serveur), procédez comme suit :

Sur des systèmes Windows :

- Sélectionnez le menu **Démarrer**, le dossier **Tous les programmes**, le groupe **DB2 Alphablox**, le nom d'instance (la valeur par défaut est **AlphabloxAnalytics**) puis le raccourci d'arrêt de DB2 Alphablox.

Sous Linux et UNIX :

- Exécutez la commande suivante à partir du répertoire `<db2alphablox_dir>/bin`, où `<db2alphablox_dir>` correspond au répertoire dans lequel DB2 Alphablox est installé :

```
./StopAlphablox.sh
```

Arrêt de DB2 Alphablox à partir des services du panneau de contrôle sur des installations Apache Tomcat

Si vous exécutez DB2 Alphablox en tant que service Windows, procédez comme suit pour arrêter DB2 Alphablox :

1. Ouvrez le panneau de contrôle Windows en sélectionnant le menu **Démarrer** Windows puis **Paramètres, Panneau de configuration**.
2. Cliquez deux fois sur **Services**. La fenêtre Services s'affiche.
3. Pour chaque service DB2 Alphablox, sélectionnez le service puis cliquez sur le bouton **Arrêter**.
4. Cliquez sur le bouton **Oui** pour confirmer que vous souhaitez fermer le service.
5. Cliquez sur le bouton **Fermer** pour fermer la fenêtre Services.

Utilisation de l'utilitaire de bibliothèque de client Essbase

Pour utiliser DB2 Alphablox avec DB2 OLAP Server ou des sources de données Hyperion Essbase, vous devez installer les fichiers de bibliothèque client Essbase appropriés et les rendre disponibles pour DB2 Alphablox. Les bibliothèques client Essbase sont disponibles auprès de Hyperion (<http://support.hyperion.com/>).

Pour changer ou mettre à jour les bibliothèques de client Essbase installées sur DB2 Alphablox, procédez comme suit :

1. Sous DB2 Alphablox, ouvrez le répertoire suivant :
`db2alphablox_dir/bin/`, où `db2alphablox_dir` correspond à l'emplacement de votre installation DB2 Alphablox (par exemple, `c:\alphablox\analytics\`).
2. Recherchez le fichier de traitement par lots `ChangeEssbase` (`ChangeEssbase.bat` sous Windows ou `ChangeEssbase.sh` sous Linux et UNIX) puis exécutez-le. Une fenêtre de console de script s'affiche.
3. Entrez le chemin complet de votre installation DB2 Alphablox (par exemple, `c:\alphablox\analytics\`) puis appuyez sur la touche **Entrée**.

4. Une liste des options de bibliothèque de classe disponibles s'affiche. Choisissez l'option appropriée, en fonction de la version Essbase des bibliothèques client à utiliser puis appuyez sur la touche **Entrée**.
5. Si l'opération aboutit, un message s'affiche indiquant que le système a été reconfiguré.
6. Pour que les changements soient pris en compte, redémarrez DB2 Alphablox.

Autres tâches d'administration et informations

Pour plus d'informations sur d'autres tâches d'administration, voir les sections suivantes :

- Chapitre 1, «Présentation de DB2 Alphablox», à la page 1
- Chapitre 19, «Commandes de la console DB2 Alphablox», à la page 141
- Chapitre 3, «Page d'accueil de DB2 Alphablox», à la page 19
- Chapitre 6, «Définitions d'application», à la page 31
- «Définitions des propriétés personnalisées», à la page 82
- Chapitre 10, «Définitions des rôles», à la page 59
- Chapitre 7, «Définitions de sources de données», à la page 43
- Chapitre 8, «Définitions d'utilisateurs», à la page 51
- Chapitre 9, «Définitions de groupes», à la page 55

Chapitre 5. Administration du client

Les administrateurs doivent connaître certains points liés à l'utilisation du client DHTML avec des applications DB2 Alphablox.

Administration du client DHTML

Lors de l'utilisation du mode de rendu DHTML avec DB2 Alphablox, des applications analytiques sont affichées via les fonctions intégrées des navigateurs Web Mozilla Firefox et Microsoft Internet Explorer pris en charge. Par conséquent, une administration minimale est requise pour la prise en charge des applications DHTML.

Configuration client DHTML

Consultez la section Configuration requise du document *Guide d'installation* pour obtenir des informations sur les configurations client DHTML prises en charge des navigateurs Web Mozilla Firefox et Microsoft Internet Explorer.

Remarques relatives au client DHTML

En général, très peu d'incidents surviennent lorsque vos utilisateurs emploient Mozilla Firefox et Microsoft Internet Explorer avec les applications DB2 Alphablox. Vous trouverez ci-dessous une liste des incidents pouvant survenir.

Feuilles de style en cascade (CSS)

Très rarement, les utilisateurs définissent des préférences personnelles pour les paramètres CSS (Feuilles de style en cascade) sous les options avancées disponibles dans les navigateurs Web Mozilla Firefox et Microsoft Internet Explorer. Etant donné la nature en cascade de CSS et l'utilisation intensive de CSS dans le client DHTML, il est possible que les modifications apportées à ces paramètres influent sur l'apparence des composants blox dans ces navigateurs.

Fenêtres en incrustation

Lorsque des utilisateurs cliquent sur des boutons ou sur des liens lors de l'utilisation du client DHTML, l'affichage des fenêtres en incrustation attendues incluant des informations supplémentaires ou des boîtes de dialogue peut ne pas aboutir. La fonction de blocage des fenêtres en incrustation intégrée aux navigateurs Web peut empêcher l'affichage de telles fenêtres. De plus, certains outils de navigateur facultatifs (par exemple, la barre d'outils Google) et des logiciels de pare-feu (Zone Alarm, par exemple) peuvent inclure des paramètres ayant une influence sur le comportement des fenêtres en incrustation des navigateurs Web. Pour remédier à cette situation, l'utilisateur doit désactiver cette fonctionnalité ou changer les préférences utilisateur afin d'autoriser les fenêtres en incrustation sur les applications DB2 Alphablox.

Chapitre 6. Définitions d'application

Le présent chapitre décrit la structure des applications J2EE et fournit des procédures permettant de créer, de modifier et de supprimer des définitions d'application dans DB2 Alphablox.

Définition d'application DB2 Alphablox

Une liste des applications DB2 Alphablox s'affiche sous la page **Applications** qui se trouve sous l'onglet **Administration**. Pour définir une nouvelle application DB2 Alphablox, cliquez sur le bouton **Create**. Pour modifier ou supprimer une définition d'application, sélectionnez une application dans la liste et appuyez sur le bouton **Edit** ou **Delete**.

Lorsque vous créez une application à l'aide des pages d'administration DB2 Alphablox, DB2 Alphablox crée une définition d'application dans le référentiel DB2 Alphablox. Le nom d'application dans le référentiel DB2 Alphablox se compose de deux parties, le nom de l'application d'entreprise et un identificateur unique (URI). Par exemple, si vous créez une application appelée MonApp, l'application aura le nom MonApp_MonApp dans le référentiel. Pour les serveurs d'applications WebSphere et WebLogic, DB2 Alphablox crée des applications installables qui doivent être importées dans le serveur d'applications.

Ci-dessous se trouve l'arborescence d'une application DB2 Alphablox standard importée dans un serveur d'applications WebSphere :

```
<application>.ear
  META-INF/
    application.xml
    MANIFEST.MF
  <application>.war
    META-INF/
      MANIFEST.MF
    WEB-INF
      timeschema.dtd
      timeschema.xml
      web.xml
      classes/
      lib/ (pour les applications Alphablox 8.4.1 uniquement)
        aastaglibs.jar
      tlds/ (pour les applications Alphablox 8.4 uniquement)
        blox.tld
        bloxform.tld
        bloxlogic.tld
        bloxui.tld
```

Les fichiers d'application, incluant des fichiers JSP, des fichiers JavaScript, des fichiers HTML, des fichiers CSS et des fichiers image sont généralement placés dans le répertoire <application>.war. En fonction de la méthode de développement utilisée, certains fichiers peuvent être placés dans des dossiers imbriqués dans le répertoire <application>.war. Les classes Java personnalisées, incluant des composants JavaBeans, sont placées dans le répertoire classes. Les fichiers TLD (tag library descriptor) requis pour les applications DB2 Alphablox se trouvent dans le répertoire tlds dans les applications DB2 Alphablox 8.4. Dans les applications DB2 Alphablox 8.4.1, les fichiers TLD sont intégrés à *aastaglibs.jar*, qui se trouve dans le répertoire *lib* de l'application. Les fichiers de schéma temporel (*timeschema.dtd* et *timeschema.xml*), requis pour certaines applications spécialisées,

se trouvent directement dans le répertoire *WEB-INF*. Lorsque DB2 Alphablox crée une applications pouvant être installée, le fichier du descripteur de déploiement (*web.xml*) est également modifié selon les besoins des applications DB2 Alphablox.

Noms d'application

Le nom d'application utilisé dans le référentiel DB2 Alphablox se compose du nom de l'application d'entreprise et d'un identificateur unique. Une fois que vous avez créé une application, vous pouvez accéder à la vue d'édition d'une application spécifique pour afficher les informations relatives aux noms concernant votre application. Cette liste inclut le nom DB2 Alphablox, le nom de l'application J2EE, le nom du module, le chemin du contexte et le nom d'affichage.

Répertoire WEB-INF

Chaque application Web contient un répertoire *WEB-INF*. Ce répertoire, généré automatiquement lors de la création de l'application, contient des métadonnées sur l'application. Lors d'un fonctionnement normal, il n'est pas nécessaire d'effectuer de modification sous le répertoire *WEB-INF*. Lors de la création d'une application, les fichiers suivants sont installés dans le répertoire *WEB-INF* :

- *web.xml*
- *aastaglibs.jar* (dans le répertoire *lib*) dans DB2 Alphablox 8.4.1 ou en tant que fichiers TLD individuels (par exemple, *blox.tld*) dans DB2 Alphablox 8.4
- *timeschema.dtd*
- *timeschema.xml*

Le fichier de descripteur de déploiement de l'application (*web.xml*) est un fichier XML standard dans toutes les applications J2EE. Il contient un marquage décrivant les attributs de l'application. Le serveur d'applications lit le fichier *web.xml* lors de l'initialisation de l'application.

Important : Les modifications apportées au fichier *web.xml* sont automatiquement effectuées par DB2 Alphablox lors de la mise à niveau du produit. Soyez vigilant si vous devez modifier le fichier *web.xml* car des entrées incorrectes peuvent provoquer un comportement inattendu de l'application.

Les fichiers TLD Alphablox, regroupés dans le fichier *aastaglibs.jar* dans Alphablox 8.4.1 (ou fichier TLD dans Alphablox 8.4), permettent aux développeurs d'accéder aux bibliothèques de balises DB2 Alphablox utilisées lors de la création des composants Blox et des applications DB2 Alphablox. Deux fichiers liés au temps (*timeschema.dtd* et *timeschema.xml*), stockés directement dans le répertoire *WEB-INF* sont disponibles pour les développeurs. Ils peuvent utiliser ces fichiers lors de la création d'applications spécialisées traitant des périodes calendaires et fiscales. Pour plus d'informations sur ces fichiers, consultez les documents *Developer's Guide* et *Developer's Reference*

Définition d'une application

Procédez comme suit pour définir une nouvelle application :

Remarque : Si vous utilisez WebSphere en tant que serveur d'applications, des étapes supplémentaires sont requises pour la création d'une application. Pour plus de détails, voir «Définition d'une application lors de l'exécution avec WebSphere», à la page 34.

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Applications**.
4. Cliquez sur le bouton **Create**. La page **Create Application** s'affiche.
5. Entrez un nom unique pour votre application. Vous pouvez utiliser les caractères A-Z, a-z, 0-9, le trait de soulignement (*_*) ainsi que des caractères spéciaux (par exemple, les caractères accentués) lors de l'exécution dans une langue autre que l'anglais. L'affichage du nom tient compte de l'utilisation des minuscules et des majuscules contrairement à l'authentification du nom réel. Les noms *Public*, *Private* et *Properties* sont réservés et ne peuvent pas être utilisés pour un nom d'objet.
6. [Facultatif] Entrez une brève description de votre application dans la zone **Description**.
7. [Facultatif] Entrez une adresse d'URL, relative au nom de l'application, qui identifie l'emplacement de la page d'origine de cette application (**Home URL**) et une autre URL relative désignant une image d'identification pour cette application (**Image URL**). Ces deux valeurs permettent de créer une entrée sous l'onglet **Applications** à partir de laquelle les utilisateurs peuvent lancer l'application.
8. Dans la zone **Default Saved State**, entrez le nom du paramètre définissant l'état enregistré souhaité. Le nom du paramètre est créé par le développeur d'applications s'il souhaite que les utilisateurs aient accès à une version enregistrée de l'application. Si aucun état enregistré par défaut n'est indiqué, l'état restauré correspond à l'état auquel se trouvait l'application lors de l'arrêt ou de l'expiration du navigateur.
9. Dans la zone de texte **Write Privileges Security Role**, entrez le nom d'un rôle existant pour l'accès à cette application. Si vous laissez cette zone vide, les privilèges d'accès par défaut sont appliqués.
10. Entrez une valeur dans la zone **Session Timeout** pour définir le délai d'expiration de la session J2EE, qui correspond au délai d'inactivité avant le délai d'expiration d'une session utilisateur. La valeur du délai d'expiration est exprimé en minutes.
11. Sélectionnez **Yes** ou **No** dans la liste déroulante **Restore Saved Application State Enabled** pour indiquer si le dernier état automatiquement enregistré doit être chargé.
12. L'option **Include on Applications Page** détermine si les utilisateurs verront cette application sur la vue de la page Application. La valeur par défaut est **Yes**.
13. [Facultatif] Définissez le **mode de rendu par défaut** pour votre application. Par défaut, une nouvelle application prend en charge le mode de rendu DHTML et installe des copies des fichiers TLD Alphablox dans le répertoire de la nouvelle application.
14. [Facultatif] Utilisez la fonction des liens d'en-tête pour configurer des liens à partir des membres affichés sur une grille pour l'ouverture d'une URL ou pour l'exécution de fonctions JavaScript. Ajoutez des adresses URL et des noms de membre uniques ou un autre élément de texte d'URL valide dans la zone de texte **Header Links**, en insérant un saut de lien entre chaque entrée avec la syntaxe suivante :

Membre = URL

Par exemple, pour créer un lien entre `http://www.ibm.com` et un membre nommé IBM, entrez la chaîne suivante dans la zone de texte Header Links :
IBM = `http://www.ibm.com`

15. Cliquez sur le bouton **Save** pour définir la nouvelle application puis revenir à la page Applications.
16. Si vous exécutez une configuration Apache Tomcat avec un serveur Web externe (Apache HTTP Server, Microsoft IIS ou Sun iPlanet), vous devez effectuer des tâches supplémentaires avant que l'application ne soit disponible. Pour plus de détails, voir «Enregistrement d'applications avec des serveurs Web externes», à la page 38.

Définition d'une application lors de l'exécution avec WebSphere

Si vous utilisez IBM WebSphere Application Server, une application doit être définie dans WebSphere, soit avant, soit après la définition de l'application dans DB2 Alphablox. La présente section décrit la procédure permettant de créer une application Web dans WebSphere. Pour plus de détails sur la création d'applications à l'aide de WebSphere, consultez la documentation WebSphere.

Il existe deux moyens de définir une application DB2 Alphablox lors de l'exécution avec WebSphere :

- «Création d'une application dans DB2 Alphablox puis enregistrement dans WebSphere»
- «Importation d'une application WebSphere existante dans DB2 Alphablox», à la page 36

Les deux méthodes sont acceptables. Choisissez celle que vous préférez.

Création d'une application dans DB2 Alphablox puis enregistrement dans WebSphere

Si vous créez une application dans DB2 Alphablox exécuté avec WebSphere, DB2 Alphablox crée la structure de l'application dans un fichier EAR. Ce dernier est créé dans le répertoire suivant :

```
<db2alphablox_dir>/installableApps/
```

Le nom du fichier est `<application>.ear`, où `<application>` correspond au nom indiqué pour l'application lors de la création de cette dernière.

Avant de pouvoir utiliser une application nouvellement créée, vous devez suivre la procédure décrite ci-dessous pour créer la structure d'application dans DB2 Alphablox puis l'enregistrer avec WebSphere :

1. Dans DB2 Alphablox, définissez tout d'abord une nouvelle application, comme il est décrit à la section «Définition d'une application», à la page 32. Un fichier EAR est créé dans le répertoire `<db2alphablox_dir>/installableApps`.
2. Ouvrez la console d'administration WebSphere puis sélectionnez **Applications > Installation d'une nouvelle application**.
3. Sur l'écran **Préparation de l'installation de l'application**, cliquez sur le bouton Parcourir et choisissez le chemin suivant :
`<db2alphablox_dir>/installableApps/<nouvelleApplication>.ear`
où `<application>` correspond au nom de l'application à installer puis cliquez sur Suivant.

Remarque : Environnements groupés uniquement : Il peut être nécessaire d'utiliser le paramètre du chemin de serveur pour rechercher le fichier <nouvelleApplication>.ear sur le réseau.

4. L'écran suivant affiche les options de liaison par défaut. Conservez les paramètres par défaut sauf si vous avez besoin d'autres liaisons puis cliquez sur Suivant.
5. L'écran **Avertissements concernant la sécurité de l'application** s'affiche. Au bas de la page, cliquez sur le bouton Continuer.
6. La section suivante, **Installation d'une nouvelle application**, est composée des cinq étapes suivantes, comme il est affiché dans la boîte de dialogue :

Etape 1 : Fournir des options pour effectuer l'installation

Vous pouvez conserver les paramètres existants et cliquez sur Suivant.

Etape 2 : Mapper les hôtes virtuels pour les modules Web

Acceptez les valeurs existantes et cliquez sur Suivant.

Etape 3 : Mapper les modules vers les serveurs d'applications

Acceptez les valeurs existantes et cliquez sur Suivant.

Etape 4 : Mapper les rôles de sécurité vers les utilisateurs/groupes

Deux rôles, AlphabloxAdministrator et AlphabloxUser doivent apparaître. Vous devez ajouter au moins un utilisateur pour chaque rôle.

Pour le rôle AlphabloxAdministrator, sélectionnez la case à cocher en regard de ce rôle puis appuyez sur les boutons **Rechercher les utilisateurs** ou **Rechercher les groupes** pour ajouter des utilisateurs administratifs. Vous devez sélectionner au moins un utilisateur. Une fois des groupes ou des utilisateurs ajoutés, ils doivent être indiqués dans la zone Utilisateurs mappés ou Groupes mappés de ce rôle.

Pour le rôle AlphabloxUser, sélectionnez la case à cocher sous la colonne Tous les authentifiés. Ainsi, tous les utilisateurs authentifiés ont accès aux applications. Vous pouvez également utiliser ces paramètres pour restreindre l'accès à des utilisateurs ou groupes spécifiques en fonction des exigences de votre application.

Lorsque vous avez terminé, cliquez sur Suivant.

Etape 5 : Récapitulatif

Au bas de cet écran, cliquez sur le bouton Terminer.

7. L'application est maintenant installée et le message "Application <application> installée avec succès" s'affiche. Cliquez sur le lien **Sauvegarde dans la configuration maîtresse**.
8. La boîte de dialogue **Sauvegarde dans la configuration maîtresse** doit s'afficher. Cliquez sur le bouton Sauvegarder. Après une courte période d'attente, vous revenez à la page d'accueil de la console d'administration.
9. Dans la console d'administration WebSphere, ouvrez **Applications > Applications d'entreprise** puis redémarrez votre application.

Vous pouvez maintenant ajouter du contenu, utiliser l'application et modifier les propriétés des pages d'accueil DB2 Alphablox.

Importation d'une application WebSphere existante dans DB2 Alphablox

Pour importer une application WebSphere dans DB2 Alphablox en ajoutant les paramètres et les fichiers DB2 Alphablox requis, suivez la procédure décrite à la section «Importation d'une application J2EE existante», à la page 40.

Modification d'une définition d'application existante

La présente section décrit comment modifier une application existante. Suivez la procédure ci-dessous pour modifier une application existante :

1. Connectez-vous à la page d'accueil DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Applications**.
4. Sélectionnez l'application à modifier dans la liste des applications puis cliquez sur le bouton **Edit**. La page **Edit Application** s'affiche.
5. Modifiez les valeurs des propriétés appropriées.

Remarque : Configurations WebSphere et WebLogic : Vous pouvez renommer les applications en modifiant la zone **Name** mais le changement du nom de l'application change le nom de l'application uniquement dans le référentiel DB2 Alphablox et ne peut pas modifier le chemin de contexte ou le nom d'application WebSphere ou WebLogic associé. Pour modifier les propriétés associées de l'application dans WebSphere ou WebLogic, utilisez la console d'administration du serveur d'applications afin de renommer le chemin de contexte d'application sous-jacent, d'utiliser le nouveau nom et de renommer le répertoire de base des documents de l'application.

Remarque : Configurations Apache Tomcat : Les applications peuvent être renommés en modifiant la zone **Name**. Lorsque vous renommez une application, le nom de répertoire de l'application dans le référentiel DB2 Alphablox ainsi que tous les signets qui référencent l'application sont renommés.

6. Cliquez sur le bouton **Save** pour changer les valeurs des propriétés de l'application et revenir à la page **Applications**.
7. Si vous exécutez une configuration Apache Tomcat avec un serveur Web externe (Apache HTTP Server, Microsoft IIS ou Sun iPlanet) et que vous renommez l'application, vous devez effectuer des tâches supplémentaires pour votre configuration. Pour plus de détails, voir «Enregistrement d'applications avec des serveurs Web externes», à la page 38.

Suppression d'une définition d'application existante

Suivez la procédure ci-dessous pour supprimer une application existante :

1. Connectez-vous à la page d'accueil DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Applications**.
4. Sélectionnez l'application à supprimer de la liste des applications puis cliquez sur le bouton **Delete**.

Important : La suppression de l'application supprime la définition de l'application de DB2 Alphablox. Le nom de l'application n'apparaît donc plus dans les listes d'applications. Toutefois, le dossier de l'application et ses fichiers associés ne sont pas supprimés et doivent l'être manuellement.

Important : Suppression d'applications DB2 Alphablox sous WebLogic : Si le déploiement automatique est activé, procédez comme suit pour supprimer intégralement une application DB2 Alphablox sur le serveur WebLogic :

- a. Utilisez la console d'administration WebLogic pour désactiver la fonction de déploiement automatique.
 - b. Supprimez la définition de l'application de DB2 Alphablox. DB2 Alphablox va annuler l'enregistrement de l'application sur le serveur WebLogic.
 - c. Supprimez le dossier de l'application.
 - d. Si vous le souhaitez, activez à nouveau la fonction de déploiement automatique.
Si cette fonction est activée, seules les étapes b et c sont requises.
5. Si vous exécutez une configuration Apache Tomcat avec un serveur Web externe (Apache HTTP Server, Microsoft IIS ou Sun iPlanet), vous devez effectuer les tâches de configuration requises après avoir supprimé l'application. Pour plus de détails, voir «Enregistrement d'applications avec des serveurs Web externes», à la page 38.

Définition d'une application lors de l'utilisation de clusters WebLogic

Si vous utilisez DB2 Alphablox dans un environnement de clusters BEA WebLogic, procédez comme suit pour créer des applications :

1. Définissez une application sur DB2 Alphablox à partir de l'onglet Administration sous les applications, comme il est décrit à la section «Définition d'une application», à la page 32.
2. Copiez l'intégralité du répertoire de l'application nouvellement créée dans le serveur d'administration WebLogic. Le répertoire sous lequel le domaine est au même niveau que le répertoire de l'application constitue un emplacement idéal.
3. Ajoutez tous les fichiers JSP nécessaires dans le répertoire de l'application au même niveau que le répertoire *WEB-INF*.
4. A partir de la console WebLogic, configurez l'application nouvellement ajoutée. Sur le serveur WebLogic, vous devriez voir le nouveau répertoire de l'application, cliquez alors sur **Select** ou accédez au niveau approprié.
5. Sélectionnez votre cluster en tant que cible de déploiement puis cliquez sur le bouton **Configure and Deploy**.
6. Une fois l'application déployée, sélectionnez AlphabloxServer en tant qu'application Web.
7. Sélectionnez l'onglet Deploy et redéployez l'application AlphabloxServer.
8. Le redéploiement va arrêter et redémarrer DB2 Alphablox.
9. Lorsque DB2 Alphablox est redémarré, vous pouvez accéder à vos applications.

Remarque : Il n'est pas nécessaire de copier vos fichiers JSP dans chaque serveur géré car l'enregistrement de l'application Web sur WebLogic effectue cette tâche pour vous. Pour plus d'informations sur cette procédure ou pour effectuer une modification, consultez la documentation WebLogic.

Enregistrement d'applications avec des serveurs Web externes

Si vous exécutez une configuration Apache Tomcat de DB2 Alphablox avec un serveur Web externe, certaines tâches doivent être effectuées lorsque vous créez ou supprimez une application.

Etant donné que le serveur Web est conçu pour servir de manière efficace des pages statiques et que le serveur d'applications est conçu pour servir des pages dynamiques, ces procédures sont nécessaires pour garantir des performances optimales du serveur Web et du serveur d'applications. Sans ces procédures, le serveur d'applications se verrait dans l'obligation d'effectuer un traitement inutile des pages statiques.

Les configurations suivantes sont décrites :

- «Enregistrement d'applications sur le serveur Apache HTTP Server», à la page 38
- «Enregistrement d'applications sur des serveurs Web Microsoft IIS (Internet Information Server)», à la page 39
- «Enregistrement d'applications sur les serveurs Web Sun iPlanet sur Alphablox 8.4», à la page 39

Enregistrement d'applications sur le serveur Apache HTTP Server

DB2 Alphablox 8.4.1 sur Apache Tomcat 5.5

Lorsque vous créez ou supprimez des définitions d'application DB2 Alphablox avec une configuration Apache Tomcat, les changements ne seront pas répercutés dans le serveur Apache HTTP Server tant que les deux serveurs n'auront pas été correctement arrêtés et redémarrés. Lorsque DB2 Alphablox est redémarré, il place un nouveau fichier de configuration Apache (`mod_jk.conf`) dans le serveur Apache HTTP Server.

Pour enregistrer les ajouts ou les suppressions d'application DB2 Alphablox, procédez comme suit :

1. Arrêtez les serveurs Apache HTTP Server et Apache Tomcat (exécutant DB2 Alphablox)
2. Redémarrez le serveur Apache Tomcat.
3. Redémarrez le serveur Apache HTTP Server.

Après avoir redémarré les serveurs, les changements apportés à l'application seront reconnus sur le serveur Apache HTTP Server.

DB2 Alphablox 8.4 sur Apache Tomcat 3.2.4

Si vous exécutez une configuration Apache Tomcat à l'aide d'Apache HTTP Server, vous devez redémarrer le serveur Web Apache dès que vous créez ou supprimez une définition d'application DB2 Alphablox. Le redémarrage est nécessaire afin qu'Apache puisse lire les changements effectués par DB2 Alphablox dans le fichier de configuration Apache (`mod_jk.conf-alphablox`) lorsque vous créez ou supprimez une définition d'application DB2 Alphablox.

Arrêt et redémarrage des serveurs Apache HTTP Server

Pour arrêter Apache, exécutez la commande shutdown lors de l'exécution dans une fenêtre de console ou lors de l'exécution en tant que service Windows, sélectionnez le service dans la section **Services** du panneau de configuration et cliquez sur le bouton **Arrêter**.

Pour redémarrer Apache, exécutez la commande startup lors de l'exécution dans une fenêtre de console ou lors de l'exécution en tant que service Windows, sélectionnez le service dans la section **Services** du panneau de configuration et cliquez sur le bouton **Démarrer**.

Pour plus de détails sur l'arrêt et le démarrage du serveur Web Apache, consultez la documentation Apache.

Enregistrement d'applications sur des serveurs Web Microsoft IIS (Internet Information Server)

Si vous exécutez une configuration autonome à l'aide du serveur Web IIS, vous devez exécuter deux tâches dès que vous créez ou supprimez une définition d'application DB2 Alphablox :

- Vous devez ajouter ou supprimer le répertoire virtuel pour l'application DB2 Alphablox.
- Vous devez redémarrer le serveur Web.

Enregistrement d'applications sur les serveurs Web Sun iPlanet sur Alphablox 8.4

Si vous exécutez une configuration autonome à l'aide du serveur Web Sun iPlanet sur DB2 Alphablox 8.4 (et non 8.4.1), vous devez exécuter deux tâches dès que vous créez ou supprimez une définition d'application DB2 Alphablox :

- «Ajout du répertoire de stockage»
- «Ajout des affectations appropriées au style DB2 Alphablox», à la page 40

Ajout du répertoire de stockage

Pour chaque application DB2 Alphablox, vous devez ajouter le répertoire de stockage au serveur Web afin qu'il serve des pages à partir de ce répertoire. Les rubriques suivantes décrivent comment ajouter le répertoire d'application au serveur Web iPlanet.

1. Ouvrez les pages d'administration du serveur Web iPlanet. Par défaut, ces pages sont accessibles via le port 8888 sur le serveur où iPlanet est en cours d'exécution mais le port peut être différent sur le système.
2. Sélectionnez le serveur dans la liste déroulante et cliquez sur le bouton **Manage**.
3. Cliquez sur le lien **Class Manager** dans le coin supérieur droit.
4. Cliquez sur l'onglet **Content Mgmt**.
5. Cliquez sur le lien **Additional Document Directories**.
6. Dans la zone de texte **URL prefix**, entrez le nom de contexte de votre application suivi d'une barre oblique (/), comme dans l'exemple suivant :
MonApplication/

Important : Ne placez pas de barre oblique devant le nom du contexte, ce caractère pouvant corrompre le fichier de configuration iPlanet.

7. Dans la zone de texte **Map to Directory**, entrez le répertoire de votre application, comme dans l'exemple suivant :
d:/Alphablox4/webapps/MonApplication

Conseil : Utilisez uniquement des barres obliques (/) pour le nom de chemin. Certaines version d'iPlanet ne gèrent pas correctement les barres obliques inversées (\) dans les noms de chemin.

8. Choisissez **NONE** dans la liste déroulante **Apply Style**.
9. Cliquez sur **OK** puis enregistrer vos changements.

Ajout des affectations appropriées au style DB2 Alphablox

Pour chaque application, vous devez affecter trois caractères génériques de préfixe d'URL au style **alphablox**. Ces affectations transmettent les requêtes appropriées à DB2 Alphablox. Vous trouverez ci-dessous la procédure permettant d'ajouter les nouvelles affectations au style **alphablox**.

1. Ouvrez les pages d'administration du serveur Web iPlanet. Par défaut, ces pages sont accessibles via le port 8888 sur le serveur où iPlanet est en cours d'exécution mais le port peut être différent sur votre système.
2. Sélectionnez le serveur dans la liste déroulante et cliquez sur le bouton **Manage**.
3. Cliquez sur l'onglet **Styles**.
4. Cliquez sur le lien **Assign Style**.
5. Choisissez le style **alphablox** dans la liste déroulante.
6. Si votre application est une application standard DB2 Alphablox, ajoutez trois **caractères génériques de préfixe d'URL** à attribuer au style **alphablox** :
*MonApplication/abx/**
*MonApplication/servlet/**
MonApplication/.jsp*
où *MonApplication* correspond au nom de contexte de votre application.
7. Si votre application utilise la compatibilité 3.x, ajoutez le **caractère générique de préfixe d'URL** à attribuer au style **alphablox** et non ceux utilisés à l'étape précédente :
*MonApplication/**
où *MonApplication* correspond au nom de contexte de votre application.
8. Enregistrez et appliquez ces changements pour toutes les demandes de transmission d'URL.
9. Si vous souhaitez confirmer que vous avez ajouté des affectations correctement, cliquez sur le lien **List Assignments** et examinez la liste des affectations pour le style **alphablox**.

Importation d'une application J2EE existante

Vous pouvez importer une application J2EE pour la définir en tant qu'application DB2 Alphablox. Après avoir importé une application J2EE dans DB2 Alphablox, une application peut accéder à tous les services d'application DB2 Alphablox, tels que le rendu de Blox, les signets, etc.

L'importation d'applications J2EE est une méthode commune permettant de rendre les applications disponibles lors de l'exécution avec WebSphere. Pour plus de détails sur la définition d'une application lors de l'exécution avec WebSphere, voir «Définition d'une application lors de l'exécution avec WebSphere», à la page 34.

Remarque : Lors de l'importation d'applications Apache Tomcat existantes qui ont été créés avant l'installation de DB2, vous devez également ajouter `crossContext=true` à divers emplacements, notamment dans le fichier `$CATALINA_HOME/conf/context.xml`. Pour plus d'informations, voir <http://tomcat.apache.org/tomcat-5.5-doc/config/context.html>.

Pour importer une application J2EE existante dans DB2 Alphablox, procédez comme suit :

1. Vous devez connaître le nom du contexte d'application d'une application J2EE existante que vous souhaitez importer dans DB2 Alphablox.
2. Connectez-vous à la page d'accueil DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe des administrateurs.
3. Cliquez sur l'onglet **Administration**.
4. Cliquez sur le lien **Applications**.
5. Cliquez sur le bouton **Create**.
6. Sélectionnez la valeur **Yes** pour la liste déroulante **Import J2EE Application**. La fenêtre **Importing Existing J2EE Application** s'affiche.
7. Appuyez sur le bouton **Lookup Application**. La fenêtre **Import an Application** s'affiche.
8. Effectuez des sélections dans les listes déroulantes **J2EE Application Name** (non disponible dans Apache Tomcat) et **Module Name**. Les zones **URI** et **Context Path** en lecture seule affichent des valeurs en fonction de vos sélections.
9. Cliquez sur le bouton **Select Application**. La fenêtre de boîte de dialogue se ferme et le formulaire **Importing Existing J2EE Application** est automatiquement rempli.
10. Cliquez sur **Save** pour terminer l'importation de l'application J2EE sélectionnée. Cliquez sur **Annuler** pour revenir à la page principale **Application** ou sélectionnez **No** sous **Import J2EE Application** pour revenir à la fenêtre **Create Application**.

Après avoir importé l'application J2EE, la nouvelle application s'affiche dans la liste des applications. L'application a la même structure que celle décrite à la section «Définition d'application DB2 Alphablox», à la page 31.

Chapitre 7. Définitions de sources de données

Le présent chapitre contient des informations administratives sur les sources de données relationnelles et multidimensionnelles et décrit les procédures permettant de créer, de modifier et de supprimer des définitions de source de données dans DB2 Alphablox.

Définition d'une source de données

DB2 Alphablox prend en charge des sources de données relationnelles et multidimensionnelles. La page **Data Source** de l'onglet **Administration** permet de définir une source de données pour l'accès d'une ou plusieurs applications DB2 Alphablox. Les applications DB2 Alphablox utilisent ces sources de données pour accéder aux bases de données relationnelles et multidimensionnelles. La page **Data Sources** sous l'onglet **Administration** change en fonction du type de source de données définie

Pour définir une source de données, procédez comme suit :

1. Connectez-vous à la page d'accueil DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Data Source**.
4. Cliquez sur le bouton **Create**. La page **Create Data Source** s'affiche.
5. Dans la zone de texte **Data Source Name**, entrez un nom unique (obligatoire) et une description de la source de données. Vous pouvez utiliser les caractères A-Z, a-z, 0-9, le trait de soulignement (_) ainsi que des caractères spéciaux (par exemple, les caractères accentués) lors de l'exécution dans une langue autre que l'anglais. L'affichage du nom tient compte de l'utilisation des minuscules et des majuscules contrairement à l'authentification du nom réel. Les noms *Public*, *Private* et *Properties* sont réservés et ne peuvent pas être utilisés pour un nom d'objet.
6. Dans la liste déroulante **Adapter**, sélectionnez l'adaptateur de base de données à utiliser pour la connexion à la source de données.

Remarque : DB2 Alphablox inclut un adaptateur de données préconfiguré permettant de tester une connexion entre le client et le serveur. Si vous suspectez une situation anormale, contactez le service clients DB2 Alphablox pour obtenir de l'aide pour l'utilisation de cet adaptateur.

7. Dans la zone de texte **Server Name**, indiquez le nom de la machine hôte sur laquelle se trouve la source de données. (Certains noms sont différents en fonction des différents adaptateurs de données.)
8. Indiquez d'autres informations propres à l'adaptateur permettant d'accéder à la source de données, par exemple : **Default Catalog**, **Port Number**, **Database (catalog)**, **Schema**, **SID**, **Username** et **Password**.

Remarque : Pour les installation de serveur d'applications WebSphere et WebLogic, **Password** est remplacé **Data Source Password** car ce mot de passe ne permet pas d'authentifier les applications. Dans ce cas, l'authentification utilisateur est gérée par votre serveur WebSphere ou WebLogic. Lorsque l'option **Use DB2 Alphablox**

Username and Password est activée sur vos applications, le mot de passe est utilisé uniquement pour l'authentification de la source de données.

Remarque : Utilisateurs Microsoft SQL Server : Prenez en compte le fait que le nom d'une base de données respecte les règles du format des identificateurs dans Microsoft SQL Server. Il existe deux classes d'identificateur : identificateurs ordinaires et identificateurs délimités. Tout identificateur qui ne respecte pas les règles du format de l'identificateur ordinaire est un identificateur délimité qui doit toujours être délimité par des guillemets ou des crochets. Par exemple, si votre base de données MS SQL se nomme test-1, le nom de la base de données (catalogue) dans le nom de source de données DB2 Alphablox doit être [test-1] (les crochets sont inclus dans le nom de la source de données) ou "test-1" (les guillemets sont inclus dans le nom de la source de données).

9. Si vous utilisez une source de données Microsoft SQL Server, vous devez indiquer un utilisateur de base de données défini en tant qu'utilisateur authentifié SQL Server. Les utilisateurs SQL Server qui utilisent l'authentification Windows ne fonctionnent pas correctement avec DB2 Alphablox.
10. Indiquez des valeurs dans les zones **Maximum Rows** et **Maximum Columns** si vous souhaitez limiter la taille d'un ensemble de résultats de requêtes provenant de cette source de données.

Remarque : N'attribuez pas de valeur inférieure à 20. La valeur par défaut 1000 devrait être suffisante pour la plupart des applications. Si une requête de grande taille est tronquée, une boîte de dialogue d'erreur s'affiche dans le navigateur.

11. Si vous utilisez une source de données Oracle, indiquez le nombre de lignes devant être extraites en préparation de l'affichage dans la zone **Row Prefetch**. La valeur par défaut est 100.
12. Cliquez sur le bouton **Save** pour définir la nouvelle source de données et revenir à la page **Data Sources**.

Changement ou suppression d'une définition de source de données existante

La présente section décrit comment modifier ou supprimer une source de données existante.

Changement d'une définition de source de données existante

Pour modifier une source de données existante, procédez comme suit :

1. Connectez-vous à la page d'accueil DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Data Source**.
4. Cliquez sur le bouton **Edit**. La page **Edit Data Source** s'affiche.
5. Changez les valeurs appropriées.
6. Cliquez sur le bouton **Save** pour changer la définition et revenir à la page **Data Sources**.

Suppression d'une définition de source de données existante

Pour supprimer une source de données existante, procédez comme suit :

1. Connectez-vous à la page d'accueil DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**.
3. Cliquez sur le lien **Data Source**. La page **Data Source** s'affiche.
4. Cliquez sur le bouton **Delete** pour supprimer la définition de source de données et revenir à la page **Data Sources**.

Important : La suppression d'une source de données la retire de manière définitive de DB2 Alphablox.

Configuration de l'authentification Microsoft pour les sources de données Microsoft Analysis Services

Les serveurs Microsoft Analysis Services, incluant Microsoft SQL Server 2000 Analysis Services et Microsoft SQL Server 2005 Analysis Services, utilisent l'authentification Windows. Pour utiliser l'authentification Windows dans les applications DB2 Alphablox qui accèdent à Microsoft Analysis Services, les propriétés `userName` et `password` transmises par DB2 Alphablox à Microsoft Analysis Services doivent être des utilisateurs et des mots de passe Windows. Si DB2 Alphablox et Microsoft Analysis Services se trouvent dans des domaines différents, ces derniers doivent être des domaines Windows sécurisés.

Pour activer DB2 Alphablox de telle sorte que les utilisateurs se connectent correctement à Microsoft Analysis Services, l'utilisateur Windows dans lequel DB2 Alphablox s'exécute (c'est-à-dire, l'utilisateur Windows connecté lors du démarrage de DB2 Alphablox) doit disposer du droit utilisateur "Agir en tant que partie du système d'exploitation". Si vous exécutez DB2 Alphablox en tant que service Windows, vous pouvez configurer le service afin qu'il s'exécute sous un utilisateur Windows par défaut. La présente section décrit les procédures permettant de configurer votre machine Windows afin d'utiliser cette forme d'authentification Microsoft Analysis Services. Elle contient les procédures suivantes :

- «Configuration des droits utilisateur Windows»
- «Configuration du service Windows», à la page 47
- «Vérification que les utilisateurs sont configurés dans Microsoft Analysis Services», à la page 47

Configuration des droits utilisateur Windows

La présente section décrit les étapes nécessaires à la configuration de votre machine Windows pour accéder aux données dans une base de données Microsoft Analysis Services.

Remarque : Si vous exécutez DB2 Alphablox dans un environnement en clusters et que vous accédez aux sources de données Microsoft Analysis Services, chaque noeud du cluster doit être configuré pour s'exécuter sous le même utilisateur Windows et chaque machine doit disposer de l'ensemble de règles de sécurité, comme il est décrit dans cette section.

Définition des droits pour l'utilisateur Windows

Pour permettre à DB2 Alphablox de se connecter à Microsoft Analysis Services correctement, l'utilisateur Windows dans lequel DB2 Alphablox s'exécute (c'est-à-dire, l'utilisateur Windows qui est connecté lors du démarrage de DB2

Alphablox ou si DB2 Alphablox s'exécute en tant que service, l'utilisateur sous lequel le service est en cours d'exécution) doit disposer du droit utilisateur "Agir en tant que partie du système d'exploitation".

Remarque : L'utilisateur Windows doit être configuré dans Microsoft Analysis Services.

Sous Windows, procédez comme suit pour activer ce droit pour l'utilisateur sous lequel DB2 Alphablox s'exécute :

1. A partir du menu **Démarrer** de Windows, sélectionnez **Tous les programmes > Outils d'administration > Gestionnaire des utilisateurs**.
La fenêtre Gestionnaire des utilisateurs Windows s'affiche.
2. Dans le menu **Stratégies**, sélectionnez **Droits de l'utilisateur**.
3. Sélectionnez la case à cocher **Afficher les droits avancés des utilisateurs**.
4. Dans la liste déroulante **Droit**, sélectionnez **Agir en tant que partie du système d'exploitation**.
5. Cliquez sur la bouton **Ajouter**.
6. Dans la boîte de dialogue **Ajouter des utilisateurs et des groupes**, vérifiez que la liste **Lister les noms de** affiche le nom de domaine ou de machine correct pour le compte à ajouter.
7. Cliquez sur le bouton **Afficher les utilisateurs**.
8. Dans la zone de liste des noms, sélectionnez le compte utilisateur à ajouter.
9. Cliquez sur le bouton **Ajouter**. Le nom d'utilisateur est répertorié dans la zone de liste **Ajouter des noms**.
10. Redémarrez la machine.

Remarque : L'utilisateur qui se trouve dans la source de données DB2 Alphablox qui accède à Microsoft Analysis Services doit correspondre à l'utilisateur Windows disposant du droit utilisateur "Agir en tant que partie du système d'exploitation".

Définition des droits pour l'utilisateur Windows

Sous Windows, procédez comme suit pour activer ce droit pour l'utilisateur sous lequel DB2 Alphablox s'exécute :

1. A partir du menu **Démarrer** de Windows, affichez le dossier du panneau de configuration en sélectionnant **Paramètres > Panneau de configuration**.
2. Ouvrez **Outils d'administration**.
3. Sélectionnez **Stratégie de sécurité locale** dans le dossier **Outils d'administration** du panneau de configuration.
4. Développez le dossier **Stratégies locales** dans la fenêtre **Paramètres de sécurité locaux**.
5. Sélectionnez le dossier **Attribution des droits utilisateur**.
6. Dans le volet de droite de la fenêtre **Paramètres de sécurité locaux**, cliquez deux fois sur le droit utilisateur **Agir en tant que partie du système d'exploitation**.
7. Dans la fenêtre **Paramètre de stratégie de sécurité locale** du droit utilisateur **Agir en tant que partie du système d'exploitation**, cliquez sur le bouton **Ajouter** et sélectionnez l'utilisateur sous lequel DB2 Alphablox s'exécute. Si DB2 Alphablox s'exécute sous un utilisateur réseau, assurez-vous de sélectionner l'utilisateur réseau correct (par exemple, CORPNET\alphablox pour un utilisateur réseau nommé *alphablox* sur le réseau CORPNET Windows).

8. Sélectionnez la case à cocher **Paramètre de stratégie locale** puis cliquez sur **OK**.
9. Fermez la fenêtre **Paramètres de sécurité locaux**.
10. Redémarrez la machine.

Remarque : L'utilisateur dans la source de données DB2 Alphablox qui accède à Microsoft Analysis Services doit être le même utilisateur Windows que celui auquel vous avez accordé le droit utilisateur "Agir en tant que partie du système d'exploitation".

Configuration du service Windows

Pour configurer DB2 Alphablox afin qu'il s'exécute sous un autre utilisateur que celui qui est connecté, vous devez exécuter DB2 Alphablox en tant que service. Si DB2 Alphablox se connecte à Microsoft Analysis Services, l'utilisateur Windows doit être configuré dans Microsoft Analysis Services et doit disposer du droit utilisateur "Agir en tant que partie du système d'exploitation".

Pour configurer le service afin qu'il démarre sous un utilisateur Windows différent, procédez comme suit :

1. A partir du menu Démarrer Windows, sélectionnez **Paramètres**.
2. Sélectionnez **Panneau de configuration** dans le menu déroulant.
3. Cliquez deux fois sur l'icône **Services**.
4. Dans la fenêtre **Services**, cliquez deux fois sur le service **DB2 Alphablox (AlphabloxAnalytics)**.
5. Dans la boîte de dialogue **Service**, sélectionnez le bouton d'option **Ce compte**.
6. Entrez un nom d'utilisateur dans la zone de texte se trouvant à droite.
7. Entrez un mot de passe dans la zone de texte **Mot de passe**.
8. Vérifiez le mot de passe en l'entrant à nouveau dans la zone de texte **Confirmer le mot de passe**.
9. Cliquez sur **OK**.

Vérification que les utilisateurs sont configurés dans Microsoft Analysis Services

Tout utilisateur indiqué dans l'application DB2 Alphablox comme devant se connecter à Microsoft Analysis Services, qu'il soit configuré via une source de données DB2 Alphablox ou défini avec des paramètres de nom d'utilisateur et de mot de passe (ou les méthodes associées) dans l'application doit être un utilisateur Windows et être correctement configuré dans Microsoft Analysis Services. Le groupe de domaine Windows Tout le monde inclut uniquement des utilisateurs de domaine et non des utilisateurs de machine locale.

Remarque : Si vous tentez de vous connecter à Microsoft Analysis Services en tant qu'utilisateur local de votre machine, cet utilisateur doit faire partie du groupe Administrateurs tout comme un administrateur OLAP afin de pouvoir se connecter de manière fiable. Si l'utilisateur ne se trouve pas dans les groupes d'administrateurs, l'utilisateur pourra se connecter à la première tentative mais la tentative suivante n'aboutira pas et une exception du type suivant sera générée :

```
com.alphablox.util.NotAuthorizedException: Impossible de  
trouver le fournisseur. Il peut ne pas avoir été correctement installé
```

Utilisation des sources de données JDBC

La présente section inclut les sections suivantes relatives aux sources de données JDBC :

- «Configuration de l'environnement pour le pilote relationnel Sybase JConnect»
- «Configuration du traçage JDBC»
- «Mise à jour d'un pilote JDBC pris en charge vers une version différente», à la page 49
- «Ajout de pilotes JDBC supplémentaires», à la page 49
- «Modification des paramètres de chemin de classes», à la page 50

Configuration de l'environnement pour le pilote relationnel Sybase JConnect

Si vous utilisez le pilote Sybase JConnect pour la connexion à une base de données relationnelle Sybase, vous devez exécuter un fichier script sur le serveur de base de données Sybase. Les fichiers de script SQL se trouvent dans le répertoire suivant :

```
<db2alphablox_dir>/tools/sql
```

où <db2alphablox_dir> correspond au répertoire dans lequel DB2 Alphablox est installé. Le tableau suivant indique le nom du script SQL et le serveur de base de données Sybase sur lequel le script doit être exécuté.

Script SQL	Serveur de base de données Sybase sur lequel exécuter le script
sql_asa.sql	SQL Anywhere 5.5.02 ou version supérieure
sql_server.sql	Versions SQL Server antérieures à la version 12
sql_server12.sql	SQL Server version 12 et versions ultérieures

Déterminez quel est le serveur de base de données Sybase que vous exécutez et exécutez le script approprié sur le serveur de base de données. Chaque script configure les procédures mémorisées et les objets système requis pour le pilote JDBC Sybase JConnect. Pour plus de détails sur l'exécution d'un fichier de script SQL sur votre serveur de base de données, consultez votre documentation Sybase.

Vous devez exécuter le script SQL approprié sur votre serveur de base de données Sybase afin de pouvoir y établir une connexion à partir de DB2 Alphablox.

Configuration du traçage JDBC

Pour tous les pilotes de source de données relationnelles dans DB2 Alphablox, vous pouvez activer le traçage JDBC. Avec le traçage JDBC activé, le pilote JDBC journalise les messages d'information dans la console DB2 Alphablox et dans le fichier journal. Tous les messages de traçage JDBC sont journalisés au niveau de message INFO.

Pour activer le traçage pour une base de données relationnelle, attribuez la valeur **Yes** à la liste déroulante **JDBC Tracing Enabled**. Cette option est disponible uniquement sur les sources de données qui utilisent un des pilotes de base de données relationnelle pris en charge. Après avoir activé le traçage JDBC, définissez le niveau de message à INFO ou à un niveau plus élevé. Par exemple, dans une fenêtre de console, entrez la commande suivante :

```
report info
```


Utilisez le traçage JDBC pour déboguer les incidents liés aux connexions aux bases de données relationnelle. Lorsque le traçage JDBC est activé, les informations suivantes sont journalisées :

- Données de journalisation du pilote JDBC
- Informations de connexion de DB2 Alphablox aux bases de données relationnelles
- Informations sur les métadonnées de colonne provenant de chaque ensemble de résultats que la base de données renvoie à DB2 Alphablox

Si des erreurs de connexion surviennent ou que vous avez besoin d'informations supplémentaires sur le traçage JDBC, contactez le service clients DB2 Alphablox.

Mise à jour d'un pilote JDBC pris en charge vers une version différente

Si vous devez mettre à jour un des pilotes JDBC pris en charge utilisés par DB2 Alphablox afin d'accéder à une base de données relationnelle, procédez comme suit :

Important : DB2 Alphablox teste uniquement les pilotes JDBC pris en charge répertoriés dans la section Configuration requise du document *Guide d'installation*. Les autres pilotes n'ont pas été testés et peuvent générer des résultats inattendus. Pour plus d'informations, contactez le service clients DB2 Alphablox.

1. Obtenez les classes Java pour le nouveau pilote JDBC. Généralement, ces fichiers ont l'extension zip ou jar (par exemple, `ojdbc14.jar`).
2. Recherchez le répertoire indiqué lors de l'installation pour vos pilotes JDBC.
3. Renommez le fichier du pilote JDBC que vous mettez à jour (par exemple, en lui attribuant le nom `<nomPilote>.zip.old`) afin de conserver une copie si vous souhaitez revenir à la version prise en charge.
4. Renommez le nouveau fichier de pilote afin qu'il porte le même nom que celui des pilotes existants puis copiez-le dans le répertoire `<db2alphablox_dir>/lib/`.
5. Mettez à jour l'environnement de démarrage DB2 Alphablox pour accéder aux nouvelles classes du pilote JDBC, comme il est décrit dans la section suivante.
6. Après avoir mis à jour les informations de chemin de classes, redémarrez le serveur pour que les modifications soient prises en compte.

Ajout de pilotes JDBC supplémentaires

Les sources de données relationnelle disponibles par défaut pour les applications DB2 Alphablox lors de l'installation ou de la mise à niveau sont définies dans le fichier `jdbcdriivers.xml`. Vous ne devez pas modifier ce fichier. Il est automatiquement remplacé lors des mises à niveau de DB2 Alphablox. Si vous devez créer des sources de données DB2 Alphablox supplémentaires, vous pouvez utiliser un fichier `jdbcdriivers_additional.xml` facultatif.

Pour créer des sources de données supplémentaires pour DB2 Alphablox, procédez comme suit :

1. Recherchez le fichier `jdbcdriivers_additional.xml` se trouvant dans le répertoire suivant :
`<db2alphablox_dir>/repository/servers/`
2. Vous pouvez modifier ce fichier mais vous devez effectuer une copie de sauvegarde de ce fichier dans le même répertoire et l'appeler

jdbcdri vers_additional.xml. Une copie de sauvegarde du fichier jdbcdri vers_additional.xml se trouve dans le même répertoire et se nomme jdbcdri vers_additional.xml.example.

3. A l'aide des exemples de pilote JDBC se trouvant dans ce fichier, créez les sources de données supplémentaires dont vous avez besoin.
 - a. Dans chaque définition de pilote JDBC que vous souhaitez rendre disponible, modifiez l'attribut désactivé de true en false. Par défaut, les pilotes JDBC exemple sont désactivés.
4. Vous devez redémarrer le serveur d'applications pour que ces modifications soient prises en compte.
5. Modifiez tous les chemins de classes DB2 Alphablox afin d'inclure votre nouveau pilote, comme il est décrit ci-dessous à la section «Modification des paramètres de chemin de classes», à la page 50.

Modification des paramètres de chemin de classes

Après avoir ajouté des pilotes supplémentaires au fichier jdbcdri vers_additional.xml, vous devez également mettre à jour les informations du chemin de classes afin que les nouveaux pilotes JDBC puissent être trouvés. Pour votre configuration, suivez la procédure ci-dessous.

WebSphere

Pour ajouter ou supprimer des pilotes JDBC sous WebSphere, il n'est pas nécessaire de modifier les paramètres de chemin de classes. Ajoutez des pilotes supplémentaires en les plaçant dans le répertoire <websphere_dir>/AppServer/lib/ext ou supprimez les anciens pilotes ou les pilotes non utilisés du répertoire. Redémarrez le serveur d'applications pour que les modifications apportées au chemin de classes soient prises en compte.

WebLogic

Pour les installations de serveur d'applications BEA WebLogic, les informations de chemin de classes peuvent être modifiées dans le fichier de commandes par lots suivant : <db2alphablox_dir>/bin/jdbcsetup.bat (Windows) ou <db2alphablox_dir>/bin/jdbcsetup.sh (UNIX). Redémarrez le serveur d'applications pour que les modifications soient prises en compte.

Tomcat

Pour modifier les informations de chemin de classes pour DB2 Alphablox s'exécutant sous Apache Tomcat uniquement, procédez comme suit :

Windows : Si Tomcat s'exécute dans une console Windows, ouvrez le fichier <db2alphablox_dir>/bin/jdbcsetup.bat dans un éditeur de texte et modifiez les paramètres de chemin de classes selon vos besoins.

Si le serveur d'applications s'exécute en tant que service Windows, ouvrez le fichier <tomcat_dir>/conf/wrapper.properties (dans votre répertoire Tomcat et non dans le répertoire DB2 Alphablox) dans un éditeur de texte et modifiez les paramètres de chemin de classes selon vos besoins. Redémarrez le serveur d'applications pour que les modifications soient prises en compte.

Linux et UNIX : Si vous utilisez Apache Tomcat avec DB2 Alphablox sur un système Linux ou UNIX, ouvrez le fichier <tomcat_dir>/bin/analytics.sh dans un éditeur de fichier et modifiez les paramètres de chemin de classes selon vos besoins. Redémarrez le serveur d'applications pour que les modifications soient prises en compte.

Chapitre 8. Définitions d'utilisateurs

La page d'administration **Users** de DB2 Alphablox permet de créer, modifier et supprimer des utilisateurs.

Les informations présentes sur les pages **Applications** et **Data Sources** sont également accessibles lorsque vous cliquez sur le lien **Aide** situé dans le coin supérieur droit de chaque page.

Remarque : Lorsque vous modifiez les utilisateurs existants, veillez à cliquer sur le bouton **Save** avant d'accéder à une autre page. Sinon, les modifications effectuées sur la page ne seront pas enregistrées.

Création d'un nouvel utilisateur

Le lien **Users** sous l'onglet **Administration** ouvre les pages permettant de créer et modifier les nouveaux utilisateurs. Au cours de l'installation, deux utilisateurs sont créés : administrateur (Admin) et invité (Guest). Cliquez sur le bouton **Create** sur la page **Users** pour définir les utilisateurs supplémentaires. Sélectionnez un utilisateur existant et cliquez sur **Edit** ou sur **Delete** selon l'action souhaitée. Si la définition d'un utilisateur est endommagée, redémarrez DB2 Alphablox pour la réparer. Cette section indique la procédure à suivre pour définir un nouvel utilisateur sur la page d'accueil de DB2 Alphablox.

Pour définir un nouvel utilisateur, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox sous le nom d'utilisateur *admin* ou sous le nom d'un utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** apparaît.
3. Cliquez sur le lien **Users**. La page **Users** apparaît.
4. Cliquez sur le bouton **Create**. La page **Create User** apparaît.

Remarque : En fonction de la configuration de la sécurité, il se peut que cette page ne contienne pas les informations relatives au mot de passe.

5. Dans la zone **Username**, entrez l'ID de l'utilisateur (obligatoire). Le nom d'utilisateur est le nom utilisé pour la connexion à DB2 Alphablox. Vous pouvez utiliser les caractères A-Z, a-z, 0-9, le trait de soulignement et des caractères spéciaux (par exemple, des caractères accentués). L'affichage du nom d'utilisateur est dépendant des majuscules/minuscules, tandis que le nom de l'utilisateur réel authentifié n'est pas dépendant des majuscules/minuscules. Les noms *Public*, *Private* et *Properties* étant réservés, ils ne peuvent pas être utilisés en tant que nom d'objet.
6. Entrez éventuellement le nom complet de l'utilisateur dans la zone **Full Name**, sa **Description** et son adresse e-mail dans **eMail Address**.
7. Dans la zone **Password**, entrez le mot de passe initial de l'utilisateur. Le mot de passe est dépendant des majuscules/minuscules.
8. Dans la zone **Confirm Password** entrez à nouveau le mot de passe pour le confirmer.

9. Dans la liste déroulante **Allow User To Edit Profile**, indiquez si l'utilisateur a le droit de modifier le profil. Si la valeur de cette configuration est **Yes**, l'utilisateur peut accéder à son profil et le modifier en cliquant sur le lien **My Profile** sur la page d'accueil de DB2 Alphablox.
10. Indiquez une valeur pour chaque propriété personnalisée. (Lorsque des propriétés personnalisées par l'utilisateur sont définies, elles apparaissent dans la liste déroulante **Allow User to Edit Profile**)
11. Cliquez sur le bouton **Save** pour définir le nouvel utilisateur et retourner à la page **Users**.

Remarque : Lorsque vous créez un nouvel utilisateur, il devient automatiquement membre du groupe *Public*.

Modification ou suppression d'un groupe ou d'un utilisateur existant

Cette section indique la procédure à suivre pour modifier les propriétés d'un utilisateur ou d'un groupe existant ou bien supprimer un utilisateur ou un groupe sur la page d'accueil de DB2 Alphablox.

Modification des propriétés d'un utilisateur existant

Pour modifier les propriétés d'un utilisateur existant, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox sous le nom d'utilisateur *admin* ou sous le nom d'un utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** apparaît.
3. Cliquez sur le lien **Users**. La page **Users** apparaît.
4. Cliquez sur le bouton **Edit**. La page **Edit User** apparaît.
5. Modifiez les valeurs si nécessaire.

Remarque : Sur des installations WebSphere et WebLogic, "Password" apparaît sous la forme "Data Source Password" car ce mot de passe n'est pas utilisé pour l'authentification avec des applications. Dans ce cas, l'authentification des utilisateurs est gérée par WebSphere ou WebLogic. Cela signifie également que si le nom d'utilisateur et le mot de passe DB2 Alphablox sont utilisés dans vos applications, le mot de passe n'est utilisé que pour l'authentification de la source de données.

6. Cliquez sur le bouton **Save** pour enregistrer les modifications de l'écran **General Properties** et revenir à la page **Users**.
7. Pour modifier les propriétés de l'application de l'utilisateur, cliquez sur l'onglet **Application Properties**. Effectuez les modifications nécessaires, puis cliquez sur le bouton **Save**.
8. Pour modifier l'appartenance à un groupe ou un rôle d'un utilisateur, cliquez sur l'onglet **Memberships**. Effectuez les modifications nécessaires. Notez que les modifications apportées à l'appartenance sont appliquées immédiatement. Pour plus d'informations sur l'appartenance à un groupe ou à un rôle, voir «Modification des groupes auxquels l'utilisateur appartient», à la page 53 et «Modification des rôles auxquels un utilisateur ou un groupe appartient», à la page 60.

Suppression d'un utilisateur

Pour supprimer un utilisateur, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox sous le nom d'utilisateur *admin* ou sous le nom d'un utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** apparaît.
3. Cliquez sur le lien **Users**. La page **Users** apparaît.
4. Sélectionnez un utilisateur dans la liste.
5. Cliquez sur le bouton **Delete** pour supprimer l'utilisateur.

Important : Lorsque vous supprimez un utilisateur, celui-ci est définitivement supprimé de DB2 Alphablox.

Modification des groupes auxquels l'utilisateur appartient

Cette section indique la procédure à suivre pour ajouter ou modifier l'appartenance à un groupe d'un utilisateur ou d'un groupe.

Pour modifier les groupes auxquels appartient un utilisateur, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox sous le nom d'utilisateur *admin* ou sous le nom d'un utilisateur membre du groupe des administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** apparaît.
3. Cliquez sur le lien **Users**. La page **Users** apparaît.
4. Cliquez sur le bouton **Edit**. La page **Edit User** apparaît.
5. Cliquez sur l'onglet **Membership**.
6. Pour ajouter un groupe aux groupes dont l'utilisateur est membre, sélectionnez le groupe dans la liste **Available Groups** et cliquez sur le bouton en forme de flèche vers la gauche pour le déplacer vers la liste **Group Memberships**.
7. Pour retirer un groupe des groupes dont l'utilisateur est membre, sélectionnez le groupe dans la liste **Group Memberships** et cliquez sur le bouton en forme de flèche vers la droite pour le déplacer vers la liste **Available Groups**.
8. Pour associer un nouveau rôle à l'utilisateur, sélectionnez le rôle dans la liste **Available Roles** et cliquez sur le bouton en forme de flèche vers la gauche pour le déplacer vers la liste **Role Membership**.

Remarque : Les modifications apportées à un groupe ou un rôle d'un utilisateur sont appliquées immédiatement.

Chapitre 9. Définitions de groupes

Dans les pages d'administration des **groupes** vous pouvez créer, modifier et supprimer des groupes.

Des informations sur ces pages de **groupes** sont également disponibles en sélectionnant le lien **Aide** dans l'angle supérieur droit de chaque page.

Remarque : Avant de passer à une autre page, assurez-vous de bien enregistrer les modifications apportées aux groupes existants en cliquant sur le bouton **Enregistrer**. Sans cela, les modifications apportées à une page seront perdues.

Création d'un groupe

Le lien **Groupes** situé sous l'onglet **Administration** vous permet d'ouvrir des pages que vous pouvez utiliser pour créer ou modifier des groupes. Lors de l'installation, deux groupes (Administrateurs et Public) sont créés. Cliquez sur le bouton **Créer** dans la page **Groupes** pour définir d'autres groupes. Sélectionnez un nom de groupe existant et cliquez sur le bouton **Modifier** ou **Supprimer** pour le modifier ou le supprimer. Si un groupe est endommagé, vous pouvez le réparer en redémarrant DB2 Alphablox. Cette section indique les étapes à suivre pour définir un nouveau groupe à partir de la page d'accueil de DB2 Alphablox.

Effectuez les opérations suivantes pour définir un nouveau groupe :

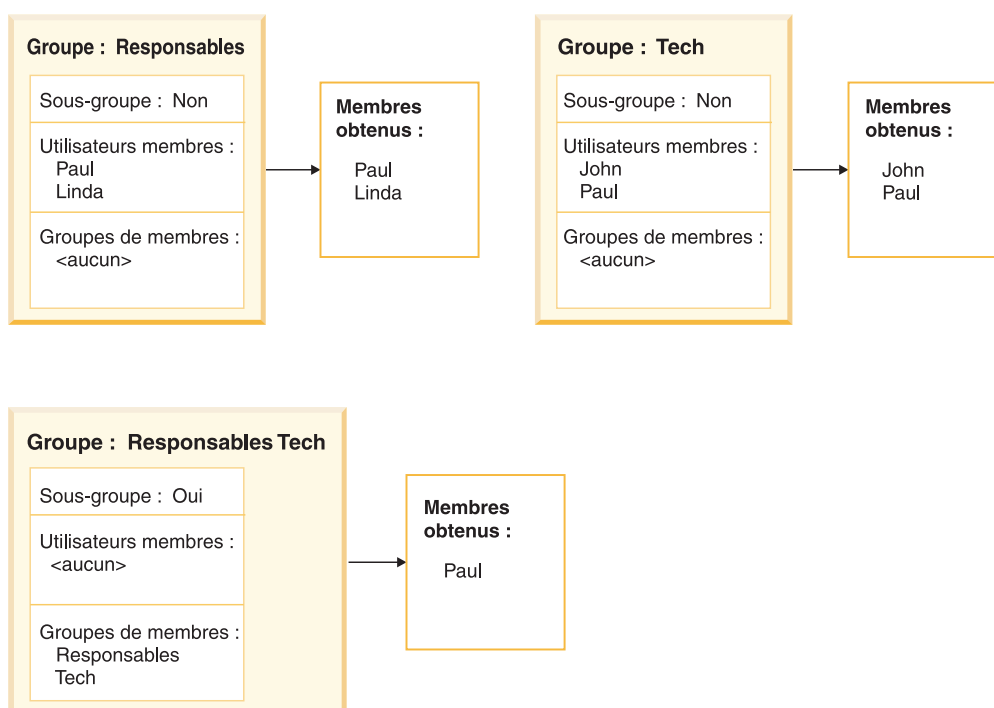
1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe administrateur.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Cliquez sur le lien **Groupes**. La page **Groupes** s'affiche.
4. Cliquez sur le bouton **Créer**. La page de **création de groupe** s'affiche.
5. Dans la zone de texte de **nom de groupe**, entrez le nom du groupe (requis). Vous pouvez utiliser les caractères A-Z, a-z, 0-9, le trait de soulignement ou des caractères spéciaux (par exemple, les caractères accentués) lorsque la langue d'exécution n'est pas l'anglais. L'affichage du nom de groupe respecte la casse, mais le nom de groupe qui est authentifié n'est pas sensible à la casse. Les noms *Public*, *Privé* et *Propriétés* sont réservés et ne peuvent pas être utilisés comme noms d'objets.
6. Eventuellement, entrez une **description** pour le groupe.
7. Dans la liste déroulante **Sous-groupe**, sélectionnez **Oui** si ce groupe doit agir comme sous-groupe ou **Non** si cela ne sera pas le cas. Pour plus d'informations sur les sous-groupes, voir «Présentation des sous-groupes», à la page 56.
8. Ajoutez des utilisateurs et/ou des groupes à ce groupe dans les zones de sélection d'**utilisateurs membres** et de **groupes membres** en faisant défiler les listes d'utilisateurs et de groupes. Maintenez la touche Ctrl enfoncée pour sélectionner plusieurs utilisateurs ou groupes que vous souhaitez ajouter à ce groupe.
9. Fournissez des valeurs pour les propriétés personnalisées éventuelles. (Si des propriétés de groupe personnalisées sont définies, elles apparaissent sous la zone de sélection des **groupes membres**.)

10. Cliquez sur le bouton **Enregistrer** pour définir le nouveau groupe et revenir à la page **Groupes**.

Présentation des sous-groupes

Un *sous-groupe* est un mécanisme par lequel les utilisateurs appartenant à plusieurs groupes peuvent devenir automatiquement des membres d'un autre groupe. Lorsque la liste déroulante **Sous-groupe** d'un groupe est définie sur **Oui**, les membres utilisateurs qui en résultent sont composés de l'intersection des utilisateurs dans chacun des groupes affectés comme membres au nouveau groupe.

La figure ci-après montre un exemple d'utilisation d'un sous-groupe. Dans cet exemple, le groupe *ResponsablesTech* utilise un sous-groupe pour inclure automatiquement les membres appartenant aux groupes *Responsables* et *Tech*.



En utilisant le sous-groupe pour le groupe *ResponsablesTech*, un administrateur peut créer un groupe pour inclure automatiquement l'ensemble des membres en commun dans les groupes des *Responsables* et des *Tech*. Dans cet exemple, *Paul* est le seul utilisateur appartenant à ces deux groupes. Si d'autres utilisateurs étaient ajoutés aux groupes *Responsables* et *Tech*, ils deviendraient automatiquement des membres du groupe *ResponsablesTech*.

Remarque : Lorsque vous utilisez des sous-groupes, vous pouvez continuer à ajouter manuellement des utilisateurs individuels au groupe en les sélectionnant dans la zone des **utilisateurs membres**.

Modification ou suppression d'un groupe

Cette section indique les étapes à suivre pour modifier les propriétés ou supprimer un utilisateur ou un groupe de la page d'accueil de DB2 Alphablox.

Modification d'un groupe

Effectuez les opérations suivantes pour modifier un groupe :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe administrateur.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Cliquez sur le lien **Groupes**. La page des **groupes** s'affiche.
4. Sélectionnez un groupe dans la liste des groupes.
5. Cliquez sur le bouton **Modifier**. La page de **modification du groupe** s'affiche.
6. Modifiez les valeurs si nécessaire.
7. Cliquez sur le bouton **Enregistrer** pour enregistrer vos modifications et revenir à la page **Utilisateurs**.
8. Pour modifier les propriétés de l'application de l'utilisateur, cliquez sur l'onglet **Propriétés de l'application**.
9. Apportez les modifications nécessaires.
10. Cliquez sur le bouton **Enregistrer**.
11. Pour modifier l'appartenance d'un utilisateur à un groupe ou un rôle, cliquez sur l'onglet des appartenances **Memberships**.
12. Apportez les modifications nécessaires. Notez que les modifications apportées à l'appartenance prennent effet immédiatement. Pour plus d'informations sur l'appartenance à un groupe ou rôle, voir «Modification des groupes auxquels l'utilisateur appartient», à la page 53 et «Modification des rôles auxquels un utilisateur ou un groupe appartient», à la page 60.
13. Cliquez sur le bouton **Enregistrer**.

Suppression d'un groupe

Effectuez les opérations suivantes pour supprimer un groupe :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe administrateur.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Cliquez sur le lien **Groupes**. La page des **groupes** s'affiche.
4. Sélectionnez un groupe dans la liste des groupes.
5. Cliquez sur le bouton **Supprimer** pour supprimer le groupe.

Important : La suppression d'un groupe supprime de façon permanente le groupe de DB2 Alphablox.

Chapitre 10. Définitions des rôles

Les pages d'administration **Roles** vous permettent de créer, de modifier et de supprimer des rôles. La page Roles apparaît uniquement si vous exécutez DB2 Alphablox dans une configuration Apache Tomcat ; si vous utilisez WebSphere ou WebLogic, le serveur d'applications fournit cette fonctionnalité. Les rôles utilisateur sont généralement indiqués dans l'outil de déploiement d'applications du serveur d'applications Web. Une fois les rôles définis dans cet outil (et ajoutés au fichier descripteur de déploiement), ils peuvent ensuite être utilisés dans DB2 Alphablox correctement.

Les informations contenues dans les pages **Roles** sont également disponibles en cliquant sur le lien **Help** dans le coin supérieur droit de chaque page.

Remarque : Pour modifier des rôles existants, veillez à cliquer sur le bouton **Save** avant de passer à une page différente. Sinon, les modifications apportées à la page ne seront pas enregistrées.

Définition de nouveaux rôles

DB2 Alphablox peut contrôler l'accès aux applications à travers les rôles. Un rôle contient une liste d'utilisateurs et/ou de groupes, ainsi que le droit d'accès associé à chaque entrée de la liste.

Pour créer un nouveau rôle, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Cliquez sur le lien **Roles**. La page **Roles** s'affiche.
4. Cliquez sur le bouton **Create**. La page **Create Role** s'affiche.
5. Dans la zone de texte **Role Name**, entrez le nom du rôle (requis). Vous pouvez utiliser les caractères A-Z, a-z, 0-9, le trait de soulignement ou des caractères spéciaux (par exemple, caractères accentués) lors du fonctionnement dans une autre langue que l'anglais. L'affichage du nom est sensible à la casse, mais le nom réel authentifié ne l'est pas. Les noms *Public*, *Privé* et *Propriétés* sont réservés et ne peuvent pas être utilisés pour un nom d'objet.
6. Eventuellement, entrez une **Description** pour le rôle.
7. Cliquez sur le bouton **Save** pour définir le nouveau rôle et retourner à la page **Roles**.

Modification et suppression de rôles existants

Pour modifier un rôle existant, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Cliquez sur le lien **Roles**. La page **Roles** s'affiche.
4. Sélectionnez un rôle dans la liste des rôles.
5. Cliquez sur le bouton **Edit**. La page **Edit Roles** s'affiche.

6. Si nécessaire, modifiez la description pour le rôle.
7. Cliquez sur le bouton **Save** avant de poursuivre, sinon les modifications ne seront pas enregistrées.
8. Pour modifier les groupes affectés au rôle, cliquez sur l'onglet **Member Groups**. Une page s'affiche, répertoriant les groupes actuellement membres du rôle et les groupes disponibles qui n'en sont pas membres. Utilisez les flèches de défilement droite et gauche pour déplacer les éléments sélectionnés de la liste **Available Groups** vers la liste **Member Groups** et inversement, de la liste **Member Groups** vers la liste **Available Groups**.
9. Cliquez sur le bouton **Save** pour enregistrer les propriétés d'appartenance des groupes pour le rôle.
10. Pour modifier les utilisateurs membres du rôle, cliquez sur l'onglet **Member Users**. Une page s'affiche, répertoriant les utilisateurs actuellement membres du rôle et les utilisateurs disponibles qui n'en sont pas membres. Utilisez les flèches de défilement droite et gauche pour déplacer les éléments sélectionnés de la liste **Available Users** vers la liste **Member Users** et inversement, de la liste **Member Users** vers la liste **Available Users**.
11. Cliquez sur le bouton **Save** pour enregistrer les propriétés d'appartenance des utilisateurs pour le rôle.

Modification des rôles auxquels un utilisateur ou un groupe appartient

Pour modifier les rôles auxquels un utilisateur ou un groupe appartient, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Cliquez sur le lien **Users** ou **Groups**. La page **Roles** s'affiche.
4. Sélectionnez l'utilisateur (ou groupe) dans la liste d'utilisateurs (ou de groupes) de la page **Users** ou **Groups**.
5. Cliquez sur le bouton **Edit**. La page **Edit User** (ou **Edit Group**) s'ouvre.
6. Cliquez sur l'onglet **Memberships**.
7. La moitié inférieure du panneau **Memberships** présente une zone pour les **Role Memberships** et une zone pour les **Available Roles**. Utilisez les flèches de défilement droite et gauche pour déplacer les éléments sélectionnés de la liste **Available Roles** vers la liste **Role Memberships** et inversement, de la liste **Role Memberships** vers la liste **Available Roles**.

Suppression d'un rôle existant

Pour supprimer un rôle existant, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Cliquez sur le lien **Roles**. La page **Roles** s'affiche.
4. Sélectionnez un rôle dans la liste des rôles.
5. Cliquez sur le bouton **Delete** pour supprimer le rôle.

Important : La suppression d'un rôle entraîne sa suppression permanente de DB2 Alphablox.

Chapitre 11. Sécurité et authentification

Ces rubriques décrivent la sécurité et l'authentification lors de l'utilisation de DB2 Alphablox dans une configuration Apache Tomcat, ne faisant appel à aucun serveur Web externe ou utilisant un autre serveur Web (Sun iPlanet, Microsoft IIS ou Apache HTTP Server).

Pour plus d'informations sur la sécurité et l'authentification lors de l'exécution avec un serveur d'applications commercial, tel que WebSphere ou WebLogic, consultez la documentation de ces produits.

Remarque : DB2 Alphablox 8.4 et 8.4.1 prennent en charge des serveurs Web externes différents. DB2 Alphablox 8.4.1 ne prend pas en charge Sun iPlanet. DB2 Alphablox 8.4.1 prend en charge Apache HTTP Server 2.0 et DB2 Alphablox 8.4 prend en charge Apache HTTP Server 1.3. Pour plus de détails, consultez la configuration système requise dans le *Guide d'installation*.

Modes d'authentification et de sécurité de DB2 Alphablox

Pour que DB2 Alphablox fournisse une page à un utilisateur, celui-ci doit être authentifié. Cela est valable quel que soit le serveur Web sous-jacent. Toutefois, lorsque vous utilisez un serveur Web externe, l'authentification sur DB2 Alphablox est généralement transparente.

Par défaut, les pages HTML ne sont pas routées vers le serveur d'applications dans les configurations de DB2 Alphablox derrière des serveurs Web et aucune demande d'authentification n'est faite tant qu'aucune page JSP ou page avec du contenu DB2 Alphablox n'est reçue par le serveur d'applications. Les serveurs Web peuvent être configurés pour authentifier les pages HTML, mais une configuration DB2 Alphablox ne prend pas en charge cette authentification par défaut. Si vous voulez vous authentifier lorsque DB2 Alphablox est configuré derrière un serveur Web, configurez votre serveur Web pour prendre en charge l'authentification de toutes les pages HTML ou vérifiez que votre page de bienvenue est un fichier JSP.

DB2 Alphablox prend en charge deux types d'authentification d'utilisateur :

Type d'authentification	Description
Authentification du serveur d'applications	Lorsque DB2 Alphablox fonctionne dans un serveur d'applications sans serveur Web externe, l'authentification et l'autorisation sont gérées par le serveur d'applications.
Authentification du serveur Web	Lorsque vous utilisez Sun iPlanet, Apache HTTP Server ou Microsoft IIS en tant que serveur Web, celui-ci effectue l'authentification. Lorsque DB2 Alphablox essaie d'authentifier un utilisateur, le serveur Web transmet un ID utilisateur à DB2 Alphablox. Remarque : Le serveur Web ne transmet pas l'ID utilisateur et le mot de passe à DB2 Alphablox car cela pourrait provoquer des problèmes de sécurité.

Modèle de sécurité dans Alphablox 8.4.1

DB2 Alphablox 8.4.1 fournit un nouveau modèle de sécurité, reposant sur l'API de sécurité Servlet 2.4 standard. Parce qu'Alphablox 8.4.1 et 8.4 prennent en charge des versions différentes d'Apache Tomcat, la prise en charge de la sécurité diffère aussi car les deux versions Tomcat prennent en charge différents modèles de sécurité et différentes spécifications de servlet. Dans DB2 Alphablox 8.4.1, les informations de sécurité sont maintenant configurées directement dans le fichier *conf/server.xml* d'Apache Tomcat. Pour configurer la sécurité dans Tomcat 5.5, consultez la documentation d'Apache Tomcat concernant la configuration des domaines (<http://tomcat.apache.org/tomcat-5.5-doc/realms-howto.html>).

Les API Alphablox qui prennent en charge ce modèle de sécurité sont fournies dans trois packages :

- `com.alphablox.security.jaas`
- `com.alphablox.security.jndi`
- `com.alphablox.security.ntlm`

Module de connexion Alphablox JAAS (Java Authentication and Authorization Service)

Le package `com.alphablox.security.jaas` fournit un module de connexion Alphablox JAAS. La classe `com.alphablox.security.jaas.AlphabloxLoginModule` authentifie les utilisateurs en accédant aux informations d'authentification à partir du référentiel DB2 Alphablox. Si l'option `debug` est définie sur `true` (valeur par défaut) dans le fichier de configuration de connexion *conf/alphablox-jaas.config* de votre répertoire d'installation Tomcat 5.5, les messages de débogage sont sortis vers le journal serveur Alphablox.

Configuration du domaine JNDI dans Tomcat 5.5

Le domaine JNDI Apache Tomcat ne prend pas en charge les groupes dynamiques LDAP, mais DB2 Alphablox propose le domaine JNDI Alphablox pouvant être utilisé pour accéder aux informations sur les utilisateurs et les groupes à partir des groupes dynamiques LDAP. La classe `com.alphablox.security.jndi.AlphabloxJNDIRealm` étend la prise en charge de base pour `JNDIRealm` dans Tomcat 5.5 en cherchant en plus l'appartenance aux rôles à travers les groupes dynamiques LDAP. Cette classe définit quatre attributs supplémentaires à ceux fournis dans `JNDIRealm`.

Attribut	Description
<code>userName</code>	Facultatif. Définit le nom d'attribut à lire comme le nom unique de l'utilisateur. Cette valeur doit correspondre à l'attribut utilisé dans les chaînes <code>userPattern</code> ou <code>userSearch</code> du domaine JNDI. La valeur par défaut est <code>uid</code> .
<code>memberURL</code>	Facultatif. Définit le nom d'attribut de l'URL de groupe dynamique. La valeur par défaut est <code>memberURL</code> .
<code>cacheEnabled</code>	Facultatif. Active ou désactive la mise en cache des groupes dynamiques vers un nom d'utilisateur associé. La valeur par défaut est <code>true</code> .
<code>dynamicGroupSearch</code>	Requis. Définit le filtre JNDI utilisé pour rechercher l'élément <code>roleBase</code> pour les groupes dynamiques LDAP. La valeur par défaut est <code>null</code> . L'activation du cache (<code>cacheEnabled="true"</code>) contribue à améliorer les performances car les groupes dynamiques n'ont pas besoin d'être sans cesse interrogés.

Consultez la documentation Javadoc du centre de documentation DB2 Alphablox pour des informations détaillées sur les API.

Ajout du domaine JNDI Alphablox à Apache Tomcat 5.5

Pour ajouter un domaine JNDI Alphablox à Tomcat, configurez le fichier *conf/server.xml* d'Apache Tomcat avec l'attribut `className` défini sur `com.alphablox.security.jndi.AlphabloxJNDIRealm` :

```
<Domaine
  className="com.alphablox.security.jndi.AlphabloxJNDIRealm"
  connectionURL="ldap://your_ldap.your_server.com:port"
  userBase="..."
  userSearch="..."
  userRoleName="..."
  roleBase="..."
  roleName="..."
  roleSearch="..."
  dynamicGroupSearch="(& (cn={0}) (objectclass=groupofur1s))"
/>
```

Remarque :

- Le caractère "&" doit être codé "&#amp;" pour le langage XML.
- Vous devez redémarrer Tomcat après avoir apporté les modifications au fichier *server.xml*.

Admin contre droits d'utilisateur

Lorsque DB2 Alphablox est installé, il crée un profil d'utilisateur unique avec des privilèges d'administration. Le nom pour ce profil est *admin* et ne peut pas être modifié. Le mot de passe par défaut pour *admin* est *password*. Lorsque vous utilisez la sécurité du serveur Web, l'utilisateur nommé *admin* doit exister sous la méthode d'authentification du serveur Web. (Lorsque vous utilisez IIS et Windows NTLM, *admin* peut être un utilisateur local plutôt qu'un utilisateur du domaine.)

L'utilisateur *admin* dispose d'un accès en lecture/écriture sur les objets DB2 Alphablox, dont les propriétés utilisateur, à travers les pages **Administration** de DB2 Alphablox. Les utilisateurs DB2 Alphablox disposent d'un accès en lecture/écriture uniquement sur leurs propres propriétés utilisateur. L'utilisateur *admin* ou un autre utilisateur DB2 Alphablox membre du groupes d'administrateurs peut accorder l'accès à une ou plusieurs applications à des utilisateurs spécifiques, à des groupes d'utilisateurs ou à tous les utilisateurs.

Suppression des droits de connexion de l'utilisateur invité pour les applications

Par défaut, DB2 Alphablox s'installe avec deux utilisateurs par défaut : Invité et Admin. Pour empêcher l'utilisateur Invité d'accéder à une application, vous pouvez modifier le fichier *web.xml* d'une application en remplaçant `AlphabloxUser` par `AlphabloxAuthenticatedUser` dans les éléments `role-name` pour les éléments `auth-constraint` et `security-role`.

Applications et domaines de sécurité du serveur d'applications

Fréquemment, les applications sont créées sur un serveur de développement, puis déplacées vers des serveurs de production lorsqu'elles sont déployées. Lorsqu'ils déplacent des applications entre deux serveurs différents qui ne partagent pas le même nom d'instance, les développeurs et administrateurs doivent savoir que le

domaine de sécurité indiqué dans le fichier `web.xml` de l'application migrée peut ne pas correspondre au domaine de sécurité du nouvel emplacement. Sur les serveurs d'applications IBM WebSphere et BEA WebLogic, lorsque des applications essaient d'accéder à d'autres applications sur le même serveur avec des domaines de sécurité différents, les utilisateurs sont invités à s'authentifier. Pour éviter cela, le fichier `web.xml` de l'application doit être modifié pour que les domaines de sécurité correspondent.

Une autre situation dans laquelle les utilisateurs peuvent être invités à s'authentifier survient lorsqu'une application reçue comme un fichier WAR ou EAR est importée dans les serveurs d'applications WebSphere et WebLogic. Le nom d'instance n'est pas vérifié et il pourrait s'agir d'une instance différente.

Authentification du serveur Web contre authentification DB2 Alphablox

DB2 Alphablox autorise uniquement les utilisateurs authentifiés à accéder aux applications. S'il n'est pas correctement authentifié, un utilisateur ne peut pas se connecter à une source de données, extraire des données, ni afficher les données sur les pages de l'application. Parce que DB2 Alphablox ne verrouille pas l'accès au niveau des répertoires, un utilisateur qui s'authentifie correctement peut utiliser le navigateur pour essayer d'ouvrir une application restreinte. Bien que l'utilisateur puisse ouvrir la page de l'application, les données n'apparaissent pas.

Pour plus d'informations sur la façon d'empêcher les utilisateurs à parcourir les répertoires et les fichiers du répertoire dans lequel résident les applications, voir «Désactiver l'exploration des répertoires», à la page 71.

Dans une installation par défaut, DB2 Alphablox effectue sa propre authentification. Si vous utilisez un autre serveur Web, toutefois, vous pouvez configurer DB2 Alphablox pour qu'il fasse faire l'authentification par le serveur Web. L'authentification par le serveur Web peut alléger la charge administrative de l'administrateur DB2 Alphablox car elle peut se faire par rapport aux comptes utilisateur pouvant déjà exister sur le réseau local. Cette section décrit comment configurer DB2 Alphablox pour que le serveur Web effectue l'authentification.

Remarque : Lorsque vous sélectionnez l'adaptateur `WebServerRealm`, la sécurité du serveur Web est prioritaire. Dans ces cas-là, le mot de passe de DB2 Alphablox n'a pas d'importance.

Utilisation des options de sécurité du serveur Web Sun iPlanet sur Alphablox 8.4 avec Apache Tomcat 3.2.4

Le modèle de sécurité Sun iPlanet fonctionne bien pour la plupart des combinaisons utilisateur/navigateur en mode texte clair ou basique. Pour administrer DB2 Alphablox via une URL à partir d'iPlanet, ajoutez l'utilisateur nommé `admin` à la base de données d'utilisateurs du serveur Web. Sinon, accédez à l'URL DB2 Alphablox directement (par exemple, utilisez `http://<nom_serveur>:<port>/AlphabloxAdmin/home`).

Pour utiliser la sécurité DB2 Alphablox avec iPlanet, désactivez la sécurité sur le serveur Web iPlanet. Tous les utilisateurs sont ensuite authentifiés par DB2 Alphablox.

En outre, veillez à définir les paramètres DB2 Alphablox appropriés pour garantir un environnement sûr et fonctionnel. Pour plus d'informations, voir «Configuration de DB2 Alphablox pour utiliser la sécurité reposant sur le serveur Web», à la page 70.

Définition des options de sécurité Microsoft pour IIS NTLM

Lorsque vous utilisez DB2 Alphablox avec un serveur Web Microsoft IIS, vous pouvez configurer l'authentification de sécurité pour qu'IIS effectue l'authentification lorsqu'un utilisateur se connecte à DB2 Alphablox (plutôt que celle-ci soit effectuée par DB2 Alphablox). Si IIS est configuré pour accepter des connexions Windows dans un domaine de confiance, alors ces utilisateurs de confiance peuvent accéder à DB2 Alphablox sans authentification supplémentaire ; c'est-à-dire qu'ils n'ont pas besoin d'entrer de nom d'utilisateur, ni de mot de passe lorsqu'ils accèdent à DB2 Alphablox tandis qu'ils sont connectés sur le domaine de confiance. Un utilisateur *de confiance* est un utilisateur Windows qui a été authentifié sur un domaine de confiance Windows, généralement à partir d'un grand serveur d'entreprise Windows.

Dans cette configuration, IIS gère l'ensemble de l'authentification des requêtes. Pour que DB2 Alphablox puisse gérer les requêtes, celles-ci doivent être authentifiées avec le nom d'utilisateur. Par conséquent, si un utilisateur Microsoft Internet Explorer modifie le paramètre de sécurité par défaut du navigateur (dans les paramètres avancés) pour que ce dernier ne demande pas de nom d'utilisateur, ni de mot de passe, DB2 Alphablox ne peut pas gérer les requêtes de ces navigateurs.

Pour configurer DB2 Alphablox pour qu'il fonctionne de façon sécurisée avec Microsoft Internet Information Services (IIS) utilisant NTLM, effectuez les tâches suivantes :

- «Installation de Microsoft IIS»
- «Installation de DB2 Alphablox et sélection de Microsoft IIS comme serveur Web»
- «Configuration des paramètres de sécurité dans Microsoft IIS», à la page 66
 - «Activation des paramètres de sécurité dans Microsoft IIS», à la page 66
 - «Restriction des droits de l'utilisateur anonyme dans Microsoft IIS», à la page 67
- «Création d'un utilisateur local Windows nommé admin pour NTLM», à la page 68
- «Configuration de la sécurité NTLM dans Tomcat 5.5 pour Alphablox 8.4.1», à la page 68
- «Connexion à DB2 Alphablox», à la page 69

En outre, veillez à définir les paramètres DB2 Alphablox appropriés pour garantir un environnement sûr et fonctionnel. Pour plus de détails, voir «Configuration de DB2 Alphablox pour utiliser la sécurité reposant sur le serveur Web», à la page 70.

Installation de Microsoft IIS

Vous devez installer Microsoft IIS *avant* d'installer DB2 Alphablox. Pour plus d'informations sur l'installation d'IIS, consultez votre documentation Microsoft.

Installation de DB2 Alphablox et sélection de Microsoft IIS comme serveur Web

Pour exécuter DB2 Alphablox en utilisant Microsoft IIS comme serveur Web, vous devez créer une nouvelle configuration Microsoft IIS en sélectionnant l'option

Créer une configuration Microsoft IIS sur la page Choix d'une configuration de serveur Web durant l'installation. L'installation installera les fichiers nécessaires sous le répertoire principal du serveur Web.

Pour utiliser DB2 Alphablox avec IIS, vous devez installer IIS avant d'installer DB2 Alphablox.

Configuration des paramètres de sécurité dans Microsoft IIS

Configurez les éléments suivants à l'aide de la console de gestion Microsoft de Microsoft IIS.

Activation des paramètres de sécurité dans Microsoft IIS : Pour activer les paramètres de sécurité appropriés, procédez comme suit :

1. Ouvrez la console de gestion de Microsoft IIS.
2. Dans le volet de gauche, cliquez avec le bouton droit sur le serveur Web approprié (généralement, **Site Web par défaut**) et sélectionnez **Propriétés**. La fenêtre Propriétés du site Web par défaut s'ouvre.
3. Cliquez sur l'onglet **Sécurité de répertoire**.
4. Appuyez sur le bouton **Modifier**. Pour utiliser l'authentification de confiance Windows à travers NTLM, cochez la case **Authentification intégrée Windows** (vérifiez que les autres cases à cocher sont vides). Notez que seuls les utilisateurs Microsoft Internet Explorer pourront se connecter ainsi.

Cette méthode fonctionne uniquement si vous indiquez le nom d'hôte pour le client.

Remarque :

- Dans DB2 Alphablox 8.4, pour indiquer les noms d'hôtes autorisés, cliquez sur l'onglet **Administration** de la page d'accueil de DB2 Alphablox, cliquez sur le lien **System** sous la section **General Properties** et entrez les clients autorisés dans la zone de texte **Authorized Client List** de la page **System**. Pour limiter l'accès des clients à la machine sur laquelle DB2 Alphablox fonctionne, tapez localhost dans la zone de texte **Authorized Client List**.
- Dans DB2 Alphablox 8.4.1, indiquez les clients autorisés directement dans Tomcat 5.5. Tomcat 5.5 vous permet d'indiquer un filtre d'hôte distant et un filtre d'adresse distante pour identifier les adresses IP ou les noms d'hôtes à autoriser ou à refuser. Par exemple, pour bloquer toutes les requêtes entrantes, sauf celles de localhost :

```
<Valve  
  className="org.apache.catalina.valves.RemoteAddrValve" allow="127.0.0.1" />
```

Pour plus d'informations sur le filtre d'hôte distant et le filtre d'adresse distante, voir <http://tomcat.apache.org/tomcat-5.5-doc/config/valve.html>.

DB2 Alphablox invite l'utilisateur à fournir un nom d'utilisateur et un mot de passe, ensuite authentifiés par DB2 Alphablox.

Type de sécurité	Description
Autoriser l'accès anonyme	<p>Cette option permet aux utilisateurs de se connecter à l'aide du compte anonyme. Ne cochez pas cette option si vous utilisez la sécurité NTLM avec IIS. Une fois les utilisateurs authentifiés, ils peuvent parcourir les répertoires et les fichiers sous le répertoire webapps contenant les applications. Pour empêcher l'exploration de ce répertoire, voir «Désactiver l'exploration des répertoires», à la page 71.</p> <p>Si vous cochez cette case, vous accordez aux utilisateurs un accès illimité aux ressources. Pour plus de détails, voir «Restriction des droits de l'utilisateur anonyme dans Microsoft IIS», à la page 67.</p>
Authentification de base (texte clair)	<p>Cette option, pour les navigateurs Mozilla Firefox et Microsoft Internet Explorer, transmet les ID utilisateur et mots de passe en texte clair.</p> <p>Remarque : Cochez cette option dans les environnements dans lesquels les navigateurs Mozilla Firefox accèdent aux applications DB2 Alphablox sur un Microsoft IIS.</p> <p>Sélectionnez cette option et l'authentification intégrée Windows dans les environnements dans lesquels les navigateurs Mozilla Firefox et Internet Explorer accèdent aux applications DB2 Alphablox sur Microsoft IIS.</p>
Authentification intégrée Windows	<p>Il s'agit de la méthode recommandée pour l'utilisation de la sécurité IIS. Cette option, uniquement destinée aux environnements utilisant une combinaison de serveurs Web IIS et de navigateurs Microsoft Internet Explorer, entraîne la transmission des ID utilisateur et mots de passe dans un format Microsoft propriétaire, pas en texte clair.</p>

- Si vous avez coché la case **Autoriser l'accès anonyme** (et que vous n'utilisez donc pas la sécurité NTLM de confiance), cliquez sur le bouton **Modifier** à droite de la case à cocher **Autoriser l'accès anonyme**. Le nom d'utilisateur anonyme (généralement au format IUSR_<HOSTNAME>) s'affiche. Notez le nom d'utilisateur anonyme et ignorez les instructions de «Restriction des droits de l'utilisateur anonyme dans Microsoft IIS», à la page 67.
- Cliquez sur le bouton **OK** pour fermer la fenêtre Méthodes d'authentification.
- Cliquez sur le bouton **OK** pour fermer la fenêtre Propriétés du site Web par défaut.

Restriction des droits de l'utilisateur anonyme dans Microsoft IIS : Cette procédure est uniquement nécessaire si vous *n'utilisez pas* la sécurité NTLM de confiance. Seuls les administrateurs avancés Microsoft IIS et Windows doivent effectuer ces étapes. La restriction de l'accès de l'utilisateur anonyme au seul répertoire webapps ne restreint pas forcément l'accès aux objets sur le serveur Web. Par conséquent, cette méthode est recommandée.

Si l'option Autoriser l'accès anonyme est définie sous IIS, procédez comme suit pour restreindre les droits de l'utilisateur anonyme :

- Ouvrez l'explorateur de fichiers Windows.
- Recherchez le répertoire *Inetpub* et cliquez dessus avec le bouton droit pour ouvrir son menu. Le répertoire *Inetpub* est celui qu'IIS utilise pour stocker les documents auxquels le serveur Web donne accès et son répertoire par défaut est *C:\Inetpub*.
- Choisissez **Propriétés** dans la liste déroulante pour ouvrir la fenêtre **Propriétés du répertoire**.

4. Cliquez sur l'onglet **Sécurité**.
5. Appuyez sur le bouton **Ajouter** en bas de la fenêtre pour ouvrir la fenêtre **Ajouter des utilisateurs et des groupes**.
6. Cliquez sur le bouton **Afficher les utilisateurs** pour afficher la liste des utilisateurs.
7. Sélectionnez le nom de l'utilisateur anonyme (par exemple, *IUSR_nom_hôte*).
8. Appuyez sur le bouton **Ajouter** pour ajouter l'utilisateur anonyme.
9. Dans la liste **Type d'accès**, sélectionnez **Aucun accès**.
10. Appuyez sur le bouton **OK** pour revenir à la fenêtre d'autorisations du répertoire et cochez la case **Remplacer les autorisations sur les sous-répertoires**.
11. Appuyez sur le bouton **OK**. Windows met à jour toute l'arborescence de répertoires sous le répertoire *InetPub*, ce qui peut prendre quelques minutes.

Création d'un utilisateur local Windows nommé admin pour NTLM

Si vous avez ajouté un compte administratif Microsoft IIS durant l'installation de DB2 Alphablox, vous pouvez ignorer cette étape et utiliser cet utilisateur administratif au lieu de créer un compte admin sur votre machine locale.

La sécurité NTLM de confiance requiert la création d'un compte Windows pour l'utilisateur *admin* DB2 Alphablox. Par défaut, lors de l'authentification avec NTLM sur Microsoft IIS, le nom d'utilisateur correspond au nom d'utilisateur du domaine du serveur Web (généralement un domaine réseau).

Dans la plupart des cas, l'utilisateur *admin* DB2 Alphablox n'est pas un utilisateur du domaine réseau. Au lieu de cela, ajoutez l'utilisateur *admin* au domaine local de l'hôte IIS. Lors de l'authentification en tant qu'utilisateur *admin* sous IIS, le nom d'utilisateur prend le format suivant :

nom-machine\admin

Configuration de la sécurité NTLM dans Tomcat 5.5 pour Alphablox 8.4.1

1. Ouvrez le fichier *conf/server.xml* de votre répertoire Tomcat 5.5
2. Recherchez la définition du connecteur AJP/13, qui ressemble à l'exemple ci-dessous :

```
<Connector port="8009"
  enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
```
3. Ajoutez l'attribut *tomcatAuthentication* à la balise Connector et définissez-le sur *false* :

```
<Connector port="8009" tomcatAuthentication="false"
  enableLookups="false" redirectPort="8443" protocol="AJP/1.3" />
```
4. Remplacez votre balise Realm existante en la mettant en commentaire et en ajoutant une nouvelle balise Realm pour utiliser *AlphabloxNTLMRealm* de DB2 Alphablox :

```
<!--
  <Realm className="org.apache.catalina.realm.JAASRealm" debug="10"
    appName="Alphablox"
    userClassNames="com.alphablox.security.jaas.UserPrincipal"
    roleClassNames="com.alphablox.security.jaas.RolePrincipal"
    useContextClassLoader="false" />
-->
  <Realm className="com.alphablox.security.AlphabloxNTLMRealm" />
```

Connexion à DB2 Alphablox

Lorsque vous vous connectez à DB2 Alphablox en tant qu'utilisateur *admin* à travers Microsoft IIS, vous devez :

- Utiliser le format correct pour le nom de connexion.
Si la machine IIS est connectée à un domaine différent du client à partir duquel l'utilisateur *admin* s'authentifie, vous devez passer le nom du domaine client avec le nom d'utilisateur. Par exemple, si l'utilisateur *admin* se connecte à partir d'un poste client au domaine WEBDEV1, le format pour la connexion devient :
WEBDEV1\admin
- Vérifiez que l'authentification de navigateur est définie correctement.

Par défaut, le navigateur Web Microsoft Internet Explorer authentifie les utilisateurs sur NTLM avec leur nom d'utilisateur et leur mot de passe Windows. La connexion à Microsoft IIS en tant qu'autre utilisateur (par exemple, *admin*) requiert la modification de ce comportement par défaut. (Parce que le nom d'utilisateur et le mot de passe *admin* ne correspondront pas au nom d'utilisateur et au mot de passe, la tentative de connexion échouera.)

Effectuez les étapes suivantes pour forcer les navigateurs Microsoft Internet Explorer à demander un nom d'utilisateur et un mot de passe durant l'authentification :

1. Dans Internet Explorer, dans le menu **Outils**, sélectionnez **Options Internet**.
2. Sélectionnez l'onglet **Sécurité**.
3. Si l'application est utilisée via Intranet (LAN), sélectionnez **Intranet local**.
4. Appuyez sur le bouton **Personnaliser le niveau** pour définir le niveau de sécurité.
5. Dans la section **Authentification utilisateur**, section **Connexion**, sélectionnez le bouton **Demander le nom d'utilisateur et le mot de passe**.
6. Cliquez sur le bouton **OK** pour fermer la fenêtre Paramètres de sécurité.
7. Cliquez sur le bouton **OK** pour fermer la fenêtre Options Internet.

L'authentification intégrée Windows signifie qu'un navigateur recevra un en-tête Negotiate du serveur, indiquant d'utiliser l'authentification Kerberos ou l'authentification intégrée Windows (anciennement NTLM ou Stimulation/Réponse NT). Pour ignorer le processus Negotiate, procédez comme suit :

1. Arrêtez DB2 Alphablox et ouvrez une invite de commande, puis naviguez jusqu'au répertoire Inetpub\AdminScripts.
2. Tapez la commande suivante :
cscript adsutil.vbs set w3svc/NTAuthenticationProviders "NTLM"
3. Pour vérifier l'authentification, tapez :
cscript adsutil.vbs get w3svc/NTAuthenticationProviders

La valeur par défaut est Negotiate, NTLM. Après avoir défini NTLM uniquement, la valeur doit uniquement indiquer "NTLM".

4. Tapez la commande suivante : `iisreset Nom_ordinateur /RESTART` Elle redémarrera IIS.
5. Redémarrez DB2 Alphablox et vous devriez pouvoir vous connecter à DB2 Alphablox à travers IIS.

Configuration de DB2 Alphablox pour utiliser la sécurité reposant sur le serveur Web

Quel que soit le serveur Web utilisé, vérifiez les paramètres DB2 Alphablox suivants pour des questions de pratique et de sécurité :

- «Configuration de la génération automatique des comptes utilisateur»
- «Filtrer les adresses IP», à la page 71
- «Définir les droits sur les répertoires», à la page 71
- «Désactiver l'exploration des répertoires», à la page 71

Configuration de la génération automatique des comptes utilisateur

Pour que le serveur Web puisse répondre, même si l'utilisateur a été correctement authentifié sur Microsoft IIS, celui-ci doit disposer d'un ID utilisateur DB2 Alphablox pour effectuer une requête. Plutôt que d'avoir à créer des comptes utilisateur pour chaque utilisateur déjà connu d'IIS ou d'iPlanet, DB2 Alphablox peut les générer automatiquement.

Pour configurer DB2 Alphablox pour créer automatiquement des ID utilisateur, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page General s'ouvre.
3. Dans la section General Properties, cliquez sur le lien **System**.
4. Cochez la case **Automatically Generate User Profiles**.
5. Appuyez sur le bouton **Save**.

Ensuite, lorsqu'un nouvel utilisateur essaie de se connecter, DB2 Alphablox l'invite à fournir un ID utilisateur et un mot de passe et crée automatiquement le compte de celui-ci. Si vous utilisez Microsoft IIS en tant que serveur Web et souhaitez utiliser l'authentification Windows pour DB2 Alphablox, vous devez effectuer la configuration supplémentaire décrite dans «Définition des options de sécurité Microsoft pour IIS NTLM», à la page 65.

Restriction : Si vous avez besoin d'utiliser un navigateur Mozilla Firefox pour administrer DB2 Alphablox lors de l'utilisation de la sécurité NTLM IIS, veillez à vous connecter avec un navigateur Mozilla Firefox la première fois. Le fait de vous connecter avec un navigateur Mozilla Firefox crée l'utilisateur avec un mot de passe en texte clair ; sinon, l'authentification échouera pour les navigateurs Mozilla Firefox.

Important : Le filtrage des adresses IP (décrit dans la section suivante) peut empêcher d'éventuels utilisateurs malveillants de contourner la sécurité du serveur Web.

Filtrer les adresses IP

Dans DB2 Alphablox 8.4 (pas 8.4.1), vous pouvez configurer DB2 Alphablox pour qu'il n'accepte que les requêtes provenant de clients spécifiques. Sous la sécurité du serveur Web, DB2 Alphablox ne doit accepter que les requêtes provenant de l'hôte du serveur Web (c'est-à-dire, le serveur Web lui-même ou les administrateurs de la machine hôte du serveur Web). Sur la page d'accueil de DB2 Alphablox, onglet **Administration**, page **General Properties**, lien **System**, vous pouvez utiliser la propriété **Authorized Client List** pour créer une liste des noms DNS et adresses IP acceptables. Si vous voulez que seules les connexions provenant de la machine locale soient acceptées, définissez **Authorized Client List** sur *localhost*.

Dans DB2 Alphablox 8.4.1, indiquez les clients autorisés directement dans Tomcat 5.5. Apache Tomcat 5.5 vous permet d'indiquer un filtre d'hôte distant et un filtre d'adresse distante pour identifier les adresses IP ou les noms d'hôtes à autoriser ou à refuser. Par exemple, pour bloquer toutes les requêtes entrantes, sauf celles du système hôte local :

```
<Valve className="org.apache.catalina.valves.RemoteAddrValve" allow="127.0.0.1" />
```

Pour plus d'informations sur le filtre d'hôte distant et le filtre d'adresse distante, voir <http://tomcat.apache.org/tomcat-5.5-doc/config/valve.html>.

Définir les droits sur les répertoires

Lorsque vous utilisez DB2 Alphablox avec Microsoft IIS, vous devez empêcher les utilisateurs d'accéder aux fichiers restreints sous le répertoire docroot du serveur Web ou le répertoire webapps de DB2 Alphablox. Pour plus de détails sur la définition des droits sur les répertoires dans IIS, consultez votre documentation IIS.

Désactiver l'exploration des répertoires

Les navigateurs Internet permettent aux utilisateurs d'accéder aux répertoires du serveur Web de façon aléatoire. Pour empêcher les utilisateurs de trouver et d'accéder aux fichiers par ce biais, à moins qu'ils n'en connaissent le chemin et le nom exacts, vous devez désactiver l'exploration des répertoires sur le serveur Web. Consultez la documentation de votre serveur Web.

Chapitre 12. Extension de DB2 Alphablox

Le présent chapitre décrit comment étendre la fonctionnalité intégrée de DB2 Alphablox en personnalisant les calculs personnalisés, le client DHTML et le gestionnaire Extensible User Manager.

Présentation générale

DB2 Alphablox comporte un grand nombre de fonctions prenant en charge les besoins de la plupart des applications analytiques. Les développeurs peuvent également étendre la plateforme afin d'utiliser les calculs personnalisés, le gestionnaire d'utilisateurs et le client DHTML. Les calculs personnalisés peuvent être requis lorsque les API de calcul standard ne sont pas suffisantes. Les extensions du gestionnaire d'utilisateurs permettent l'authentification, l'autorisation et la personnalisation. Le client DHTML peut également être personnalisé pour gérer vos besoins uniques. Vous trouverez ci-dessous des sections sur les extensions de calcul, du gestionnaire d'utilisateurs et du client DHTML. Des informations relatives à la configuration de DB2 Alphablox pour la gestion de vos personnalisations sont également disponibles.

Extensions de calcul

DB2 Alphablox comporte des API de calcul intégrées qui prennent en charge les besoins de calcul standard de la plupart des applications analytiques. Cependant, si vous estimez que la fonctionnalité incluse ne suffit pas à vos besoins, vous pouvez créer vos propres extensions de calcul personnalisées en procédant comme suit :

1. Rédigez votre extension de calcul afin d'étendre la classe `Function`.
 - a. Implémentez la méthode suivante :

```
public double getResult(double[] variables)
```

La méthode `getResult` renvoie le résultat du calcul défini par l'utilisateur dans la liste des numéros que sont les variables.
 - b. Vérifiez que le fichier appartient au package suivant :

```
com.alphablox.util.calculator
```
 - c. La première lettre du nom du fichier doit être en majuscules et les suivantes en minuscules.
2. Compilez votre fichier.
3. Ajoutez le fichier à un fichier JAR.
4. Placez le fichier JAR dans le répertoire suivant :

```
<alphabloxDirectory>\lib
```
5. Modifiez les fichiers de démarrage afin d'inclure ce fichier JAR dans le chemin de classes. Pour plus de détails sur le mode de définition de ce chemin de classes, voir «Définition du chemin de classes», à la page 75.

Des informations supplémentaires sur les calculs personnalisés sont disponibles à la section `DataBlox Reference` du document `Developer's Reference for the DHTML Client`.

Extensions du gestionnaire d'utilisateurs

Le gestionnaire d'utilisateurs DB2 Alphablox peut être étendu afin de prendre en charge les besoins d'authentification, d'autorisation et de personnalisation d'utilisateur personnalisés qui ne sont pas gérés par la fonctionnalité prête à l'emploi DB2 Alphablox. Pour plus d'informations sur les commandes serveur et les interfaces Java permettant d'étendre le gestionnaire d'utilisateurs DB2 Alphablox, voir «Extensible User Manager», à la page 99. Pour plus d'informations sur la définition d'un chemin de classes pour l'accès à vos classes Java, voir «Définition du chemin de classes», à la page 75.

Extensions de client DHTML

Le client DHTML DB2 Alphablox propose aux utilisateurs un ensemble étendu de fonctions. Pour un grand nombre de pages d'application, cette fonctionnalité peut être suffisante. Toutefois, grâce au modèle d'interface utilisateur Blox intégré au client DHTML, les développeurs peuvent personnaliser leurs pages, utiliser des composants JavaBeans personnalisés et créer des bibliothèques de balises JSP personnalisées pour améliorer les applications analytiques.

Pour plus de détails sur l'extension de la fonction de DB2 Alphablox à l'aide du modèle d'interface utilisateur de blox, voir les documents *Developer's Guide for the DHTML Client* et *Developer's Reference for the DHTML Client*.

Configuration de DB2 Alphablox pour la prise en charge des classes Java personnalisées

Comme décrit précédemment, des personnalisations du client DHTML, des fonctions de calcul et le gestionnaire d'utilisateurs peuvent être créés à l'aide de classes Java personnalisées. Lors de la création et de l'utilisation de vos classes personnalisées, vous devez prendre en compte les points suivants :

- Les classes personnalisées, généralement regroupées dans des fichiers JAR, peuvent être stockées à n'importe quel emplacement sur le serveur.

Important : Si vous stockez vos classes Java dans un répertoire d'installation DB2 Alphablox, vos fichiers peuvent être supprimés lors de mises à niveau. Par conséquent, Alphablox recommande de stocker vos classes dans un emplacement hors du répertoire DB2 Alphablox ou de les inclure dans votre répertoire d'application :

```
<alphabloxDirectory>/webapps/<applicationDirectory>
```

- Les modifications apportées au fichier de traitement par lots DB2 Alphablox (description à la section suivante, «Définition du chemin de classes», à la page 75) seront perdues lors des mises à niveau DB2 Alphablox.

Définition du chemin de classes

Pour que DB2 Alphablox utilise vos classes personnalisées, vous devez ajouter le répertoire classes au chemin de classes DB2 Alphablox indiqué dans le fichier de traitement par lots de démarrage. Suivez une des procédures ci-dessous en fonction de votre serveur d'applications.

Pour les implémentations Apache Tomcat :

1. Dans un éditeur de texte, ouvrez le fichier <db2alphablox_dir>/appserver/bin/aas.bat (plateformes Windows) ou le fichier <db2alphablox_dir>/appserver/bin/aas.sh (plateformes Linux et UNIX), où <db2alphablox_dir> correspond au répertoire dans lequel DB2 Alphablox est installé.
2. Recherchez la ligne qui définit le chemin de classes et ajoutez une ligne en utilisant la même syntaxe pour désigner l'emplacement des classes. Par exemple, si vos classes se trouvent dans le répertoire c:/myclasses/classes :

```
set CLASSPATH=%CLASSPATH%;%AAS_LIB%\xerces.jar
// Ajoutez le chemin de classes ci-dessous
set CLASSPATH=%CLASSPATH%; c:/myclasses/classes
```

Si vous exécutez DB2 Alphablox en tant que service Windows, définissez également le chemin de classes dans le fichier de paramètres de services en <db2alphablox_dir>\appserver\conf\wrapper.properties :

```
wrapper.class_path=$WRAPPER_AAS_HOME$\lib\aasserver.jar
// Ajoutez le chemin de classes ci-dessous
wrapper.class_path=c:/myclasses/classes
```
3. Enregistrez les modifications.
4. Redémarrez DB2 Alphablox pour que les modifications soient prises en compte.

Pour les serveurs d'applications WebLogic, procédez comme suit :

1. Dans un éditeur de texte, ouvrez le fichier <db2alphablox_dir>/bin/aassetup.bat (plateforme Windows) ou <db2alphablox_dir>/bin/aassetup.sh (plateformes Linux et UNIX)
2. Recherchez la ligne qui définit le chemin de classes et ajoutez une ligne en utilisant la même syntaxe pour désigner l'emplacement des classes. Par exemple, si vos classes se trouvent dans le répertoire c:/myclasses/classes :

```
set CLASSPATH=%CLASSPATH%;%AAS_CP%
// Ajoutez le chemin de classes ci-dessous
set CLASSPATH=%CLASSPATH%; c:/myclasses/classes
```

Si vous exécutez DB2 Alphablox en tant que service Windows, définissez également chemin de classes dans le fichier de paramètres de services en <db2alphablox_dir>\bin\aassetup_nt_service.bat
3. Redémarrez WebLogic. Si DB2 Alphablox s'exécute en tant que service Windows, exécutez à nouveau le fichier installSvc.cmd sous <BEA>/weblogic700/server/bin.

Si vous utilisez WebSphere, vous devez créer un fichier JAR qui contient vos classes (vous ne pouvez pas avoir de classes indépendantes), puis placez le fichier JAR sous <db2alphablox_dir>/AppServer/lib/ext. Redémarrez WebSphere pour que la modification soit prise en compte.

Chapitre 13. Configuration des propriétés de DB2 Alphablox

Le lien **General** sous l'onglet **Administration** de la page d'accueil de DB2 Alphablox fournit des liens vers un ensemble de pages Web permettant d'effectuer les tâches administratives courantes. Notez que toutes ces tâches, comme beaucoup d'autres, peuvent également être effectuées à l'aide des commandes de console, comme décrit dans le Chapitre 19, «Commandes de la console DB2 Alphablox», à la page 141. Pour accéder à DB2 Alphablox, entrez l'URL suivante dans un navigateur :

`http://<nom_serveur>/AlphabloxAdmin/home`

où <nom_serveur> représente le nom du serveur et le numéro de port sur lequel DB2 Alphablox s'exécute. Ce chapitre inclut les procédures permettant de définir les propriétés d'exécution de DB2 Alphablox et décrit le journal DB2 Alphablox.

Tâches d'administration de DB2 Alphablox

La section **General Properties** de la page **General** sous l'onglet **Administration** fournit des liens vers l'interface, permettant d'effectuer les tâches DB2 Alphablox suivantes :

- «Configuration des propriétés de démarrage»
- «Configuration des propriétés système», à la page 79
- «Indication du port Telnet», à la page 80
- «Configuration de DB2 Alphablox Cube Manager», à la page 81

Configuration des propriétés de démarrage

Le processus d'installation et de configuration initial a établi des valeurs pour les propriétés sur la page **Startup**. Cet ensemble de propriétés représente le minimum absolu requis pour le démarrage de DB2 Alphablox.

Remarque : Consultez et définissez les valeurs correctes pour les autres propriétés de DB2 Alphablox, comme cela est décrit dans «Configuration des propriétés système», à la page 79, avant de déployer les applications.

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe Administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **General Properties** de la page, cliquez sur le lien **Startup** pour consulter et modifier les propriétés de démarrage de DB2 Alphablox :

Sur la page **Startup**, vous pouvez fournir des valeurs pour les propriétés de démarrage suivantes :

Nom de la propriété	Valeur par défaut	Description
Nom de l'instance	AlphabloxAnalytics	Le nom pour cette instance de DB2 Alphablox. Ce nom est déterminé durant l'installation et ne peut pas être modifié à moins de réinstaller DB2 Alphablox. Remarque : Il ne s'agit pas du nom de domaine (DNS) de la machine DB2 Alphablox.

Nom de la propriété	Valeur par défaut	Description
Nom du fichier journal	Server.log	Le nom du fichier journal pour cette session de DB2 Alphablox. Il réside dans le répertoire <référentiel>/servers/<Nom_instance>/logs.
Nom du fichier de commande	abc.console	Le nom d'un fichier de commandes facultatif. DB2 Alphablox recherche un fichier par ce nom durant le démarrage et lit une série de commandes DB2 Alphablox à partir de celui-ci. Les commandes de ce fichier utilisent la même syntaxe que celles entrées à travers une console DB2 Alphablox.
Niveau des messages par défaut	INFO	Le niveau minimal (moins sévère) des messages à afficher (et à écrire vers le fichier journal). Les valeurs valides, dans l'ordre de sévérité croissante, sont : DEBUG, VERBOSE, INFO, SYSTEM, WARNING, ERROR et FATAL. Pour une description des niveaux des messages, voir «Niveau des messages», à la page 149.
Activer le journal DB2 Alphablox	Oui	Indique s'il faut permettre à DB2 Alphablox d'écrire des messages vers le journal.
Durée d'inactivité de DB2 Alphablox	15 minutes	Indique le nombre de minutes d'inactivité de DB2 Alphablox avant sa suspension. La suspension est un état dans lequel DB2 Alphablox utilise moins de ressources. Une requête relance automatiquement l'activité du serveur.
Niveau des messages par défaut de la console DB2 Alphablox	FATAL	Le niveau minimal (moins sévère) des messages à envoyer à System.out (pour les implémentations DB2 Alphablox Apache Tomcat, la sortie est affichée sur la console Tomcat). Les valeurs valides, dans l'ordre de sévérité croissante, sont : DEBUG, VERBOSE, INFO, SYSTEM, WARNING, ERROR et FATAL.
Charger tous les objets utilisateur	activée	Charger tous les objets utilisateur dans la mémoire. Par défaut, le gestionnaire d'utilisateurs stocke les objets utilisateur dans le référentiel DB2 Alphablox. Utilisée conjointement au paramètre "Conserver les utilisateurs chargés (en quelques minutes)" des propriétés système, permet de mieux gérer la mémoire et de démarrer plus vite le gestionnaire d'utilisateurs lorsque vous avez de nombreux utilisateurs.

4. Entrez les valeurs requises pour ces propriétés.
5. Cliquez sur le bouton **Save** pour appliquer les modifications.

Remarque : Vous devez arrêter et redémarrer DB2 Alphablox pour que ces modifications prennent effet.

Configuration des propriétés système

Après l'installation initiale et avant le déploiement des applications, consultez et modifiez les propriétés système pour DB2 Alphablox.

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **General Properties** de la page, cliquez sur le lien **System** pour consulter et/ou modifier les propriétés système de DB2 Alphablox :

Sur la page de propriétés **System**, vous pouvez fournir des valeurs pour les propriétés suivantes :

Nom de la propriété	Valeur par défaut	Description
Préfixe URL du serveur Web	<vide>	La valeur correcte dépend du serveur Web utilisé. Ne modifiez pas cette valeur après l'installation sans d'abord contacter le support clients DB2 Alphablox.
Nouveau niveau de message de journal de début	INFO	Etablit le niveau de message le moins sévère à écrire vers un nouveau fichier journal.
Nouveau niveau de message de journal de fin	FATAL	Etablit le niveau de message le plus sévère à écrire vers un nouveau fichier journal. Si vous donnez la même valeur ici que pour la propriété Nouveau niveau de message de journal de début (par exemple, ERROR), un journal contenant uniquement les messages du niveau indiqué est créé.
Activer l'authentification	Oui	Détermine si les utilisateurs doivent s'authentifier avant d'accéder aux ressources DB2 Alphablox et aux applications DB2 Alphablox. Lorsque la valeur est "Non", tous les utilisateurs se connectent en tant qu'invité sans avoir à fournir de mot de passe.
Générer automatiquement les profils utilisateur	Non	Détermine si le profil utilisateur est automatiquement créé lorsqu'un utilisateur se connecte à DB2 Alphablox. Lorsque cette propriété est définie sur Oui , toute requête utilisateur, quel que soit le mot de passe, est acceptée et l'utilisateur peut accéder à DB2 Alphablox.
Liste des clients autorisés	<vide>	Répertorie les utilisateurs autorisés à accéder à DB2 Alphablox. La valeur est une liste séparée par des virgules d'adresses IP et/ou de noms d'hôte ; les caractères génériques ne sont pas autorisés. Une liste vide indique que tous les utilisateurs sont autorisés à accéder à DB2 Alphablox. Remarque : Lors de l'utilisation de la sécurité du serveur Web, il est important de définir les clients autorisés pour que seule la machine du serveur Web puisse accéder à DB2 Alphablox.

Nom de la propriété	Valeur par défaut	Description
Taille de l'historique des messages	100	Définit le nombre de messages enregistrés dans la zone d'historique des messages à afficher dans la console. Lorsque la zone est remplie, DB2 Alphablox repart au début, en remplaçant les messages les plus anciens (les messages sont toujours enregistrés dans le fichier journal).
Enregistrer lors de la fermeture	Oui	Indique s'il faut enregistrer l'état actuel de DB2 Alphablox (et toutes ses propriétés) lorsque sa session se termine. Si les administrateurs ou assembleurs ont apporté des modifications au cours d'une session et n'ont pas réussi à les enregistrer, le fait de laisser cette valeur sur "Oui" permet de ne pas perdre les modifications en question.
Thème du client HTML par défaut	coleman	Détermine le thème par défaut à utiliser lors du rendu des pages de l'application au format HTML. (Pour plus d'informations, consultez le <i>Developer's Guide</i> .)
Serveur SMTP	indiqué durant l'installation	Définit le nom d'un serveur SMTP pour que les applications qui utilisent JavaMail pour envoyer du courrier puissent fonctionner correctement. A définir sur le nom d'un serveur de courrier SMTP valide.
Conserver les utilisateurs chargés (en quelques minutes)	-1	Définit le nombre de minutes de conservation des objets utilisateur en mémoire après la dernière instance d'une session utilisateur. La valeur par défaut, -1, conserve les utilisateurs chargés indéfiniment (tant que DB2 Alphablox fonctionne). Utilisé conjointement au paramètre "Charger tous les objets utilisateur" des propriétés de démarrage, ce paramètre permet de mieux gérer la mémoire et de démarrer plus vite User Manager lorsque vous avez de nombreux utilisateurs.

1. Entrez les valeurs requises pour ces propriétés.
2. Cliquez sur le bouton **Save** pour appliquer les modifications.

Remarque : Vous devez arrêter et redémarrer DB2 Alphablox pour que les modifications apportées aux propriétés prennent effet.

Indication du port Telnet

Pour indiquer des ports sur lesquels DB2 Alphablox écoute, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **General Properties** de la page, cliquez sur le lien **Telnet Console** pour afficher et modifier les ports DB2 Alphablox.

Sur la page **Telnet Console**, les valeurs des propriétés suivantes peuvent être modifiées :

Nom de la propriété	Valeur par défaut	Description
Port de la console Telnet	23	Le port sur lequel la version Telnet de la console DB2 Alphablox fonctionne. Si DB2 Alphablox est le périphérique Telnet par défaut, il n'est pas nécessaire d'entrer un numéro de port lors de l'accès aux consoles Telnet. Définissez cette valeur sur 0 (zéro) pour désactiver l'accès Telnet à la console DB2 Alphablox.
Nom d'utilisateur Telnet	admin	Le nom d'un utilisateur pouvant exécuter une session Telnet.
Mot de passe Telnet	password	Le mot de passe pour le nom d'utilisateur Telnet.
Confirmer le mot de passe Telnet		Lorsque vous entrez un mot de passe dans la zone de texte Mot de passe Telnet , vous devez le confirmer dans celle-ci.
Délai d'attente Telnet	15 minutes	Indique le nombre de minutes qui s'écoulent avant l'expiration d'une session Telnet sur DB2 Alphablox. Lorsqu'une session Telnet expire, la session est arrêtée.
Redémarrer le gestionnaire de console	non cochée	Lorsqu'elle est cochée, cette propriété redémarre le gestionnaire de console quand vous appuyez sur le bouton Enregistrer .

4. Entrez les numéros de ports requis.
5. Cliquez sur le bouton **Save** pour appliquer les modifications.

Remarque : Vous devez arrêter et redémarrer les gestionnaires de services correspondants pour que ces modifications prennent effet. Pour redémarrer le ou les gestionnaires de services, cochez la case sous le ou les ports que vous avez modifiés et cliquez sur le bouton **Enregistrer**.

Configuration de DB2 Alphablox Cube Manager

Le fait de cliquer sur le lien **DB2 Alphablox Cube Manager** ouvre une page qui permet aux administrateurs de limiter le nombre de cubes DB2 Alphablox pouvant être chargés dans le serveur. Chaque cube DB2 Alphablox peut éventuellement utiliser une grande quantité de ressources système ; par conséquent, les administrateurs peuvent vouloir limiter le nombre de cubes en fonction de la mémoire disponible sur la machine sur laquelle DB2 Alphablox fonctionne.

Pour limiter le nombre de cubes DB2 Alphablox sur le système, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **General Properties** de la page, cliquez sur le lien **DB2 Alphablox Cube Manager** pour afficher et/ou modifier les propriétés système de DB2 Alphablox :
4. Cochez la case **Maximum Cubes**.

5. Entrez le nombre maximal de cubes DB2 Alphablox dans la zone de texte **Maximum Cubes**.
6. Cliquez sur le bouton **Save** pour appliquer les modifications.

Pour des informations détaillées sur la création et la gestion de cubes DB2 Alphablox à l'aide du DB2 Alphablox Cube Server, consultez le document *DB2 Alphablox Cube Server - Guide d'administration*.

Définitions des propriétés personnalisées

Les propriétés personnalisées peuvent être utilisées pour associer les détails de connexion des utilisateurs aux profils utilisateur DB2 Alphablox. Pour plus de détails sur le profil utilisateur, voir Chapitre 8, «Définitions d'utilisateurs», à la page 51. DB2 Alphablox fournit un exemple de fichier de propriétés personnalisées pour vous aider.

Les valeurs valides pour la propriété, ainsi que sa valeur par défaut, sont prises de la définition de la propriété. La propriété utilisateur est disponible pour les développeurs par programme, par exemple, à l'aide de la méthode `getUserProperty()` `RepositoryBlox`.

Remarque : Lorsque vous définissez une propriété personnalisée dont la valeur requiert des espaces, utilisez ` ` au lieu de taper les espaces. Les espaces tapés ne sont pas reconnus et sont supprimés de la valeur de la propriété.

Définition d'une nouvelle propriété utilisateur

Pour définir une nouvelle propriété utilisateur, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **Custom Properties** de la page, cliquez sur le lien **User Definitions**. La page **User Definitions** s'affiche.
4. Cliquez sur le bouton **Create**. La page **Create User Custom Properties** s'affiche.
5. Fournissez un nom (requis) dans la zone de texte **Property Name** et une description dans la zone de texte **Display Label**. La description s'affiche dans la définition de l'application sur les pages s'affichant depuis le lien **Applications** sous l'onglet **Administration**.

Remarque : Si vous utilisez un gestionnaire d'utilisateurs LDAP ou un autre gestionnaire d'utilisateurs externe (tel que NTLM) disposant de son propre éditeur, vous devrez choisir entre **External Repository** et **Local Repository** sous **Property Name**. Cliquez sur le bouton d'option **External Repository** et choisissez dans la liste déroulante pour que DB2 Alphablox charge la propriété personnalisée.

6. Cliquez sur le bouton **Multiple Select** si vous voulez que la propriété accepte plusieurs valeurs.
7. Sélectionnez le niveau **User Access** dans la liste déroulante. (Des privilèges administratifs sont toujours requis pour définir, modifier ou supprimer les propriétés d'applications.)

- **Hidden** : La propriété *n'apparaît pas* sur les pages New User and Group, Change User and Group et Manage User Profile. Sa valeur peut être modifiée dans une application (par exemple, à travers JavaScript) ou à travers cette page uniquement.
 - **Visible** : La propriété et sa valeur par défaut s'affichent sur les pages New User and Group, Change User and Group et Manage User Profile, mais l'utilisateur ne peut pas modifier sa valeur.
 - **Edit** : La propriété et sa valeur par défaut s'affichent et peuvent être modifiées sur les pages New User and Group, Change User and Group et Manage User Profile. Si la propriété dispose de plusieurs valeurs valides, l'utilisateur sélectionne une valeur dans une liste. Si la propriété ne dispose pas d'une valeur valide, l'utilisateur peut en entrer une.
8. Entrez une **Valeur par défaut** pour la propriété, le cas échéant.
 9. Dans la zone de texte **Value List**, entrez une chaîne facultative séparée par des virgules de valeurs de propriétés valides. Les valeurs entrées ici sont utilisées pour remplir la liste dans laquelle les utilisateurs font leur choix. Si aucune valeur n'est entrée et si l'option **User Access** est définie sur **Edit**, une zone de texte s'affiche sur les pages respectives dans lesquelles les utilisateurs peuvent entrer une valeur.
 10. Cliquez sur le bouton **Save** pour enregistrer la nouvelle propriété et revenir à la page **General**.

Modification d'une propriété utilisateur

Pour modifier une propriété utilisateur, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **Custom Properties** de la page, cliquez sur le lien **User Definitions**. La page **User Definitions** s'affiche.
4. Sélectionnez la définition à modifier et cliquez sur le bouton **Edit**. La page **Edit User Custom Property** s'affiche avec ses valeurs actuelles dans les zones d'entrée.
5. Apportez les modifications nécessaires.
6. Cliquez sur le bouton **Save** pour valider les modifications et revenir à la page **General**.

Suppression d'une propriété utilisateur

Pour supprimer une propriété utilisateur, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **Custom Properties** de la page, cliquez sur le lien **User Definitions**. La page **User Definitions** s'affiche.
4. Sélectionnez la définition de propriété à supprimer de la liste des propriétés existantes.
5. Cliquez sur le bouton **Delete** pour supprimer la propriété et revenir à la page **General**.

Remarque : La suppression d'une définition de propriété est permanente ; vous ne pouvez pas la récupérer.

Définition d'une nouvelle propriété d'application personnalisée

Pour définir une définition d'application personnalisée, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **Custom Properties** de la page, cliquez sur le lien **Application Definitions**. La page **Application Definitions** s'affiche.
4. Cliquez sur le bouton **Create**. La page **Create Application Custom Property** s'affiche.
5. Fournissez un nom (requis) dans la zone de texte **Property Name** et une description dans la zone de texte **Display Label**. La description s'affiche dans la définition de l'application sur les pages s'affichant depuis le lien **Applications** sous l'onglet **Administration**.
6. Cliquez sur le bouton **Multiple Select** si vous voulez que la propriété accepte plusieurs valeurs.
7. Sélectionnez le niveau **User Access** dans la liste déroulante. (Des privilèges administratifs sont toujours requis pour définir, modifier ou supprimer les propriétés d'applications.)
 - **Hidden** : La propriété *n'apparaît pas* sur ses pages respectives. Sa valeur peut être modifiée dans une application (par exemple, à travers JavaScript) ou à travers cette page uniquement.
 - **Visible** : La propriété et sa valeur par défaut s'affichent sur leurs pages respectives, mais l'utilisateur ne peut pas modifier la valeur.
 - **Edit** : La propriété et sa valeur par défaut s'affichent et peuvent être modifiées sur leurs pages respectives. Si la propriété dispose de plusieurs valeurs valides, les utilisateurs sélectionnent la valeur souhaitée dans une liste. Si la propriété ne dispose pas d'une liste de valeurs valides, les utilisateurs peuvent entrer une valeur.
8. Entrez une **Valeur par défaut** pour la propriété, le cas échéant.
9. Dans la zone de texte **Value List**, entrez une chaîne facultative séparée par des virgules de valeurs de propriétés valides. Les valeurs entrées ici sont utilisées pour remplir la liste dans laquelle les utilisateurs font leur choix. Si aucune valeur n'est entrée et si l'option **User Access** est définie sur **Edit**, une zone de texte s'affiche sur les pages respectives dans lesquelles les utilisateurs peuvent entrer une valeur.
10. Cliquez sur le bouton **Save** pour enregistrer la nouvelle propriété et revenir à la page **General**.

Modification d'une propriété d'application

Pour modifier une définition de propriété d'application, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **Custom Properties** de la page, cliquez sur le lien **Application Definitions**. La page **Application Definitions** s'affiche.
4. Sélectionnez la définition à modifier et cliquez sur le bouton **Edit**. La page **Edit Application Custom Property** s'affiche avec ses valeurs actuelles dans les zones d'entrée.
5. Apportez les modifications nécessaires.

6. Cliquez sur le bouton **Save** pour valider les modifications et revenir à la page **General**.

Suppression d'une propriété d'application

Pour supprimer une définition de propriété d'application, procédez comme suit :

1. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe d'administrateurs.
2. Cliquez sur l'onglet **Administration**. La page **General** s'affiche.
3. Dans la section **Custom Properties** de la page, cliquez sur le lien **Application Definitions**. La page **Application Definitions** s'affiche.
4. Sélectionnez la définition de propriété à supprimer de la liste des propriétés existantes.
5. Cliquez sur le bouton **Delete** pour supprimer la propriété et revenir à la page **General**.

Remarque : La suppression d'une définition de propriété est permanente ; vous ne pouvez pas la récupérer.

Remarque : Dans ce cas, tout ce qui se trouve sous la liste déroulante **Allow User To Edit Profile** apparaît en raison du contenu du fichier *UserPropDesc.properties*.

Création et gestion de collections de commentaires

DB2 Alphablox peut être utilisé pour gérer les commentaires des utilisateurs et les rendre disponibles sur des cellules de données dans des grilles multidimensionnelles. Les sections suivantes expliquent comment utiliser la page **Comments Management**, disponible sous l'onglet **Administration** DB2 Alphablox pour configurer une collection de commentaires à utiliser avec une source de données particulière. Les commentaires sont consultables dans le client DHTML sur les navigateurs Web pris en charge.

Remarque : Les collections de commentaires requièrent des étapes de configuration supplémentaires pour fonctionner avec le pilote JDBC Microsoft pour SQL Server, qui est une option incluse dans WebSphere. Pour obtenir de l'aide, contactez le support clients de DB2 Alphablox.

Accès à la boîte de dialogue **Comments Management**

Pour accéder à la page **Comments Management**, qui s'ouvre dans une nouvelle fenêtre, procédez comme suit :

1. Ouvrez la page d'accueil de DB2 Alphablox.
2. Cliquez sur l'onglet **Administration**.
3. Sous **Runtime Management** dans le menu de navigation sur la gauche, cliquez sur le lien **Comments Management**.

4. La boîte de dialogue Comments Management doit s'afficher dans une nouvelle fenêtre de navigateur.

Remarque : Pour utiliser la boîte de dialogue Comments Management, vous devez disposer des droits de création et de suppression des tables relationnelles. Pour utiliser l'API CommentsBlox lors du développement d'applications de commentaires personnalisées, vous pouvez avoir besoin des droits de sélection, insertion, mise à jour, suppression, création et élimination des tables.

Définition et accès à une source de données

Pour définir une source de données pour qu'elle soit disponible pour stocker vos collections de commentaires, procédez comme suit :

1. Sous l'en-tête Datasource, cliquez sur la liste de sélection Name et sélectionnez la source de données DB2 Alphablox définie dans laquelle vous voulez que vos collections de commentaires soient stockées. [Remarque : Pour en savoir plus sur la définition des sources de données dans DB2 Alphablox, voir Définitions des sources de données]
2. Dans les zones d'entrée **Username** et **Password**, entrez un identifiant de connexion et un mot de passe pour un utilisateur autorisé pour créer les tables dans la source de données. [Remarque : Si ces zones sont vides, le nom d'utilisateur et le mot de passe de source de données par défaut, si définis dans la définition de source de données DB2 Alphablox, seront automatiquement soumis après connexion.]
3. Appuyez sur le bouton Connect. Si la source de données définie et les informations utilisateur sont valides, une connexion à votre source de données doit s'établir et la section Collection de la boîte de dialogue Comments Management sera activée.

Définition des collections de commentaires

Pour que les utilisateurs puissent commencer à ajouter des commentaires aux cellules de données d'une grille, vous devez d'abord créer une collection de commentaires pour les y stocker.

Remarque : Les dimensions de votre collection de commentaires peuvent être ajoutées ou supprimées, tant qu'aucun commentaire ne s'y trouve.

1. Pour créer une nouvelle collection de commentaires, procédez comme suit :
2. Cliquez sur le bouton Create sous la zone de texte Collection. A droite de la fenêtre de navigateur, trois nouvelles sections s'affichent.
3. Dans la section Name, entrez le nom de votre collection de commentaires. [Remarque : Les noms ne doivent pas inclure d'espace et doivent être conformes aux exigences d'attribution de noms pour les tables de la base de données utilisée pour stocker les commentaires.]
4. Dans la section Fields, vous pouvez ajouter ou supprimer des zones pour votre collection de commentaires. Par défaut, trois zones requises sont incluses : auteur, horodatage et texte du commentaire. Vous pouvez ajouter d'autres zones, en définissant leur nom et, si vous le souhaitez, une description pour chacune d'elles. Une zone CellValue peut également être ajoutée. Cette zone fera que la valeur de la cellule actuelle pour la cellule sélectionnée sera automatiquement ajoutée au commentaire associé à cette cellule.

5. Dans la section Dimensions, vous devez sélectionner une source de données multidimensionnelle dans la liste des sources de données DB2 Alphablox définies. Une liste de dimensions disponibles dans le cube sélectionné doit apparaître dans la zone de texte sous la source de données répertoriée.
6. Sélectionnez les dimensions auxquelles vous voulez que les utilisateurs puissent ajouter des commentaires. Pour sélectionner plusieurs dimensions, vous devez appuyer sur la touche Ctrl et cliquer sur chaque dimension supplémentaire.

Remarque : Les dimensions sélectionnées affectent la portée des commentaires ajoutés par vos utilisateurs, comme le montrent les deux exemples suivants :

- Exemple 1 : Si vous avez un cube à trois dimensions (année, produit et région) et que vous ne sélectionnez que deux dimensions (produit et année) dans la liste Dimensions, alors les commentaires ajoutés à un membre de la dimension Région (non sélectionnée dans la liste Dimensions) seront affichés dans tous les autres membres Région.
 - Exemple 2 : Si vous avez un cube à trois dimensions (année, produit et région) et que vous sélectionnez les trois dimensions dans la liste Dimensions, alors les commentaires s'afficheront uniquement dans la cellule à laquelle les commentaires ont été ajoutés.
7. Appuyez sur le bouton Save. A ce stade, le nom de la collection récemment définie doit s'afficher dans la liste Collection.

Collections de commentaires utilisant des bases de données Microsoft SQL Server ou Sybase

Si vous prévoyez d'utiliser des bases de données Microsoft SQL Server ou Sybase pour stocker les commentaires pouvant inclure des caractères non-ASCII, que l'on trouve généralement dans les langues étrangères, vous devez procéder comme suit :

1. Pour chaque collection de commentaires que vous définissez, vous devez modifier les types de données des colonnes des tables automatiquement générées de varchar en nvarchar.

Les cinq tables suivantes sont partagées par toutes les collections de commentaires définies à l'aide de la source de données :

- Collections : Nom_Collection
- Commentaires : Texte_Commentaire
- Contexte : Nom_Contexte
- Dimensions : Nom_Dim
- Définitions de zones : descr, Nom_Zone
- Valeurs de zones : valeur_zone

De même, pour chaque collection de commentaires, une table unique est créée :

- ABXMBRT_<Nom_collection>

où <Nom_collection> est le nom que vous avez indiqué pour la collection de commentaires. Vous devez modifier le type de données en nvarchar pour la colonne suivante :

- ABXMBRT_<Nom_collection> : Nom_Membre

2. Toutes les pages JSP de votre application doivent avoir le jeu de caractères Unicode indiqué dans une directive de page JSP sur chaque page, comme suit :

```
<%@ page contentType="text/html; charset=UTF-8" %>
```

Affichage des définitions des collections de commentaires

Pour afficher la définition d'une collection existante, procédez comme suit :

1. Lorsque vous êtes connecté à une source de données définie, une liste de collections de commentaires prédéfinie s'affiche dans la zone de texte Collection et les boutons Créer, Afficher et Supprimer sont activés. Si vous n'êtes pas connecté, suivez d'abord les étapes de la rubrique Définition et accès à une source de données pour vous connecter à une source de données.
2. Cliquez sur l'une des collections de commentaires répertoriées dans la zone de texte Collection, puis appuyez sur le bouton Afficher. La définition de la collection doit s'afficher dans la moitié droite de la fenêtre.

Suppression de collections de commentaires

Pour supprimer une collection de commentaires définie, procédez comme suit :

1. Connectez-vous à une source de données définie (voir Définition et accès à une source de données ci-dessus).
2. Dans la zone de texte Collection, cliquez sur le nom de la collection de commentaires que vous voulez supprimer.
3. Appuyez sur le bouton Supprimer. Le nom de la collection doit être supprimé de la liste.

Ajout et affichage des commentaires

Une fois qu'une collection de commentaires est définie, les utilisateurs peuvent ajouter des commentaires aux cellules de données des grilles qui le permettent. Lorsqu'un utilisateur clique avec le bouton droit sur une cellule de données permettant les commentaires, un menu s'affiche. Dans le sous-menu Commentaires, les utilisateurs peuvent choisir d'ajouter ou d'afficher des commentaires.

Pour plus d'informations sur CommentsBlox, voir la section CommentsBlox du document *Developer's Reference*.

Création d'un processeur PDF distant

Pour des raisons de performances, de gestion de la mémoire ou pour partager le traitement PDF pour plusieurs hôtes DB2 Alphablox, vous pouvez décider d'exécuter votre moteur PDF sur un serveur dédié distant.

Configuration d'un processeur PDF distant

Pour utiliser un processeur PDF distant plutôt que le serveur interne :

1. Créez un répertoire principal sur votre serveur distant nommé *AlphabloxPDFRemote*.
2. Dans ce répertoire principal, créez deux nouveaux répertoires : *lib* et *repository*.
3. Dans le répertoire *repository*, créez deux répertoires : *font* et *theme*.
4. A partir du répertoire *db2alphablox_dir/lib/* de votre installation DB2 Alphablox, copiez les fichiers JAR suivants dans le répertoire *lib* du serveur distant :
 - *bforeport.jar*
 - *pdfserver.jar*
 - *xalan.jar*
 - *xercesImpl.jar*
 - *xml-apis.jar*

- icu4j_3_4_1.jar
5. Copiez tous les fichiers du répertoire *db2alphablox_dir/repository/theme* alphablox dans le répertoire *theme (repository/theme)* du serveur distant.
 6. Copiez tous les fichiers du répertoire *db2alphablox_dir/repository/font* alphablox dans le répertoire *font (repository/font)* du serveur distant.
 7. A partir du répertoire *db2alphablox_dir/bin/* de votre installation DB2 Alphablox, copiez le fichier script StartPDFRemoteServer (*StartPDFReportServer.bat* pour les systèmes Windows ou *StartPDFReportServer.sh* pour les systèmes Linux et UNIX) vers le répertoire *AlphabloxPDFRemote* du serveur distant.
 8. Modifiez le fichier script StartPDFRemoteServer, comme suit :
 - Modifiez le paramètre *LIB* pour qu'il pointe vers le répertoire *lib* du serveur distant.
 - Vérifiez que le paramètre *JAVA* pointe vers un interpréteur JRE ou un kit JDK valide.
 - Modifiez le paramètre *LISTEN_PORT* si nécessaire.
 - Vérifiez que le script démarre dans le répertoire actuel (par défaut le fichier script est dans le répertoire supérieur).
 9. Exécutez le fichier de traitement par lots. Le processeur PDF distant doit maintenant être disponible.
 10. Pour l'utiliser avec vos applications DB2 Alphablox, suivez les étapes de la section suivante pour configurer DB2 Alphablox pour utiliser le serveur distant.

Configuration de l'administration des rapports PDF distants

Le traitement des rapports PDF est administré à l'aide des paramètres PDF Reports Runtime Management des pages d'administration de DB2 Alphablox. Pour configurer un serveur distant, procédez comme suit :

1. Ouvrez les pages d'administration de DB2 Alphablox, puis cliquez sur l'onglet Administration.
2. Sur le côté gauche de la page, sous Runtime Management, cliquez sur le lien PDF Reports (DHTML) pour ouvrir la boîte de dialogue PDF Reports.
3. Pour configurer un serveur PDF distant, apportez les modifications suivantes :
 - a. Cliquez sur le bouton Edit.
 - b. Remplacez la valeur du paramètre Server Use Internal Server par Use PDF Report Server.
 - c. Définissez le nom d'hôte et le port de l'hôte pour le serveur distant.
 - d. Choisissez une option de compression, si vous le souhaitez ou laissez la valeur par défaut No Compression. Les options reposent sur les paramètres de compression définis dans le kit JDK. En réalité, la différence de compression entre les paramètres bas et élevés est minime. Le paramètre bas compresse déjà beaucoup les fichiers, tandis que le paramètre élevé propose seulement une compression légèrement supérieure.
 - e. Enregistrez vos modifications. Si vous avez correctement configuré le serveur PDF distant, l'état (Status) doit indiquer Available.

Fichiers journaux DB2 Alphablox

DB2 Alphablox journalise divers événements et erreurs DB2 Alphablox vers un fichier journal. Chaque message écrit vers la console DB2 Alphablox est également écrit vers le fichier journal. Les fichiers journaux sont conservés jusqu'à leur suppression ou déplacement et aident à conserver un historique de l'activité de DB2 Alphablox. L'historique est utile pour l'audit administratif et pour le support clients de DB2 Alphablox.

Paramètres d'intervalle de remplacement des fichiers journaux

Par défaut, DB2 Alphablox crée un nouveau fichier journal chaque jour et à chaque fois que le serveur d'applications démarre. En tant qu'administrateur, vous pouvez choisir de modifier les paramètres de remplacement des fichiers journaux dans les pages d'administration de DB2 Alphablox ou en modifiant le fichier `server.properties`.

Pour définir l'intervalle dans les pages d'administration de DB2 Alphablox, ouvrez un navigateur sur l'onglet Administration des pages d'administration de DB2 Alphablox, puis cliquez sur System sous General Properties. Par défaut, l'option New Log Rollover Interval est définie sur un remplacement chaque jour. Vous pouvez modifier ce paramètre pour qu'il n'y ait jamais de remplacement ou sur des minutes, heures, jours, semaines ou des kilooctets spécifiques.

Pour définir l'intervalle de remplacement des fichiers journaux dans le fichier `server.properties`, ouvrez le fichier `server.properties`, dans le répertoire suivant :

```
<repository_dir>/servers/<instanceName>/
```

Trouvez le paramètre pour `LogRollOverInterval` et remplacez-le par le paramètre souhaité à l'aide des valeurs suivantes :

Définissez-le sur l'une des valeurs suivantes (par défaut, 1D) :

#M - Remplacement après nombre (#) de minutes

#H - Remplacement après nombre (#) d'heures

#D - Remplacement après nombre (#) de jours

#W - Remplacement après nombre (#) de semaines

#K - Remplacement après nombre (#) de K octets écrits

NONE - Aucun remplacement

Noms des fichiers journaux

Le fichier journal se trouve dans le répertoire suivant :

```
<repository_root>/servers/<instance_name>/logs
```

Le nom par défaut du fichier journal actif est `Server.log`. Le nom est configuré sur la page accessible en cliquant sur l'onglet **Administration**, section **General Properties**, lien **Startup**.

DB2 Alphablox crée un nouveau fichier journal à chaque fois qu'il démarre. Lorsque DB2 Alphablox démarre, le fichier journal précédent est renommé comme suit :

`Server<horodatage>.log`

où *<horodatage>* est une heure de la dernière entrée du journal. L'horodatage est au format suivant :

`JJMMAA_HHmnSS`

Par exemple, si DB2 Alphablox s'est fermé à 18h22mn35s le 3 juin 2000, lorsqu'il redémarrera l'ancien fichier journal sera renommé `Server030600_182235.log` et un nouveau fichier journal nommé `Server.log` sera créé pour stocker les entrées de journal actuelles. Notez que l'horodatage est sur l'horloge de 24 heures. Si le nom du fichier journal est remplacé sur les pages **Administration**, l'horodatage est ajouté au nouveau nom.

Remarque : Le fichier journal actif ne présente pas l'horodatage ajouté à son nom ; son nom est exactement celui indiqué sur les pages **Administration**.

Gestion des fichiers journaux

Les anciens fichiers journaux peuvent être conservés indéfiniment sans problème, mais vous pouvez établir un processus pour les archiver après un certain temps. L'inconvénient de la conservation des anciens fichiers journaux est qu'ils occupent de l'espace sur le disque. La quantité d'espace qu'ils occupent dépend de l'activité de DB2 Alphablox et du niveau des messages par défaut défini sur les pages **Administration**. Si le niveau des messages est défini sur **DEBUG** ou **VERBOSE**, le fichier journal peut devenir rapidement volumineux.

Comme les anciens fichiers journaux sont renommés à l'aide de l'horodatage dans leur nom de fichier, vous pouvez créer des utilitaires pour vous aider à les gérer. Par exemple, vous pouvez créer un script pour supprimer ou déplacer les fichiers journaux quand ils datent de plus de trois mois.

Chapitre 14. Gestionnaire d'utilisateurs et personnalisation (Alphablox 8.4.1)

Le gestionnaire de personnalisation gère des fonctions de personnalisation qui permettent d'adapter le contenu des applications. Dans DB2 Alphablox 8.4.1, la fonction de sécurité et la fonction de personnalisation qui sont fournies dans le gestionnaire d'utilisateurs sont deux modèles différents.

DB2 Alphablox 8.4.1 fournit un modèle de sécurité basé sur Servlet 2.4 et sur le standard Java Authentication and Authorization Service. Ce modèle de sécurité est abordé dans «Modèle de sécurité dans Alphablox 8.4.1», à la page 62.

Au niveau de la personnalisation, le nouveau gestionnaire de personnalisation fournit une interface qui permet de communiquer la création, le chargement ou la suppression d'un utilisateur ou d'un groupe à DB2 Alphablox. Le gestionnaire de personnalisation permet également de spécifier et de modifier des propriétés dans un utilisateur ou un groupe, de modifier des utilisateurs ou des groupes contenus dans un groupe et de créer votre propre utilisateur ou groupe et l'ajouter dans le référentiel Alphablox.

Gestionnaire de personnalisation

Le nouveau gestionnaire de personnalisation dans DB2 Alphablox 8.4.1 vous permet de lire des propriétés d'utilisateur et de groupe à partir d'un référentiel d'utilisateurs externe. L'interface du gestionnaire dans le module `com.alphablox.personalization` est avertie lorsqu'un objet utilisateur ou groupe est créé, chargé ou supprimé. Cet objet de gestionnaire de personnalisation vous permet également d'ajouter aux utilisateurs et groupes DB2 Alphablox des propriétés qui sont définies dans un référentiel d'utilisateurs.

Si vous utilisez un gestionnaire de personnalisation, ajoutez un fichier `config.xml` dans le répertoire `alphablox_dir/repository/servers/instance`. Au démarrage, DB2 Alphablox effectue les opérations suivantes :

1. Recherche le fichier `config.xml` dans le référentiel.
2. Crée une instance de l'objet de gestionnaire de personnalisation.
3. Définit les propriétés fournies.
4. Affecte l'objet de gestionnaire de personnalisation au service de gestionnaire d'utilisateurs.

Voici un exemple de la balise `<Server>` :

```
<Server>
  <Service name="User Manager">
    <PersonalizationManager
      className="myPackage.CustomPersonalizationManager"
    />
  </Service>
</Server>
```

Le gestionnaire de personnalisation présente les fonctions et les comportements suivants :

- Les propriétés d'utilisateur ou de groupe dans votre référentiel d'utilisateurs externe sont disponibles dans les pages d'administration de DB2 Alphablox uniquement une fois que vous avez défini les mêmes propriétés personnalisées dans les pages d'administration de DB2 Alphablox.
- Après avoir défini les mêmes propriétés comme propriétés personnalisées, DB2 Alphablox effectue uniquement des opérations de lecture. Toutes les modifications apportées dans les pages d'administration de DB2 Alphablox sont des modifications locales dans DB2 Alphablox, en mémoire uniquement ; ces modifications ne sont pas inscrites dans le référentiel d'utilisateurs externe. Ces valeurs seront remplacées au prochain chargement de l'objet utilisateur.
- Si vous devez modifier les informations de connexion ou d'autres attributs dans le fichier *config.xml*, vous devez redémarrer Apache Tomcat, qui à son tour redémarre DB2 Alphablox.

Accès aux propriétés utilisateur et groupe JNDI

Les informations fournies dans cette section s'appliquent à DB2 Alphablox 8.4.1.

Vous pouvez rendre les propriétés d'utilisateur et de groupe dans votre référentiel JNDI- disponibles dans DB2 Alphablox et accessibles via l'interface API RepositoryBlox. Pour rendre ces propriétés disponibles, vous devez les spécifier dans le fichier *alphablox_dir/repository/servers/instance/config.xml* :

1. Ajoutez un fichier *config.xml* dans votre répertoire *alphablox_dir/repository/servers/instance* :

```
<Server>
  <Service name="User Manager">
    <PersonalizationManager
      className="com.alphablox.personalization.jndi.JNDIPersonalizationManager"
      connectionURL="ldap://nommachine:port"
      userBase="ou=People,dc=société,dc=com"
      userSearch="(& (uid={0}) (objectclass=person))"
      userProperties="Mail, Phone, Fax"
      groupBase="ou=Groups,dc=société,dc=com"
      groupSearch="(& (cn={0}) (objectclass=groupofuniquenames))"
      groupProperties="Description"
    />
  </Service>
</Server>
```

Le tableau suivant répertorie les attributs qui sont disponibles pour accéder aux propriétés d'utilisateur et de groupe JNDI.

Attribut	Description
connectionURL	Obligatoire. L'URL de connexion JNDI. Par exemple : ldap://machine1.ibm.com:10456
contextFactory	Facultatif. La fabrique de contexte à utiliser lors de la création du contexte initial du répertoire. La fabrique par défaut est com.sun.jndi.ldap.LdapCtxFactory.
groupBase	Obligatoire. La base JNDI à utiliser lors de la recherche de groupes.
groupProperties	Obligatoire. Les propriétés d'utilisateur à affecter du référentiel JNDI à l'objet utilisateur DB2 Alphablox. Cette propriété remplace toutes les propriétés d'utilisateur personnalisées définies dans les pages d'administration de DB2 Alphablox.
groupSearch	Obligatoire. La chaîne de recherche JNDI à utiliser lors de la recherche d'un groupe. La recherche de groupe peut contenir la syntaxe de remplacement MessageFormat {0}, qui sera remplacée par le nom unique du groupe recherché.
groupSubtree	Facultatif. Détermine si les sous-arborescences de groupBase sont utilisées lors de la recherche d'un groupe.
password	Facultatif. Le mot de passe à utiliser lors de la connexion à l'URL de connexion.
userBase	Obligatoire. La base JNDI à utiliser lors de la recherche d'utilisateurs.
userName	Facultatif. Le nom d'utilisateur à utiliser lors de la connexion à l'URL de connexion.
userProperties	Obligatoire. Les propriétés d'utilisateur à affecter du référentiel JNDI à l'objet utilisateur Alphablox. Cette propriété remplace toutes les propriétés d'utilisateur personnalisées définies dans les pages d'administration de DB2 Alphablox.
userSearch	Obligatoire. Définit la chaîne de recherche JNDI à utiliser lors de la recherche d'un utilisateur. La recherche d'utilisateur peut contenir la syntaxe de remplacement MessageFormat {0}, qui est remplacée par le nom unique de l'utilisateur recherché.
userSubtree	Facultatif. Les sous-arborescences de userBase à utiliser lors de la recherche d'un utilisateur.

Pour accéder aux valeurs des propriétés d'utilisateur dans vos fichiers JSP à l'aide de l'interface RepositoryBlox, procédez comme suit :

1. Ajoutez les mêmes propriétés d'utilisateur personnalisées à l'aide des pages d'administration de DB2 Alphablox. Accédez à l'onglet **Administration** > page **Propriétés générales** > section sur les **propriétés client**. Reportez-vous à la section «Définitions des propriétés personnalisées», à la page 82 pour plus de détails sur les étapes suivre.

2. Dans vos fichiers JSP, accédez à la propriété personnalisée suivante et à sa valeur via l'interface API RepositoryBlox :

```
<%@ taglib uri="bloxtld" prefix="blox" %>
...
<blox:repository id="myRepository" />
...
<html>
<head>
    <blox:header />
</head>
<body>
Your region is: <%= myRepository.getUserProperty("region") %>
...

```

3. Redémarrez Apache Tomcat, qui à son tour redémarre DB2 Alphablox, afin que les modifications apportées au fichier *config.xml* prennent effet.

Chapitre 15. Gestionnaire d'utilisateurs (Alphablox 8.4)

Ce chapitre explique comment configurer et utiliser la fonctionnalité de gestion des utilisateurs de DB2 Alphablox pour l'intégration LDAP et comment développer le moteur de personnalisation Extensible User Manager afin d'implémenter une sécurité personnalisée.

Présentation du gestionnaire d'utilisateurs de DB2 Alphablox

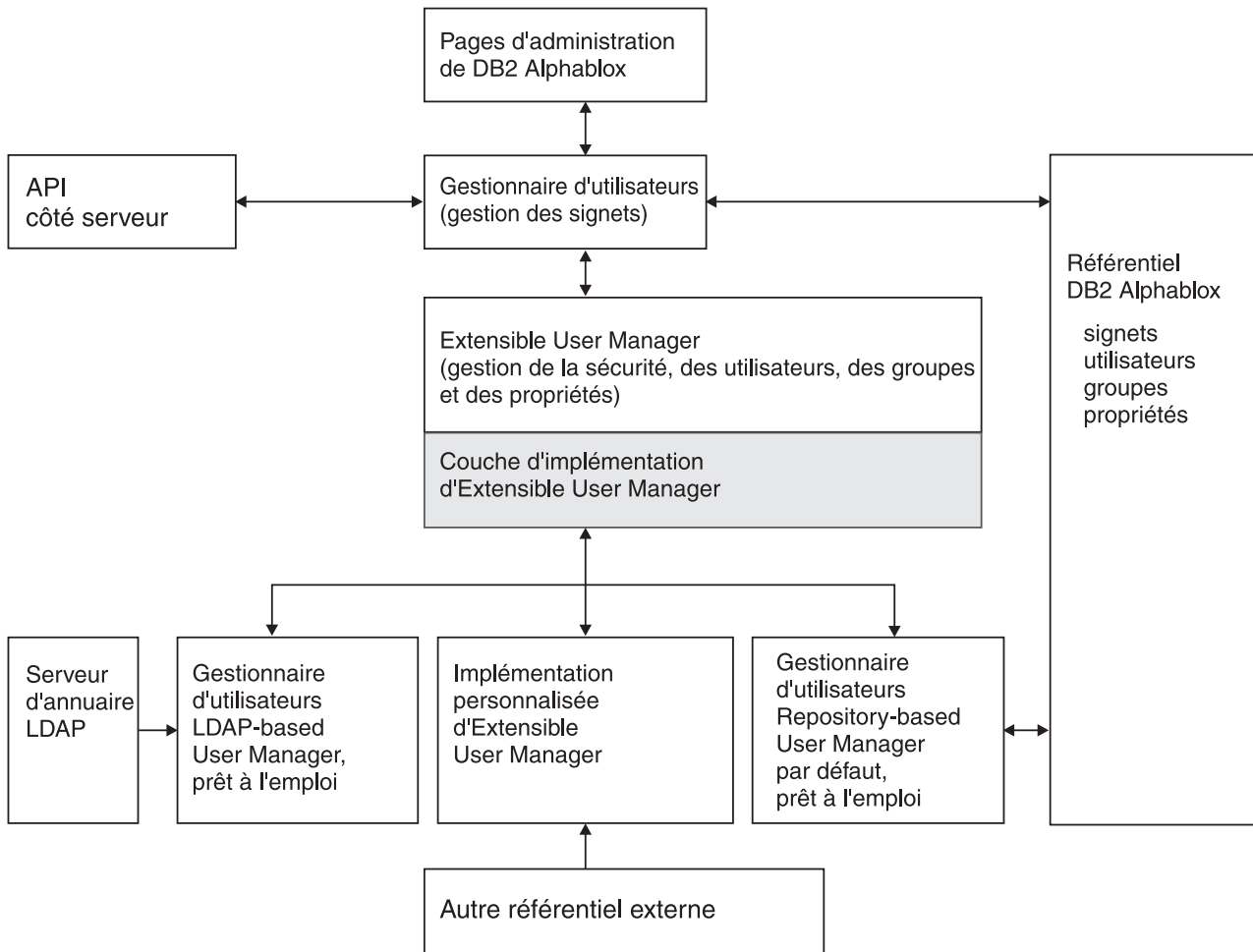
Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Le gestionnaire d'utilisateurs de DB2 Alphablox gère l'authentification et les autorisations des utilisateurs et fournit des fonctions permettant de personnaliser le contenu des applications. Par défaut, DB2 Alphablox utilise le référentiel DB2 Alphablox et l'API de sécurité J2EE pour gérer les informations relatives aux utilisateurs et aux groupes. Deux méthodes de sécurité J2EE (`isUserInRole()` et `getUserPrinciple()`) permettent d'identifier l'utilisateur et de déterminer si celui-ci est associé à un rôle spécifique.

DB2 Alphablox fournit également une solution d'intégration LDAP (Lightweight Directory Access Protocol) prête à l'emploi. Cette solution permet à DB2 Alphablox d'authentifier les utilisateurs et de les autoriser à utiliser un serveur d'annuaire LDAP pour reconnaître les utilisateurs, les groupes et les propriétés personnalisées de DB2 Alphablox.

Le gestionnaire d'utilisateurs est constitué à partir du moteur de personnalisation appelé Extensible User Manager. Dans un environnement où la sécurité doit être personnalisée, le moteur de personnalisation Extensible User Manager fournit des interfaces qui permettent de développer l'une des deux solutions de sécurité prêtes à l'emploi (de type référentiel DB2 Alphablox ou LDAP). Vous pouvez choisir un autre gestionnaire d'utilisateurs externe, par exemple NTLM ou un EJB existant (Enterprise JavaBeans).

Le schéma ci-dessous représente l'architecture du gestionnaire d'utilisateurs et du moteur de personnalisation associé, Extensible User Manager :



Notez que sur le schéma :

- les informations relatives aux utilisateurs, aux groupes et aux propriétés correspondantes sont accessibles par programmation à l'aide de l'API côté serveur via RepositoryBlox.
- Pour la sécurité et la personnalisation, DB2 Alphablox fournit deux solutions prêtes à l'emploi de gestion d'utilisateurs basé sur référentiel DB2 Alphablox ou sur LDAP : DB2 Alphablox Repository-based User Manager et LDAP-based User Manager.
- Le gestionnaire d'utilisateurs basé sur référentiel par défaut lit et écrit des données dans le référentiel DB2 Alphablox.
- Le gestionnaire d'utilisateurs basé sur LDAP prêt à l'emploi n'effectue que des opérations de LECTURE sur le serveur LDAP.
- L'API Extensible User Manager permet d'utiliser un autre référentiel externe, par exemple NTLM ou un EJB existant (Enterprise JavaBeans).
- Quel que soit le référentiel utilisé (DB2 Alphablox, LDAP ou une autre source externe), vous pouvez accéder aux propriétés utilisateur via l'API RepositoryBlox.

Extensible User Manager

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Le moteur de personnalisation Extensible User Manager se trouve au coeur de DB2 Alphablox. Il permet de :

- personnaliser les fonctions prêtes à l'emploi d'authentification et/ou d'autorisation des utilisateurs,
- rattacher un gestionnaire d'utilisateurs externe pour accéder aux informations relatives aux utilisateurs et aux groupes provenant d'autres sources situées hors du référentiel DB2 Alphablox, ou
- utiliser d'autres paramètres de sécurité que ceux de l'API de sécurité J2EE

Le fonctionnement des deux implémentations de sécurité et de personnalisation prêtes à l'emploi DB2 Alphablox (DB2 Alphablox Repository-based User Manager et LDAP-based User Manager) repose sur ce moteur. Pour utiliser le gestionnaire d'utilisateurs basé sur LDAP, une procédure de configuration est nécessaire. Cette procédure est décrite dans «LDAP-Based User Manager», à la page 99. Pour implémenter la sécurité personnalisée ou pour personnaliser l'une des deux implémentations de sécurité prêtes à l'emploi, vous pouvez développer Extensible User Manager. Vous trouverez des informations détaillées sur le développement de l'interface Extensible User Manager dans «Interfaces Extensible User Manager», à la page 103.

LDAP-Based User Manager

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

DB2 Alphablox fournit une implémentation prête à l'emploi d'Extensible User Manager permettant d'utiliser un serveur d'annuaire LDAP pour reconnaître les utilisateurs, les groupes et les propriétés personnalisées de DB2 Alphablox :

- DB2 Alphablox effectue uniquement des opérations de LECTURE sur le serveur LDAP.
- Les administrateurs peuvent uniquement créer des utilisateurs ou des groupes DB2 Alphablox qui sont déjà définis sur le serveur LDAP.
- Les administrateurs ne peuvent pas renommer ni modifier les membres sur l'onglet **Administration** de la page d'accueil DB2 Alphablox.
- La suppression d'utilisateurs ou de groupes via l'onglet **Administration** de la page d'accueil de DB2 Alphablox ne supprime pas les éléments du référentiel sous-jacent et n'affecte pas le serveur LDAP.
- L'utilisateur "admin" n'est plus disponible par défaut sauf s'il s'agit d'un utilisateur LDAP.
- L'utilisateur "guest" et le groupe "public" sont disponibles même s'ils n'existent pas dans LDAP.

Configuration de DB2 Alphablox de façon à utiliser LDAP-based User Manager

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

LDAP-based User Manager peut être utilisé avec les serveurs d'applications WebSphere, WebLogic et Tomcat. Pour configurer DB2 Alphablox de façon à tirer profit de l'intégration prête à l'emploi avec vos serveurs d'annuaire LDAP, procédez comme suit :

1. A l'aide de la commande de console Telnet ExtUserManager, configurez le serveur de façon à ce qu'il utilise LDAP, comme décrit dans «Configuration des propriétés de LDAP-based User Manager», à la page 100.
2. Veillez à ce que la valeur de la propriété de serveur autoCreateUsers soit true à l'aide de la commande de console Telnet suivante :

```
set server autoCreateUsers true
```

Remarque : La valeur par défaut de autoCreateUsers est false. Pour que DB2 Alphablox crée automatiquement les utilisateurs après leur authentification auprès du serveur LDAP, attribuez la valeur true à la propriété.

Remarque : Pour le gestionnaire d'utilisateurs externe, le groupe "public" et l'utilisateur "guest" doivent être définis (il peut s'agir d'instances factices).

Spécifications supplémentaires pour les configurations Apache Tomcat

1. Créez le groupe AlphabloxAdministrator sur le serveur d'annuaire LDAP.
2. Ajoutez au moins un utilisateur au groupe AlphabloxAdministrator ou ajoutez un groupe existant contenant au moins un utilisateur au groupe AlphabloxAdministrator de sorte qu'il en constitue un sous-groupe.

Configuration des propriétés de LDAP-based User Manager

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

1. Établissez une connexion de la console Telnet sur DB2 Alphablox.

Remarque : Si vous accédez à la console Telnet via l'onglet **Administration** de la page d'accueil DB2 Alphablox, redémarrez DB2 Alphablox afin que les modifications soient prises en charge. Sur une console Telnet standard, il n'est pas nécessaire de redémarrer DB2 Alphablox.

2. Entrez la commande suivante :

```
ExtUserManager setToDefaults ldap  
"<ldapProps>"
```

où <ldapProps> sont les paires propriété-valeur séparées par un point-virgule. Par exemple :

```
"host:localhost;port:389;admin:cn=DirectoryManager;  
password:password;debug:false;base:dc=alphablox,dc=com"
```

Remarque : Les valeurs ci-dessus doivent être saisies sur une seule ligne, sans retour à la ligne.

Une fois la commande ExtUserManager exécutée, DB2 Alphablox passe du gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager au gestionnaire d'utilisateurs LDAP-based User Manager.

Remarque : Ce changement entraîne la déconnexion de tous les utilisateurs connectés.

Le tableau suivant répertorie toutes les propriétés obligatoires et facultatives qui peuvent être définies dans la chaîne <ldapProps> :

Propriété	Description	Exemple
host	Hôte du serveur d'annuaire LDAP	host:localhost
port	Port du serveur d'annuaire LDAP	port:389
admin	Nom de l'utilisateur administrateur du serveur d'annuaire LDAP	admin:cn=Directory Manager
password	Mot de passe de l'administrateur du serveur d'annuaire LDAP	password:MonMotDePasse
base	Base de recherche du serveur d'annuaire LDAP	base:o=MonEntreprise.com
debug	[Facultatif] Paramètre du mode de débogage, la valeur par défaut étant false	debug:true

Conseil : La base de recherche LDAP permet de définir le point de départ de la recherche. Elle pointe vers le nom distinctif d'une entrée de la hiérarchie d'annuaire afin d'améliorer les performances de la recherche.

Accès aux propriétés utilisateur personnalisées

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Vous pouvez définir des propriétés utilisateur personnalisées pour personnaliser davantage vos applications. Après avoir défini une propriété utilisateur personnalisée, vous pouvez lui attribuer une valeur différente pour chaque utilisateur, puis accéder par programmation aux valeurs de cette propriété à l'aide de l'API de RepositoryBlox. Avec LDAP-based User Manager ou un gestionnaire d'utilisateurs externe, DB2 Alphablox charge uniquement les propriétés utilisateur personnalisées marquées comme propriétés externes.

Vous pouvez définir les propriétés utilisateur personnalisées sur l'onglet **Administration** de la page d'accueil de DB2 Alphablox. Sur la page de définition des propriétés personnalisées, vérifiez que l'option **External Property** est cochée. Pour connaître la procédure en détail, voir «Définitions des propriétés personnalisées», à la page 82. Après avoir marqué la propriété utilisateur personnalisée comme propriété externe, lorsque les informations sur la valeur de la propriété utilisateur personnalisée sont demandées (par un appel `RepositoryBlox.getUserProperty("maPropUtilisateurExterne")`, par exemple), le gestionnaire d'utilisateurs de DB2 Alphablox récupère la valeur dans la source externe.

Fonctionnement durant la phase d'exécution

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

LDAP-based User Manager fonctionne de la manière suivante durant l'exécution :

- DB2 Alphablox ne crée pas automatiquement les groupes durant l'exécution. Les administrateurs doivent enregistrer explicitement le groupe LDAP via l'onglet **Administration** de la page d'accueil de DB2 Alphablox.

- DB2 Alphablox crée automatiquement les utilisateurs si ceux-ci ont été correctement authentifiés auprès du serveur LDAP et si la valeur true est attribuée à la propriété de serveur autoCreateUsers via une commande Telnet (voir «Configuration de DB2 Alphablox de façon à utiliser LDAP-based User Manager», à la page 99).
- DB2 Alphablox charge uniquement les propriétés utilisateur et groupe LDAP définies comme propriétés personnalisées externes.

Commande de console Telnet pour Extensible User Manager

La commande ExtUserManager sur la console Telnet permet d'indiquer à DB2 Alphablox d'utiliser une autre classe pour gérer les utilisateurs et les groupes ou d'utiliser un

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).
référentiel externe.

La syntaxe générale de la commande ExtUserManager est la suivante :

ExtUserManager <Propriété> <Valeur>

où Propriété peut être :

Propriété	Description	Valeur
umclassname	Nom de la classe de gestionnaire d'utilisateurs	La valeur doit être un nom de classe valide disponible dans le chemin de classes qui implémente l'interface IUserManager, sinon une erreur se produit. Voir «Implémentations personnalisées pour la sécurité», à la page 104 pour plus d'informations.
userclassname	Nom de la classe d'utilisateur	La valeur doit être un nom de classe valide disponible dans le chemin de classes qui implémente l'interface IUser, sinon une erreur se produit. Voir «Implémentations personnalisées pour la sécurité», à la page 104 pour plus d'informations.
groupclassname	Nom de la classe de groupe	La valeur doit être un nom de classe valide disponible dans le chemin de classes qui implémente l'interface IGroup, sinon une erreur se produit. Voir «Implémentations personnalisées pour la sécurité», à la page 104 pour plus d'informations.
customstartupproperty	Propriété d'initialisation personnalisée que les développeurs peuvent insérer dans le code pour lire une valeur lors du démarrage du gestionnaire d'utilisateurs.	La valeur de cette propriété.
setToDefaults	Référentiel à utiliser par défaut	Voir la section suivante intitulée «Configuration du référentiel par défaut», à la page 103.

Remarque : Après avoir défini une nouvelle propriété d'initialisation, arrêtez puis redémarrez DB2 Alphablox.

Configuration du référentiel par défaut

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Pour configurer toutes les propriétés de façon à utiliser le gestionnaire d'utilisateurs LDAP par défaut, entrez la commande Telnet suivante :

```
ExtUserManager setToDefaults ldap <ldapProps>
```

où ldapProps est une propriété contenant toutes les informations nécessaires à la connexion à LDAP. Il n'est pas nécessaire de redémarrer DB2 Alphablox, mais tous les utilisateurs connectés seront déconnectés. Pour connaître la procédure détaillée de la configuration requise pour utiliser un gestionnaire d'utilisateurs LDAP ainsi que la syntaxe de la propriété ldapProps, voir «LDAP-Based User Manager», à la page 99.

Pour réinitialiser toutes les propriétés de façon à utiliser le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager par défaut, entrez la commande Telnet suivante.

```
ExtUserManager setToDefaults repository
```

Cette commande indique au gestionnaire d'utilisateurs de s'arrêter puis de redémarrer afin de récupérer les informations utilisateur les plus récentes sur le référentiel DB2 Alphablox. Il n'est pas nécessaire de redémarrer DB2 Alphablox, mais tous les utilisateurs connectés seront déconnectés.

Suppression d'utilisateurs et de groupes ne se trouvant plus dans le référentiel d'utilisateurs externe

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

La commande suivante indique à Extensible User Manager de supprimer du référentiel les utilisateurs et les groupes qui ne se trouvent plus dans les référentiels d'utilisateurs externes (par exemple LDAP ou NTLM) :

```
ExtUserManager clean
```

Interfaces Extensible User Manager

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

L'infrastructure d'Extensible User Manager comporte trois interfaces principales : IUserManager, IUser et IGroup. Le tableau ci-dessous répertorie la fonction et les méthodes de chaque interface. Il répertorie également la classe qui implémente chaque interface particulière.

Interface	Description	Classe d'implémentation
<i>IUserManager</i>	Personne chargée de la création et de la recherche des utilisateurs et des groupes durant l'exécution.	AbstractUserManager
<i>IUser</i>	Mise en place d'une vue en lecture seule pour un utilisateur du référentiel sous-jacent destinée à l'authentification, l'autorisation et aux propriétés utilisateur.	AbstractUser
<i>IGroup</i>	Mise en place d'une vue en lecture seule destinée aux informations relatives aux membres et aux propriétés de groupe du référentiel sous-jacent	AbstractGroup

Les interfaces et les classes d'implémentation se trouvent dans le module `com.alphablox.personalization`. La documentation Javadoc relative à ce module est disponible dans le répertoire suivant :

`<db2alphablox_dir>/system/documentation/javadoc/blox/index.html`

où `<db2alphablox_dir>` est le répertoire dans lequel est installé DB2 Alphablox.

Implémentations personnalisées pour la sécurité

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Extensible User Manager comporte trois propriétés DB2 Alphablox permettant d'identifier l'utilisateur, le groupe et la classe de gestionnaire d'utilisateurs à utiliser pour l'authentification et l'autorisation :

- `umclassname`
La valeur par défaut pour le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager est `com.alphablox.personalization.repository.RepUserManager`. La valeur par défaut pour le gestionnaire d'utilisateurs LDAP-based User Manager est `com.alphablox.personalization.ldap.LDAPUserManager`.
- `userclassname`
La valeur par défaut de cette propriété pour le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager est `com.alphablox.personalization.repository.RepUser`. La valeur par défaut pour le gestionnaire d'utilisateurs LDAP-based User Manager est `com.alphablox.personalization.ldap.LDAPUser`.
- `groupclassname`
La valeur par défaut de cette propriété pour le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager est `com.alphablox.personalization.repository.RepGroup`. La valeur par défaut pour le gestionnaire d'utilisateurs LDAP-based User Manager est `com.alphablox.personalization.ldap.LDAPGroup`.

Vous pouvez écrire une classe personnalisée d'utilisateur, de groupe ou de gestionnaire d'utilisateurs, puis définir la nouvelle valeur de ces propriétés DB2

Alphablox à l'aide de la commande Telnet ExtUserManager de façon à les faire pointer vers votre classe d'utilisateur, de groupe ou de gestionnaire d'utilisateurs.

Le tableau suivant indique la classe à développer pour personnaliser une partie ou l'ensemble du schéma de sécurité prêt à l'emploi.

Objectif	Solution
Authentification ou autorisation avec le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager par défaut	Extension de la classe AlphabloxUser
Authentification ou autorisation avec le gestionnaire d'utilisateurs LDAP-based User Manager	Extension de la classe ldapUser
Connexion unique	Implémentation de la méthode getPassword() dans la classe AlphabloxUser
Lecture des propriétés utilisateurs/groupe dans un autre référentiel que DB2 Alphablox ou LDAP	Extension de la classe AlphabloxUser ou AlphabloxGroup
Lecture des informations relatives aux membres de groupes dynamiques dans un autre référentiel que DB2 Alphablox ou LDAP	Extension de la classe AlphabloxUser ou AlphabloxGroup
Prise en charge NTLM	Implémentation complète des interfaces Extensible User Manager

Remarque : Afin que DB2 Alphablox utilise vos classes, vous devez ajouter le répertoire de ces classes au chemin de classes de DB2 Alphablox défini dans le fichier de traitement par lots de l'initialisation. Pour plus de détails sur la manière de définir le chemin de classes, voir «Définition du chemin de classes», à la page 75.

Connexion unique

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Pour implémenter la connexion unique lorsque les sources de données DB2 Alphablox sont configurées pour utiliser les noms d'utilisateur et les mots de passe DB2 Alphablox, vous pouvez utiliser la méthode getPassword() dans la classe AlphabloxUser pour obtenir le mot de passe codé en base64. Le tableau suivant présente les solutions de connexion unique dans différentes situations :

Gestionnaire d'utilisateurs	Sources de données avec LDAP ?	Méthode d'authentification
LDAP-based User Manager	Oui	Authentification DB2 Alphablox standard (au niveau du navigateur) : aucune action requise
LDAP-based User Manager	Non	Utilisez getPassword()
Autre référentiel	Indifférent	Utilisez getPassword()

Exemples de sécurité personnalisée

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Voici trois exemples d'extension d'Extensible User Manager pour personnaliser la sécurité. Chacun des exemples montre :

- comment étendre la classe d'utilisateur, de groupe ou de gestionnaire d'utilisateurs par défaut utilisée par le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager par défaut, puis
- définir le nom de la nouvelle classe à l'aide de la commande Telnet ExtUserManager.

Pour obtenir un exemple d'implémentation complète d'un gestionnaire d'utilisateurs externe, voir l'exemple de gestionnaire d'utilisateurs (Simple User Manager) dans le répertoire suivant :

```
<db2alphablox_dir>/system/documentation/admin/Examples/
```

Les fichiers source Java sont fournis.

Exemple 1 : Configuration de DB2 Alphablox pour utiliser un gestionnaire d'utilisateurs externe

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Cet exemple présente une manière de personnaliser le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager par défaut. Dans ce cas, la liste de propriétés personnalisées peut être lue dans un fichier lors du démarrage du gestionnaire d'utilisateurs. La procédure inclut les étapes suivantes :

1. Ecriture d'une classe de gestionnaire d'utilisateurs personnalisée dans un module (dans cet exemple, la classe s'appelle MyUserManager et le module qui la contient s'appelle com.myCompany.user)
 - La classe étend la classe AlphabloxUserManager utilisée dans le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager
2. Configuration de la nouvelle valeur de la propriété de serveur umclassname à l'aide de la commande Telnet suivante :
ExtUserManager umclassname com.myCompany.user.MyUserManager

La classe MyUserManager personnalisée peut être la suivante :

```
package com.myCompany.user;
import com.alphablox.personalization.alphablox.*;
import com.alphablox.personalization.*;
import java.util.*;

public class MyUserManager extends AlphabloxUserManager {
    void start(Properties props) throws PEngineException {
        super.start(props);
        String myXmlFile = (String) prop.get("customstartupprop");
        // lecture du fichier xml et autres opérations
    }
}
```


Exemple 2 : Configuration de DB2 Alphablox pour utiliser une classe d'utilisateur différente

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Cet exemple présente une manière de personnaliser les autorisations utilisateur dans le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager par défaut. Dans ce cas, vous souhaitez autoriser tous les utilisateurs à accéder à DB2 Alphablox ou à une application et à remplacer la méthode `isUserInRole()` sur l'interface `IUser`. La procédure inclut les étapes suivantes :

1. Ecriture d'une classe d'utilisateur personnalisée dans un module (dans cet exemple, la classe s'appelle `MyUser` et le module qui la contient s'appelle `com.myCompany.user`)
 - La classe étend la classe `AlphabloxUser` par défaut utilisée dans le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager
2. Configuration de la nouvelle valeur de la propriété de serveur `userclassname` à l'aide de la commande Telnet suivante :

```
ExtUserManager userclassname com.myCompany.user.MyUser
```

où `MyUser` peut être l'exemple suivant :

```
package com.myCompany.user;
import com.alphablox.personalization.repository.*;
import com.alphablox.personalization.*;
import java.util.*;

public class MyRepositoryUser extends AlphabloxUser {
    public boolean isUserInRole(HttpServletRequest req,
        String [] roles) throws PEngineException {
        // Tous les utilisateur y ont accès
        return true;
    }
}
```

Exemple 3 : Configuration de DB2 Alphablox pour utiliser une classe de groupe différente

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Cet exemple présente une manière de personnaliser le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager par défaut pour imprimer un message système à chaque vérification d'appartenance à un groupe. La procédure inclut les étapes suivantes :

1. Ecriture d'une classe de groupe personnalisée dans un module (dans cet exemple, la classe s'appelle `MyGroup` et le module qui la contient s'appelle `com.myCompany.user`)
 - La classe étend la classe `AlphabloxGroup` par défaut utilisée dans le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager
2. Configuration de la nouvelle valeur de la propriété de serveur `groupclassname` à l'aide de la commande Telnet suivante :

```
ExtUserManager groupclassname com.myCompany.user.MyGroup
```

où `MyGroup` peut être l'exemple suivant :

```
package com.myCompany.user;
import com.alphablox.personalization.alphablox.*;
import com.alphablox.personalization.*;
import java.util.*;
```

```

public class MyGroup extends AlphabloxGroup {
    public boolean containsUser(IUser user,
        boolean checkSubGroups) throws PEngineException {
        boolean exists = super.containsUser(user, checkSubGroups);
        System.out.println("User membership checked.");
        return exists;
    }
}

```

Références relatives aux méthodes d'interface

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

Cette section indique la documentation de référence relative aux trois interfaces d'Extensible User Manager. Chaque méthode utilisée dans ces interfaces, ainsi que la syntaxe correspondante sont décrites et accompagnées de remarques sur leur utilisation.

Pour obtenir un exemple de gestionnaire d'utilisateurs personnalisé, voir l'exemple de gestionnaire d'utilisateurs (Simple User Manager) dans le répertoire suivant :

<db2alphablox_dir>/system/documentation/admin/Examples/

Interface	Méthodes
<i>IUserManager</i>	findGroup() findUser() getExternalProperties() getPrincipleUserName() hasExternalEditor() resume() setCaseSensitiveGroups() setCaseSensitiveUsers() start() stop() suspend()
<i>IUser</i>	authenticate() authorize() getName() getPassword() getPropertiesSubset() isUserInRole() refresh()
<i>IGroup</i>	containsGroup() containsUser() getName() getPropertiesSubset() refresh()

Interface IUserManager

Remarque : Cette section s'applique uniquement à DB2 Alphablox 8.4 (pas de prise en charge dans la version 8.4.1).

La classe `AbstractUserManager` implémente l'interface `IUserManager` pour trouver des informations sur les utilisateurs et les groupes et déterminer l'identité des utilisateurs durant l'exécution. Pour étendre cette classe, ajoutez l'instruction d'importation suivante dans votre code :

```
import com.alphablox.personalization.*;
```

Pour consulter un exemple d'extension de la classe `AbstractUserManager`, voir «Exemple 1 : Configuration de DB2 Alphablox pour utiliser un gestionnaire d'utilisateurs externe», à la page 106.

findGroup()

Permet de trouver un groupe et de renvoyer l'instance appropriée de `IGroup`.

Syntaxe

```
IGroup findGroup(String id, boolean fromCache);  
                // émission de PEngineException
```

où :

Argument	Description
<code>id</code>	ID du groupe.
<code>fromCache</code>	Recherche du groupe dans la mémoire cache

Utilisation

Cette méthode renvoie la valeur null :

- si le groupe est absent de la mémoire et la valeur de `fromCache` est true
- si le groupe est absent de la mémoire, ce n'est pas un groupe valide dans le référentiel du gestionnaire d'utilisateurs sous-jacent, et la valeur de `fromCache` est true

findUser()

Permet de trouver un utilisateur et de renvoyer l'instance appropriée de `IUser`.

Syntaxe

```
IUser findUser(String id, boolean fromCache);  
                // émission de PEngineException
```

où :

Argument	Description
<code>id</code>	ID utilisateur
<code>fromCache</code>	Recherche de l'utilisateur dans la mémoire cache

Utilisation

Cette méthode renvoie la valeur null :

- si l'utilisateur est absent de la mémoire et la valeur de `fromCache` est true

- si l'utilisateur est absent de la mémoire, ce n'est pas un utilisateur valide dans le référentiel du gestionnaire d'utilisateurs sous-jacent, et la valeur de *fromCache* est true

getExternalProperties()

Renvoie un tableau de type "chaîne de caractères" contenant les propriétés définies via l'éditeur externe.

Syntaxe

```
String[] getExternalProperties(); //émission de PEngineException
```

getPrincipleUserName()

Renvoie le nom d'utilisateur associé à cette requête.

Syntaxe

```
String getPrincipleUserName(HttpServletRequest requête);  
// émission de PEngineException
```

où :

Argument	Description
requête	Requête HTTP en cours

Utilisation

Cette méthode permet à DB2 Alphablox de déterminer le nom d'utilisateur lors de la création d'une nouvelle session DB2 Alphablox. Vous pouvez utiliser cette méthode pour remplacer le fonctionnement par défaut de DB2 Alphablox.

hasExternalEditor()

Indique si le gestionnaire d'utilisateurs externe est associé à un éditeur.

Syntaxe

```
boolean hasExternalEditor();
```

Utilisation

Renvoie la valeur true si le gestionnaire d'utilisateurs externe est associé à un éditeur particulier. Par exemple, cette méthode renvoie la valeur true si vous utilisez le gestionnaire d'utilisateurs LDAP-based User Manager. Elle renvoie la valeur false si vous utilisez le gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager par défaut. Cette méthode ne doit être implémentée pour renvoyer la valeur true que dans le cas d'une implémentation complète d'Extensible User Manager avec un éditeur externe.

resume()

Relance le gestionnaire d'utilisateurs.

Syntaxe

```
void resume(); // émission de PEngineException
```

Utilisation

Cette méthode est appelée lorsque le service du gestionnaire d'utilisateurs DB2 Alphablox est relancé. Appelez cette méthode pour relancer le gestionnaire d'utilisateurs DB2 Alphablox après une interruption du service.

Exemples

Le code suivant consigne un message indiquant que le gestionnaire d'utilisateurs a été relancé :

```
import com.alphablox.personalization.*;
public class MyUserManager extends AbstractUserManager {
    ...
    public void resume() throws PEngineException {
        System.out.println("Resumed");
    }
    ...
}
```

Voir aussi

«suspend()», à la page 112

setCaseSensitiveGroups()

Indique si les noms de groupe sont dépendants des majuscules/minuscules.

Syntaxe

```
void setCaseSensitiveGroups(boolean caseSensitive);
// émission de PEngineException
```

où :

Argument	Description
caseSensitive	true – les noms de groupe sont dépendants des majuscules/minuscules

Utilisation

Si la valeur de cette méthode est true, les noms de groupe sont dépendants des majuscules/minuscules et le gestionnaire d'utilisateurs respecte cette condition.

setCaseSensitiveUsers()

Indique si les noms d'utilisateur sont dépendants des majuscules/minuscules.

Syntaxe

```
void setCaseSensitiveUsers(boolean caseSensitive);
// émission de PEngineException
```

où :

Argument	Description
caseSensitive	true — les noms d'utilisateur sont dépendants des majuscules/minuscules

Utilisation

Si la valeur de cette méthode est true, les noms d'utilisateur sont dépendants des majuscules/minuscules et le gestionnaire d'utilisateurs respecte cette condition.

start()

Lance le gestionnaire d'utilisateurs.

Syntaxe

```
void start(java.util.Properties props);
// émission de PEngineException
```

où :

Argument	Description
props	Propriétés DB2 Alphablox liées à la personnalisation. Propriétés valides : <code>umclassname</code> , <code>groupclassname</code> , <code>customproperty</code> et <code>ldapprops</code> . Pour plus de détails, voir «Commande de console Telnet pour Extensible User Manager», à la page 102.

Utilisation

Cette méthode est appelée à chaque démarrage du gestionnaire d'utilisateurs DB2 Alphablox. Cette méthode permet d'établir la connexion avec le référentiel approprié et d'instancier les objets utilisateur et groupe selon les besoins. Dans le cas d'un gestionnaire d'utilisateurs DB2 Alphablox Repository-based User Manager, tous les utilisateurs et les groupes du référentiel sont instanciés. Dans le gestionnaire d'utilisateurs LDAP-based User Manager, seuls les utilisateurs et les groupes enregistrés dans DB2 Alphablox doivent être créés. Par conséquent, il peut être utile de créer des utilisateurs et des groupes selon les besoins, par exemple lorsque les méthodes `findUser()` et `findGroup()` sont appelées.

stop()

Arrête le gestionnaire d'utilisateurs et libère toutes les ressources.

Syntaxe

```
void stop(); // émission de PEngineException
```

Utilisation

Cette méthode est appelée lorsque le service du gestionnaire d'utilisateurs DB2 Alphablox est arrêté.

Voir aussi

«start()», à la page 111, «suspend()», à la page 112.

suspend()

Interrompt l'exécution du gestionnaire d'utilisateurs et libère les ressources non utilisées, par exemple la connexion à la base de données.

Syntaxe

```
void suspend(); // émission de PEngineException
```

Utilisation

Cette méthode est appelée lorsque le service du gestionnaire d'utilisateurs DB2 Alphablox est interrompu.

Voir aussi

«resume()», à la page 110

Interface IUser

Pour étendre cette classe, ajoutez l'instruction d'importation ci-dessous au code :

```
import com.alphablox.personalization.*;
```

Pour obtenir un exemple d'extension de la classe `AbstractUser`, voir «Exemple 2 : Configuration de DB2 Alphablox pour utiliser une classe d'utilisateur différente», à la page 107.

authenticate()

Détermine si le nom d'utilisateur et le mot de passe indiqués via la requête HTTP sont identiques à ceux stockés dans le référentiel.

Syntaxe

```
boolean authenticate(HttpServletRequest request,  
                    String authorizationHeader);  
// émission de PEngineException
```

où :

Argument	Description
request	Requête HTTP.
authorizationHeader	Détermine si le groupe doit être recherché dans la mémoire cache

Utilisation

Renvoie la valeur true si le mot de passe enregistré par l'utilisateur est identique au mot de passe du paramètre. Cette méthode s'applique uniquement dans les configurations DB2 Alphablox Apache Tomcat (Apache Tomcat et non Microsoft IIS) où DB2 Alphablox utilise l'authentification standard (c'est-à-dire au niveau du navigateur). Cette méthode est appelée par les intercepteurs DB2 Alphablox Tomcat lorsque l'utilisateur accède à la session pour la première fois. La méthode comporte deux paramètres : l'objet de requête et l'en-tête d'autorisation codée. Cet en-tête contient le nom d'utilisateur et le mot de passe. Vous pouvez utiliser la méthode suivante pour décoder l'en-tête :

```
AbstractUserManager.getDecoder().decode(authorizationHeader);
```

Une fois le mot de passe obtenu, vous pouvez l'enregistrer dans la mémoire de l'objet si vous prévoyez de l'utiliser comme valeur de retour de la méthode getPassword() afin de permettre la connexion unique à la source de données.

authorize()

Détermine si l'utilisateur se trouve dans la liste des rôles.

Syntaxe

```
boolean authorize(HttpServletRequest request,  
                 String[] roles);  
// émission de PEngineException
```

où :

Argument	Description
request	Requête HTTP.
roles	Liste des rôles dans laquelle l'utilisateur est recherché pour déterminer s'il est associé à un rôle.

Utilisation

Renvoie la valeur true si l'utilisateur se trouve dans l'un des rôles de la liste.

getEmail()

Obtention de l'adresse électronique de l'utilisateur.

Syntaxe

```
String getEmail(); // émission de PEngineException
```

Utilisation

Renvoie la valeur null si DB2 Alphablox gère le nom de l'adresse électronique au lieu de le lire dans le référentiel externe. Dans le cas de LDAP, l'adresse électronique et le nom complet de l'utilisateur sont récupérés depuis LDAP et leur valeur n'est pas modifiable dans les pages Admin de DB2 Alphablox. Dans ce cas, vous pouvez utiliser cette méthode pour obtenir l'adresse électronique de l'utilisateur.

getFullName()

Obtention du nom complet de l'utilisateur.

Syntaxe

```
String getFullName(); // émission de PEngineException
```

Utilisation

Renvoie la valeur null si DB2 Alphablox gère le nom complet au lieu de le lire dans le référentiel externe. Dans le cas de LDAP, l'adresse électronique et le nom complet de l'utilisateur sont récupérés depuis LDAP et leur valeur n'est pas modifiable dans les pages Admin de DB2 Alphablox. Dans ce cas, vous pouvez utiliser cette méthode pour obtenir le nom complet de l'utilisateur.

getName()

Obtention du nom de l'utilisateur.

Syntaxe

```
String getName(); // émission de PEngineException
```

Utilisation

Le nom de l'utilisateur doit être un nom d'utilisateur valide dans DB2 Alphablox.

getPassword()

Obtention du mot de passe de l'utilisateur.

Syntaxe

```
String getPassword(); // émission de PEngineException
```

Utilisation

Obtention du mot de passe de l'utilisateur, codé en base64. Si la source de données est configurée de façon à utiliser le nom d'utilisateur et le mot de passe DB2 Alphablox, DB2 Alphablox utilise le mot de passe codé enregistré dans le référentiel pour l'authentification et l'accès à la source de données. Utilisez cette méthode pour la sécurité personnalisée, telle que la connexion unique aux sources de données. Tout mot de passe enregistré par cette méthode sera utilisé temporairement dans la mémoire. Notez que ce mot de passe doit être codé à l'aide du code suivant :

```
AbstractUserManager.getEncoder().encode(password);
```

avant d'être renvoyé aux demandeurs.

Voir aussi

«Connexion unique», à la page 105.

getPropertiesSubset()

Obtention du sous-ensemble de propriétés utilisateur pertinentes.

Syntaxe

```
java.util.Properties getPropertiesSubset(String[] propList);  
// émission de PEngineException
```

où :

Argument	Description
propList	Tableau de type "chaîne de caractères" du sous-ensemble de propriétés

Utilisation

Renvoie un objet de propriétés pour le sous-ensemble des propriétés utilisateur transmises. Les valeurs sont obtenues depuis la mémoire, sauf si le développeur modifie cela.

isUserInRole()

Détermine si l'utilisateur appartient à l'un des rôles définis.

Syntaxe

```
boolean isUserInRole(HttpServletRequest request,  
String[] roles);  
// émission de PEngineException
```

où :

Argument	Description
request	Requête HTTP
roles	Rôle(s) associé(s) à l'utilisateur.

Utilisation

Renvoie la valeur true si l'utilisateur appartient à l'un des rôles de la liste. L'implémentation des rôles est effectuée par les développeurs. L'implémentation de cette méthode repose sur la méthode standard `Request.isUserInRole(Stringrole)` de l'API J2EE. DB2 Alphablox utilise cette méthode pour déterminer l'accès utilisateur à la fonctionnalité d'administration et l'accès en écriture aux signets.

Avec le gestionnaire d'utilisateurs LDAP-based User Manager, cette méthode appelle la méthode `IGroup.containsUser()`, car les rôles sont équivalents aux groupes LDAP.

Voir aussi

«containsGroup()», à la page 116

refresh()

Régénère toutes les informations mises en mémoire cache et récupère les informations les plus récentes dans le référentiel sous-jacent.

Syntaxe

```
void refresh(); // émission de PEngineException
```

Utilisation

Provoque la régénération des informations utilisateur enregistrées en mémoire et récupère les informations les plus récentes dans le référentiel externe.

Interface IGroup

Pour étendre cette classe, ajoutez l'instruction d'importation ci-dessous au code :

```
import com.alphablox.personalization.*;
```

Pour consulter un exemple d'extension de la classe `AbstractGroup`, voir «Exemple 3 : Configuration de DB2 Alphablox pour utiliser une classe de groupe différente», à la page 107.

containsGroup()

Détermine si le groupe contient un sous-groupe.

Syntaxe

```
boolean containsGroup(IGroup group, boolean checkSubGroups);  
// émission de PEngineException
```

où :

Argument	Description
<code>group</code>	Instance de <code>IGroup</code> .
<code>checkSubGroups</code>	true — vérification des sous-groupes; false — pas de vérification des sous-groupes.

Utilisation

Renvoie la valeur `true` si le groupe contient un sous-groupe.

containsUser()

Détermine si le groupe contient l'utilisateur indiqué.

Syntaxe

```
boolean containsUser(IUser user, boolean checkSubGroups);  
// émission de PEngineException
```

où :

Argument	Description
<code>user</code>	Instance de <code>IUser</code> .
<code>checkSubGroups</code>	true — vérification des sous-groupes; false — pas de vérification des sous-groupes

Utilisation

Renvoie la valeur `true` si le groupe contient l'utilisateur indiqué.

getName()

Obtention du nom du groupe.

Syntaxe

```
String getName(); // émission de PEngineException
```

Utilisation

Le nom du groupe doit être un nom de groupe valide dans DB2 Alphablox.

getPropertiesSubset()

Obtention du sous-ensemble de propriétés du groupe pertinentes.

Syntaxe

```
java.util.Properties getPropertiesSubset(String[] propList);  
// émission de PEngineException
```

où :

Argument	Description
propList	Tableau de type "chaîne de caractères" contenant les propriétés

Utilisation

Renvoie un objet de propriétés pour au moins la liste des propriétés du groupe transmises. Les valeurs sont obtenues depuis la mémoire, sauf si le développeur modifie cela.

refresh()

Régénère toutes les informations mises en mémoire cache et récupère les informations les plus récentes dans le référentiel sous-jacent.

Syntaxe

```
void refresh(); // émission de PEngineException
```

Utilisation

Provoque la régénération des informations du groupe enregistrées en mémoire et récupère les informations les plus récentes dans le référentiel externe.

Chapitre 16. Utilisation d'un référentiel de base de données

Ce chapitre décrit comment configurer DB2 Alphablox pour utiliser une base de données relationnelle pour le référentiel DB2 Alphablox.

Présentation du référentiel DB2 Alphablox

Le référentiel DB2 Alphablox est un magasin d'objets que DB2 Alphablox utilise pour garder la trace des applications, des utilisateurs, des groupes, des signets et d'autres informations. Le référentiel peut se trouver dans le système de fichiers du système d'exploitation ou dans une base de données relationnelle. Si une configuration DB2 Alphablox est définie pour l'exécution dans un environnement groupé, le référentiel doit se trouver dans une base de données relationnelle. La base de données relationnelle permet à plusieurs noeuds de serveurs de lire et écrire dans le référentiel, pour assurer un état cohérent au niveau des données du référentiel.

Référentiel dans l'environnement DB2 Alphablox

Dans DB2 Alphablox, le gestionnaire de référentiels contrôle l'accès au référentiel. Le gestionnaire de référentiels utilise l'interface JNDI (Java Naming and Directory Interface) pour communiquer avec le référentiel, que celui-ci se trouve dans un système de fichiers ou dans une base de données relationnelle. La couche JNDI utilise un fournisseur de services différent pour les différents types de référentiels : système de fichiers DB2 Alphablox ou base de données.

Les bases de données prises en charge pour le référentiel sont les mêmes que celles prises en charge pour l'accès à la source de données, comme il est décrit dans le *Guide d'installation*. Pour plus d'informations sur l'architecture de DB2 Alphablox, voir «Architecture DB2 Alphablox», à la page 9.

L'utilitaire de conversion du référentiel est un programme Java utilisé pour convertir le référentiel entre un système de fichiers et un autre système de fichiers, un système de fichiers et une base de données, une base de données et un système de fichiers, ou entre différentes bases de données. L'utilitaire de conversion s'exécute automatiquement pendant le processus d'installation ou il peut s'exécuter manuellement. Pour plus d'informations sur l'utilisation de l'utilitaire de conversion, voir «Utilisation de l'utilitaire de conversion du registre», à la page 120.

Avantages du référentiel relationnel

L'utilisation d'une base de données relationnelle pour le référentiel DB2 Alphablox présente les deux avantages principaux suivants :

- Plusieurs serveurs peuvent accéder à un référentiel relationnel, ce qui permet d'obtenir un environnement groupé pour DB2 Alphablox et de fournir une possibilité d'évolutivité pratiquement illimitée.
- Un référentiel relationnel vous offre tous les outils professionnels disponibles dans un environnement de base de données pour toutes les opérations d'intégrité transactionnelle, de sauvegarde et de restauration, de retour à un état cohérent, de réplication de bases de données, etc.

Le référentiel stocke des objets qui sont indispensables au fonctionnement de DB2 Alphablox. Le fait de disposer de tous les outils d'une base de données pour assurer l'intégrité des données garantit un système plus robuste et plus fiable.

Configuration du référentiel DB2 Alphablox

L'état initial du référentiel DB2 Alphablox est établi pendant le processus d'installation. Le référentiel DB2 Alphablox utilise deux *fournisseurs de services* pour accéder au référentiel, selon son type : le **fournisseur de services de système de fichiers DB2 Alphablox** et le **fournisseur de services de base de données DB2 Alphablox**. Le **fournisseur de services de système de fichiers DB2 Alphablox** est l'état par défaut. Toutes les bases de données relationnelles qui sont prises en charge par DB2 Alphablox sont des bases de données prises en charge pour le **fournisseur de services de base de données DB2 Alphablox**. Pour plus d'informations sur la spécification du type de référentiel pendant le processus d'installation et sur les bases de données relationnelles prises en charge, voir le *Guide d'installation*.

Vérification du type de référentiel

Pour vérifier le fournisseur de services du référentiel qui est utilisé pour votre serveur, effectuez les étapes suivantes :

1. Assurez-vous que DB2 Alphablox est en cours d'exécution.
2. Connectez-vous à la page d'accueil de DB2 Alphablox en tant qu'utilisateur *admin* ou en tant qu'utilisateur membre du groupe administrateur.
3. Cliquez sur l'onglet **Administration**. La page **General** sous l'onglet **Administration** s'affiche.
4. Dans la section des **propriétés générales** de la page, cliquez sur le lien du **gestionnaire de référentiels** pour afficher les propriétés du référentiel DB2 Alphablox.

L'entrée correspondant au **fournisseur de services de référentiel** indique le fournisseur de services JNDI utilisé, l'entrée de l'**adaptateur de base de données** indique le pilote JDBC utilisé, et les autres entrées indiquent des informations de configuration de la source de données. Si le fournisseur de services de système de fichiers est utilisé, l'entrée d'**emplacement du référentiel** indique le répertoire de stockage des fichiers de référentiel.

Remarque : Vous ne pouvez pas modifier le type de référentiel dans la page des propriétés du gestionnaire de référentiels. Pour modifier le type de référentiel, vous devez utiliser l'utilitaire de conversion.

Utilisation de l'utilitaire de conversion du registre

Si vous voulez changer le type de référentiel (par exemple, de système de fichiers en une base de données Oracle), vous devez exécuter l'utilitaire de conversion du registre. L'utilitaire de conversion est un programme Java qui s'exécute dans une fenêtre de ligne de commande (fenêtre MS-DOS sous les systèmes Windows, xterm ou autre fenêtre de commande sous les systèmes Linux et UNIX). Il crée les tables et/ou les fichiers nécessaires pour déplacer un référentiel ou registre DB2 Alphablox d'un emplacement vers un autre. Vous pouvez utiliser l'utilitaire de conversion du registre pour déplacer un référentiel d'un système de fichiers vers un autre système de fichiers, d'un système de fichiers vers une base de données, d'une base de données vers un système de fichiers, ou d'une base de données vers une autre base de données.

Démarrage de l'utilitaire de conversion du registre

Vous pouvez démarrer l'utilitaire de conversion du registre à partir du menu **Démarrer** sous les systèmes Windows sur lesquels DB2 Alphablox est installé ou en exécutant le fichier suivant sous les systèmes Windows.

Important : Vous devez toujours arrêter DB2 Alphablox avant d'exécuter l'utilitaire de conversion du registre.

```
<db2alphablox_dir>/Tools/convert/ConvertRepository.exe
```

ou en exécutant le fichier suivant sous les systèmes Linux et UNIX :

```
<db2alphablox_dir>/Tools/convert/ConvertRepository
```

où <db2alphablox_dir> représente le répertoire dans lequel DB2 Alphablox est installé.

Options interactives de ligne de commande de l'utilitaire de conversion du registre

Le tableau suivant décrit les options interactives de ligne de commande du menu principal de l'utilitaire de conversion du registre :

Option	Description
1 Set DB2 Alphablox File Manager Root	Définit le répertoire pour l'emplacement des fichiers de référentiel utilisés pour un référentiel de système de fichiers. Le répertoire doit être un répertoire DB2 Alphablox valide pour éviter l'échec de l'opération et la génération d'une erreur.
2 Set DB2 Alphablox Instance Name	Sélectionne l'instance de DB2 Alphablox à laquelle l'utilitaire de conversion du registre accédera. L'instance spécifiée doit être une instance DB2 Alphablox valide pour éviter l'échec de l'opération et la génération d'une erreur.

Option	Description
3 Convert One Repository to Another	<p>Convertit un référentiel existant en un autre référentiel, en déplaçant toutes les données nécessaires d'un référentiel à l'autre. D'autres options permettent de convertir le référentiel d'un fichier en une base de données, d'une base de données en un fichier, d'un fichier en un autre fichier et d'une base de données en une autre base de données. Vous êtes invité à utiliser l'une des options suivantes lorsque vous convertissez un référentiel :</p> <ul style="list-style-type: none"> • COPY : copie le contenu d'un référentiel dans un autre, tout en laissant le référentiel original à sa place. • MOVE : déplace le contenu d'un référentiel vers un autre référentiel. Le référentiel original est supprimé. <p>Vous êtes également invité à utiliser l'une des options suivantes pour le référentiel de destination :</p> <ul style="list-style-type: none"> • NEW : crée un référentiel à la destination indiquée. • UPDATE : met à jour les données dans le référentiel de destination depuis le répertoire source au lieu de le remplacer par un nouveau référentiel. • OVERWRITE : recrée les tables et remplace toutes les données dans le référentiel de destination par les données du référentiel source. L'option OVERWRITE ne conserve pas les données dans le référentiel de destination.
4 Create an Empty Database Repository	<p>Crée les tables dans une base de données requise pour le référentiel DB2 Alphablox. Cette option permet de créer uniquement les tables avec leur contenu initial ; elle ne les remplit pas avec des objets de référentiel.</p>
5 Verify and Repair a Repository	<p>Vérifie si un référentiel présente des problèmes et, le cas échéant, signale ces problèmes. Inclut des options permettant de vérifier les référentiels de fichiers et de base de données. Dans le cas des référentiels de bases de données, cette option permet également de corriger certains problèmes.</p>
6 Change DB2 Alphablox to use a different Repository	<p>Renvoie une instance de DB2 Alphablox vers un autre référentiel. Le référentiel vers lequel vous pointez doit exister et être accessible depuis la machine DB2 Alphablox.</p>

Option	Description
7 Conversion Utility Options	Permet de passer du mode de message prolix au mode de message normal et inversement pour l'utilitaire et d'effacer la mémoire tampon contenant l'historique que l'utilitaire de conversion utilise pour mémoriser les informations entrées. Fournit également une option permettant de spécifier différents fichiers de schéma DDL pour créer les tables de référentiel. Ne spécifiez pas de fichiers DDL différents sans les avoir testés de manière approfondie.
8 Exit	Quitte l'utilitaire de conversion du registre.

L'utilitaire de conversion du registre conserve un fichier journal intitulé `repositoryconvert.log` dans lequel sont consignées toutes les activités de la session de conversion de l'utilitaire. Le fichier `repositoryconvert.log` se trouve dans le même répertoire que l'utilitaire de conversion (`<db2alphablox_dir>/Tools/convert`).

Conversion d'un système de fichiers en base de données

Avant de convertir un référentiel d'un système de fichiers en une base de données, vous devez disposer des informations suivantes :

- le chemin complet vers votre référentiel de système de fichiers (par exemple, `d:\alphablox\repository`),
- les informations de connexion à la base de données.

La procédure suivante décrit les étapes nécessaires pour convertir un référentiel DB2 Alphablox de système de fichiers en base de données.

1. Arrêtez DB2 Alphablox.
2. Démarrez l'utilitaire de conversion de registre (pour plus de détails, voir «Démarrage de l'utilitaire de conversion du registre», à la page 121).
3. Sélectionnez l'option 3, **Convert one repository to another**, en entrant le numéro 3 et en appuyant sur la touche Entrée.
4. Sélectionnez l'option 1, **Convert file to database**, en entrant le numéro 1 et en appuyant sur la touche Entrée.
5. Confirmez le répertoire racine du référentiel en appuyant sur la touche Entrée.

Remarque : Si le répertoire indiqué comme étant le répertoire par défaut n'est pas le répertoire du référentiel, assurez-vous que vous accédez bien à l'instance correcte de DB2 Alphablox. L'instance par défaut à laquelle accède l'utilitaire de conversion de registre est le nom de l'instance installée, qui par défaut indique `AlphabloxAnalytics`. Si le nom de votre instance est différent, quittez la séquence en cours et définissez le nom d'instance correct à partir de l'option 2 dans le menu principal.

6. Si tous les éléments sont corrects, sélectionnez **Continuer** en entrant le numéro 1 et en appuyant sur la touche Entrée.
7. Sélectionnez la base de données qui correspond à votre serveur de base de données. Par exemple, si vous utilisez Oracle 8.1.7, entrez 2.
8. Entrez les informations de configuration pour votre base de données lorsque vous y êtes invité.

9. Si tous les éléments sont corrects, sélectionnez **Continuer** en entrant le numéro **1** et en appuyant sur la touche Entrée.
10. Entrez COPY ou MOVE. COPY conserve l'ancien référentiel à sa place et en crée une copie dans le référentiel de destination ; MOVE supprime l'ancien référentiel et en crée un nouveau dans le référentiel de destination.
11. Entrez NEW, UPDATE ou OVERWRITE. Utilisez NEW pour créer un référentiel avec de nouvelles tables s'il n'en existe pas une déjà. Utilisez UPDATE pour conserver la structure de table existante du référentiel et la mettre à jour avec les données du référentiel source. Utilisez OVERWRITE pour supprimer les anciennes données et tables de base de données, et recréer de nouvelles tables et données à leur place.
12. Si vous voulez que le nouveau référentiel soit utilisé par votre instance de DB2 Alphablox, entrez Y à l'invite **Update DB2 Alphablox** pour utiliser le nouveau référentiel.
13. Pour mettre à jour les propriétés nécessaires pour l'instance DB2 Alphablox, sélectionnez ALL à l'invite **Update DB2 Alphablox Properties**. ALL indique la conversion de toutes les propriétés du serveur ; SPECIFIC indique la conversion des propriétés du serveur spécifiques à la machine locale uniquement (et non des propriétés groupées) ; GLOBAL indique la conversion des propriétés partagées par la grappe uniquement (et non les entrées de la machine locale). NONE indique la modification d'aucune propriété dans l'instance DB2 Alphablox.
14. Si tous les éléments sont corrects, sélectionnez **Continuer** en entrant le numéro **1** et en appuyant sur la touche Entrée.
15. A la fin de la conversion, vous revenez au menu principal. Entrez **8** pour quitter l'utilitaire de conversion du registre.

Conversion d'une base de données en système de fichiers

Avant de convertir un référentiel d'une base de données en un système de fichiers, vous devez disposer des informations suivantes :

- les informations de connexion au répertoire de la base de données,
- le chemin complet vers l'emplacement voulu pour le référentiel du système de fichiers (par exemple, d:\alphablox\Référentiel)

La procédure suivante décrit les étapes nécessaires pour convertir un référentiel DB2 Alphablox de base de données en système de fichiers.

1. Arrêtez DB2 Alphablox.
2. Démarrez l'utilitaire de conversion de registre (pour plus de détails, voir «Démarrage de l'utilitaire de conversion du registre», à la page 121).
3. Sélectionnez l'option 3, **Convert one repository to another**, en entrant le numéro **3** et en appuyant sur la touche Entrée.
4. Sélectionnez l'option 2, **Convert database to file**, en entrant le numéro **2** et en appuyant sur la touche Entrée.
5. Sélectionnez la base de données qui correspond à votre serveur de base de données. Par exemple, si vous utilisez Oracle 8.1.7, entrez **2**.
6. Entrez les informations de configuration pour votre base de données lorsque vous y êtes invité.
7. Si tous les éléments sont corrects, sélectionnez **Continuer** en entrant le numéro **1** et en appuyant sur la touche Entrée.

L'utilitaire de conversion du registre tentera de se connecter à la base de données.

8. Entrez le répertoire pour la racine du référentiel de destination, puis appuyez sur la touche Entrée.
9. Si tous les éléments sont corrects, sélectionnez **Continuer** en entrant le numéro **1** et en appuyant sur la touche Entrée.
10. Entrez COPY ou MOVE. COPY conserve l'ancien référentiel à sa place et en crée une copie dans le référentiel de destination ; MOVE supprime l'ancien référentiel et en crée un nouveau dans le référentiel de destination.
11. Entrez NEW, UPDATE ou OVERWRITE. Utilisez NEW pour créer un référentiel s'il n'en existe pas un déjà. Utilisez UPDATE pour conserver la structure existante du référentiel et la mettre à jour avec les données du référentiel source. Utilisez OVERWRITE pour supprimer les anciennes données et structures et recréer les nouvelles structures et données à leur place.
12. Si vous voulez que le nouveau référentiel soit utilisé par votre instance de DB2 Alphablox, entrez Y à l'invite **Update DB2 Alphablox to use the New Repository**.
13. Pour mettre à jour les propriétés nécessaires pour l'instance DB2 Alphablox, sélectionnez ALL à l'invite **Update DB2 Alphablox Properties**. ALL indique la conversion de toutes les propriétés du serveur ; SPECIFIC indique la conversion des propriétés du serveur spécifiques à la machine locale uniquement (et non des propriétés groupées) ; GLOBAL indique la conversion des propriétés partagées par la grappe uniquement (et non les entrées de la machine locale). NONE indique la modification d'aucune propriété dans l'instance DB2 Alphablox.
14. Si tous les éléments sont corrects, sélectionnez **Continuer** en entrant le numéro **1** et en appuyant sur la touche Entrée.
15. A la fin de la conversion, vous revenez au menu principal. Entrez **8** pour quitter l'utilitaire de conversion du registre.

Configuration d'une instance afin qu'elle utilise un référentiel existant

Avant de configurer une instance de DB2 Alphablox afin qu'elle utilise un référentiel existant, rassemblez les informations de connexion pour le référentiel de la base de données auquel vous voulez vous connecter.

La procédure suivante décrit les étapes nécessaires pour configurer une instance de DB2 Alphablox afin qu'elle utilise un référentiel DB2 Alphablox existant.

1. Arrêtez DB2 Alphablox.
2. Démarrez l'utilitaire de conversion de registre (pour plus de détails, voir «Démarrage de l'utilitaire de conversion du registre», à la page 121).
3. Sélectionnez l'option 6, **Change DB2 Alphablox to use a different repository**, en entrant le numéro **6** et en appuyant sur la touche Entrée.
4. Sélectionnez l'option 2, **Database repository is the target**, en entrant le numéro **2** et en appuyant sur la touche Entrée.
5. Sélectionnez la base de données qui correspond à votre serveur de base de données. Par exemple, si vous utilisez Oracle 8.1.7, entrez **2**.
6. Entrez les informations de configuration pour votre base de données lorsque vous y êtes invité.
7. Si tous les éléments sont corrects, sélectionnez **Continuer** en entrant le numéro **1** et en appuyant sur la touche Entrée.

L'utilitaire de conversion du registre tentera de se connecter à la base de données, puis mettra à jour l'instance de DB2 Alphablox afin qu'elle utilise le référentiel spécifié.

8. Entrez 8 pour quitter l'utilitaire de conversion du registre.

Syntaxe de ligne de commande

Dans la plupart des cas, vous pouvez utiliser l'utilitaire de conversion du registre en mode interactif, comme décrit dans les sections précédentes. Vous pouvez, toutefois, utiliser des options de ligne de commande pour spécifier différents fichiers DDL de base de données ou pour exécuter l'utilitaire de conversion en tant que partie d'un script automatisé. La syntaxe de base pour l'utilitaire de registre est la suivante :

```
java -cp [chemin_de_classe] com.alphablox.util.convert.Convert opération
      destination [source] [arguments]
```

où :

<i>chemin_de_classe</i>	est le chemin de la classe Java pour l'instance de DB2 Alphablox. Recherchez dans les fichiers de démarrage de DB2 Alphablox (par exemple, <code>AnalysisServer.bat</code> sous les plateformes Windows et <code>AnalysisServer.sh</code> sous les plateformes Linux et UNIX) un exemple de <i>chemin_de_classe</i> .
<i>opération</i>	est l'opération d'utilitaire de conversion du registre à exécuter. Pour voir une description des opérations disponibles, reportez-vous à la section «Descriptions des opérations», à la page 126.
<i>destination</i>	est un chemin relatif ou absolu vers un fichier de propriétés décrivant le référentiel de destination. Pour voir des modèles de fichiers de destination, reportez-vous à la section «Modèles de fichiers de propriétés source et de destination», à la page 128.
source	est un chemin relatif ou absolu vers un fichier de propriétés décrivant le référentiel source. Vous devez spécifier l'argument <i>source</i> pour les opérations de copie et de déplacement. Pour voir des modèles de fichiers de destination, reportez-vous à la section «Modèles de fichiers de propriétés source et de destination», à la page 128.
arguments	est un ou plusieurs des arguments spécifiés dans la table dans «Arguments», à la page 127.

Si aucune *opération*, *destination*, *source* ou aucun *argument* n'est spécifié, l'utilitaire de conversion du registre s'exécute en mode interactif.

Descriptions des opérations : Le tableau suivant décrit les opérations disponibles pour l'utilitaire de conversion du registre.

Opération	Description
HELP	Affiche l'aide de ligne de commande.
CHANGE	Change le référentiel actif en référentiel de destination.
COPY	Conserve le référentiel source à sa place et en crée une copie dans le référentiel de destination.
DELETE	Supprime de façon permanente le référentiel de destination.

MOVE	Supprime le référentiel source et crée un référentiel dans le référentiel de destination.
NEW CREATE	Crée un référentiel dans le référentiel de destination et le remplit avec les valeurs du référentiel par défaut.
VERIFY	Exécute des opérations de validation sur le référentiel de destination et rapporte les résultats.

Arguments : Le tableau suivant décrit les arguments disponibles pour l'utilitaire de conversion du registre.

Argument	Description
DEBUG	Fournit des informations de débogage supplémentaires pour aider au diagnostic des problèmes.
LOG : <i>fichier</i>	Spécifie le nom du fichier journal de l'utilitaire de conversion, dans lequel sont consignées toutes les activités de l'utilitaire. Le nom par défaut du <i>fichier</i> est <i>repositoryconvert.log</i> .
SERVER : <i>nom_instance</i>	Spécifie le nom d'instance de l'instance de DB2 Alphablox. L'argument SERVER est requis.
OVERWRITE	Utilisé avec une opération NEW, MOVE, COPY ou CHANGE, l'argument OVERWRITE supprime les anciennes données et structures dans le référentiel de destination et recrée les nouvelles structures et données à leur place. Sans cet argument, l'utilitaire de conversion du registre s'arrête s'il détecte un référentiel existant à la destination.
PROPS : <i>option</i>	Utilisé avec l'argument SERVER : <i>nom_instance</i> , spécifie les propriétés de serveur à mettre à jour sur le référentiel de destination. Les valeurs pour <i>option</i> sont les suivantes : <ul style="list-style-type: none"> • ALL : indique la conversion de toutes les propriétés de serveur. • GLOBAL : indique la conversion des propriétés partagées par la grappe uniquement (et non les entrées de la machine locale). • SPECIFIC : indique la conversion des propriétés de serveur spécifiques à la machine locale uniquement (et non les propriétés groupées).
UPDATE	Spécifie qu'une opération MOVE ou COPY doit mettre à jour le référentiel de destination avec des informations du référentiel source plutôt que de remplacer le contenu du référentiel de destination.
DDL : <i>fichier</i>	Remplace le fichier de schéma DDL par défaut utilisé pour la création des tables, des index et du contenu initial du référentiel de base de données.

Important : Soyez extrêmement prudents lorsque vous utilisez cet argument.

USEDEST

Lorsqu'il est spécifié, cet argument modifie les propriétés de serveur pour utiliser le référentiel de destination. Si l'argument n'est pas spécifié, les propriétés de référentiel du serveur ne sont pas modifiées.

Modèles de fichiers de propriétés source et de destination : Cette section montre le contenu des modèles de fichiers source et de destination utilisés avec l'utilitaire de conversion du registre. Les fichiers de propriétés spécifient le type de référentiel, les informations de connexion du référentiel, l'emplacement des fichiers de propriétés du référentiel sur l'ordinateur exécutant DB2 Alphablox, ainsi que le nom du fichier DDL.

L'exemple suivant montre un fichier de propriétés pour un référentiel se trouvant dans une base de données Oracle sur un serveur nommé *oracle817* :

```
RepositoryTarget=JDBCTarget
java.naming.factory.initial=com.alphablox.jndisp.AlphabloxContextFactory
java.naming.provider.url.server=oracle817
java.naming.provider.url.port=1521
java.naming.provider.url.sid=orcl817
database_driver=oracle.jdbc.driver.OracleDriver
fileroot=C:\alphablox\analytics\repository\servers\
commandfile=oracle.dmlsql
user=user
password=password
```

L'exemple suivant montre un fichier de propriétés pour un référentiel se trouvant dans un système de fichiers :

```
RepositoryTarget=ABXTarget
java.naming.factory.initial=com.alphablox.jndisp.AlphabloxContextFactory
java.naming.provider.url=C:\alphablox\analytics\repository\
fileroot=C:\alphablox\analytics\repository\servers\
```

Chapitre 17. Utilisation du regroupement de connexions

Le présent chapitre décrit comment configurer DB2 Alphablox pour utiliser le regroupement de connexions avec des sources de données relationnelles et multidimensionnelles.

Regroupement de connexions - Présentation

Dès qu'une application Web doit interagir avec une base de données relationnelle ou multidimensionnelle, elle doit d'abord s'y connecter. Chacune de ces connexions connaît une surcharge, lors de l'utilisation des ressources pour établir la connexion, la gérer et la libérer lorsqu'elle n'est plus requise. Dans des applications Web, les interactions utilisateur avec les bases de données sont généralement de courte durée et souvent, les opérations de connexion à une base de données et de déconnexion de cette dernière sont plus longues que le traitement des requêtes.

Pour traiter plus efficacement ces interactions de base de données, les serveurs d'application et les bases de données incluent une fonction de regroupement de connexions. Les pools de connexions sont de groupes préétablis de connexions qui peuvent être partagées avec des applications, permettant les interactions de base de données nécessaires mais n'utilisant pas autant de ressources et de temps que des connexions individuelles.

S'il est disponible et correctement configuré, DB2 Alphablox peut utiliser des regroupements de connexion pour améliorer les performances de vos applications. Le regroupement de connexions avec DB2 Alphablox peut être utilisé avec des serveurs d'applications (IBM WebSphere et BEA WebLogic) pour le regroupement de connexions de base de données relationnelles ou avec le support de regroupement de connexions dans les services de déploiement IBM DB2 OLAP Server ou Hyperion Essbase.

Regroupement de connexions de bean géré par message

Le regroupement de connexions de base de données multidimensionnelle n'est pas pris en charge à l'aide de connexions JDBC standard qui fonctionnent uniquement avec des sources de données relationnelles. Pour DB2 OLAP Server ou Hyperion Essbase, DB2 Alphablox peut utiliser les implémentations du regroupement de connexions disponible dans les services de déploiement IBM DB2 OLAP Server ou Hyperion Essbase.

Regroupement de connexions DB2 OLAP Server et Hyperion Essbase

Les services de déploiement IBM DB2 OLAP Server ou Hyperion Essbase peuvent être configurés pour la prise en charge du regroupement de connexions. Chaque pool de connexions défini indique le nom d'utilisateur ou le mot de passe DB2 OLAP Server ou Essbase à utiliser. L'accès d'un pool de connexions peut être défini pour tous les utilisateurs ("Autoriser tous les utilisateurs") ou pour une liste d'utilisateurs et de groupes définis. Si une source de données DB2 Alphablox définit l'adaptateur des services de déploiement IBM DB2 OLAP Server ou Hyperion Essbase Deployment et que les pools de connexions sont correctement configurés dans les services de déploiement IBM DB2 OLAP Server ou Hyperion Essbase, DB2 Alphablox peut utiliser ces pools de connexions. Pour plus de détails

sur la configuration des pools de connexions Essbase, voir la documentation relative aux services de déploiement IBM DB2 OLAP Server ou Hyperion Essbase.

Microsoft Analysis Services et regroupement de connexions

L'utilisation du regroupement de connexions dans les implémentations Microsoft Analysis peut être bénéfique à l'amélioration des performances des applications analytiques. Par défaut, le regroupement de connexions est désactivé.

Lorsque le pool de connexions est activé et que la méthode *disconnect()* du DataBlox est appelée, le DataBlox sera déconnectée mais la connexion ADO ne sera pas fermée, elle sera placée dans un pool de connexions et pourra alors être réutilisée. Lorsqu'une connexion DataBlox ultérieure demande une connexion qui correspond à un élément du pool, la connexion est supprimée du pool et attribuée au DataBlox. Sinon, une nouvelle connexion est créée. Une connexion placée en pool est indexée par rapport au nom d'utilisateur de la source de données, au mot de passe, à la chaîne du fournisseur, au catalogue (nom de la base de données Microsoft Analysis Services) et au schéma (qui a généralement la valeur null car il n'est pas utilisé par Microsoft Analysis Services). Ces cinq propriétés doivent correspondre afin qu'un composant DataBlox puisse utiliser une des connexions du pool. La modification de la chaîne de fournisseur dans différentes sources de données génère différentes connexions unique dans le pool. Chacune de ces connexions uniques est une clé de pool de connexions. Par exemple, s'il existe deux connexions utilisateur1 vers basededonnées1 utilisant la chaîne de fournisseur par défaut de "MSOLAP" et une autre connexion du même utilisateur vers la même base de données avec une chaîne de fournisseur "MSOLAP;Default Isolation Mode=1;Execution Location=3;Client Cache Size=0;", il existera trois connexions utilisant deux différentes clés de pool de connexions.

Activation du pool de connexions

Par défaut, le pool de connexions MSOLAP est désactivé. Pour l'activer, vous devez attribuer la valeur true à la propriété de serveur MSOLAPEnableConnectionPool.

Pour activer le pool de connexions, procédez comme suit :

1. Ouvrez une console telnet sur votre serveur DB2 Alphablox (ou utilisez la fenêtre Admin Console, disponible dans les pages d'administration DB2 Alphablox).
2. Définissez la propriété serveur MSOLAPEnableConnectionPool en saisissant "*set MSOLAPEnableConnectionPool true*" et appuyez sur la touche Entrée. Le message suivant s'affiche : "Server property 'Enable pooling of the MSOLAP connection pool' set to true."
3. Entrez "save" puis appuyez sur la touche Entrée. Le message "Server properties saved" s'affiche.

Remarque : Si vous n'enregistrez pas la propriété, la propriété du serveur s'applique uniquement lors de la session serveur en cours.

Pour désactiver ce pool de connexions, effectuez à nouveau la procédure précédente mais attribuez la valeur false à la propriété MSOLAPEnableConnectionPool.

Utilisation du pool de connexions

Lors de l'utilisation du pool de connexions DB2 Alphablox Microsoft Analysis Services, vous devez vous assurer que dans tous les DataBlox, la valeur true est attribuée à la propriété autoDisconnect. Si ce n'est pas le cas, les connexions ne seront pas renvoyées au pool disponible avant la fin de la session utilisateur. De plus, le comportement de la méthode clearClientCache() est différent que lorsqu'il n'existait pas de regroupement de connexions. Si la méthode clearClientCache() est appelée lorsque le regroupement de connexions est activé, elle efface du pool disponible toutes les connexions dont la clé de pool de connexions correspond à la connexion du DataBlox. Vous pouvez également effacer l'intégralité du pool de connexions à l'aide de la méthode ODBOBridgeJavaAPI.clearPool(). Cette méthode efface toutes les connexions ADO du pool disponible.

Restriction du pool de connexions

Pour les environnements avec un fort accès concurrent et comportant peu de connexions regroupées uniques (noms d'utilisateur), vous pouvez améliorer la consommation de mémoire et les performances en restreignant la taille du pool. La propriété serveur MSOLAPConnectionPoolLimit a été ajoutée. Elle limite le nombre de connexions MSOLAP actives que DB2 Alphablox a ouvert par clé de pool de connexions. Cet élément inclut les connexions qui se trouvent dans le pool. Cette fonction doit être utilisée avec vigilance. Elle est conçue uniquement pour l'environnement unique décrit ici. La propriété MSOLAPConnectionPoolLimit limite le nombre de connexions par clé de pool de connexions. Utilisez la propriété serveur MSOLAPConnectionPoolLimit afin de définir le nombre maximal de connexions.

Par exemple, l'attribution de la valeur 4 à MSOLAPConnectionPoolLimit définit le nombre maximal de connexions MSOLAP ouvertes par clé de pool de connexions à 4. Dans cet exemple, supposons que les connexions à MSAS sont identiques à l'exception de la propriété de nom d'utilisateur et que vous disposez uniquement de trois noms d'utilisateur (Cadre, Directeur et Admin) qui peuvent se connecter. Si un composant DataBlox tente de se connecter à l'aide de l'ID utilisateur Admin et qu'il s'agit de la cinquième connexion, il doit attendre qu'une connexion pour ce nom d'utilisateur se libère (ou se ferme) et soit placée à nouveau dans le pool, ainsi le nombre de connexions ouvertes est inférieur à la limite.

Optimisation du pool de connexions

A l'aide du regroupement de connexions DB2 Alphablox Microsoft Analysis Services, vous pouvez ajuster le pool afin d'optimiser les performances et l'utilisation de la mémoire. Plusieurs facteurs doivent être pris en compte

Cache côté client de Microsoft Analysis Services : Un concept qui n'est généralement pas reconnu dans la connectivité est l'importance du cache côté client Microsoft Analysis Services. MSAS peut placer en cache plusieurs éléments sur l'hôte DB2 Alphablox. Par exemple, les mesures calculées seront toujours calculées et placées en cache dans la couche ADO sur l'hôte Alphablox. Si les informations sont réutilisées de manière continue entre les différentes sessions Alphablox, il est préférable d'avoir une connexion ADO ouverte en permanence qui contient ces informations.

Remarques sur le regroupement de connexions : Le regroupement de connexions est utile lorsque vous avez un nombre limité de clés de pool de connexion utilisées lors du processus de connexion (nombre restreint de noms utilisateur distincts) ou lorsque les sessions DB2 Alphablox ont plusieurs connexions Blox qui utilisent la même clé de pool de connexions.

N'utilisez pas de regroupement de connexions si tous les utilisateurs (pour lesquels des clés de pool de connexions uniques sont requises) sont uniques et que vous avez un seul DataBlox par clé de pool de connexions ou si vous avez un grand nombre de clés de pools de connexions et que vous ne pouvez pas les effacer à la fin des sessions utilisateur.

Utilisation de la propriété autoDisconnect : Vous devez presque toujours utiliser la propriété DataBlox autoDisconnect lors de l'utilisation du regroupement de connexions. Sinon, les connexions ADO ne seront pas renvoyées au pool pour réutilisation.

Utilisation de la méthode clearClientCache() : Si vous utilisez un grand nombre de clés de pool de connexions (autrement dit, chaque connexion à Microsoft Analysis Services est un utilisateur ayant un nom spécifique), vous pouvez appeler la méthode DataBlox clearClientCache() sur chaque composant DataBlox avec une clé de pool de connexions unique lorsque l'utilisateur a terminé la session. Cette action est généralement effectuée via une page de déconnexion appelée lorsque l'utilisateur se déconnecte de manière explicite ou ferme sa fenêtre de navigateur.

Utilisation de la méthode clearPool() : Dans la plupart des cas, il peut être judicieux de vider régulièrement le pool de connexions. Un simple servlet temporisateur qui vide le pool tous les soirs est une bonne solution pour la plupart des environnements.

Contrôle et gestion du pool de connexions : Les méthodes statiques suivantes sont disponibles sur la classe ODBOBridgeJava pour contrôler et gérer le pool de connexions :

ODBOBridgeJavaAPI.poolSize()

Renvoie le nombre de connexions dans le pool disponible

ODBOBridgeJavaAPI.clearPool()

Déconnecte les connexions du pool disponible

ODBOBridgeJavaAPI.getMaximumNumberOfConnections()

Extrait le nombre maximal de connexions MSAS pouvant être ouvertes

ODBOBridgeJavaAPI.getNumberOfOpenConnections()

Extrait le nombre en cours de connexions MSAS ouvertes

Le nombre de connexions ADO est égal au nombre de connexions dans le pool disponible plus le nombre de connexions utilisées.

L'exemple de scriptlet JSP utilise les méthode précédentes pour gérer et contrôler un pool de connexions :

```
<% ODBOBridgeJavaAPI.setMaximumNumberOfConnections(15);
   int count = ODBOBridgeJavaAPI.poolSize();
   out.write("Number of connections in the pool: "+count+"<br>");
   count = ODBOBridgeJavaAPI.getNumberOfOpenConnections();
   out.write("Number of connections active: "+count+"<br>");
   count = ODBOBridgeJavaAPI.getMaximumNumberOfConnections();
   out.write("Connection limit: "+count+"<br>");
%>
```

Regroupement de connexions de base de données relationnelle

Un pool de connexions de base de données relationnelle est un groupe nommé de connexions JDBC identiques à une base de données relationnelle créées lorsqu'une pool de connexions est enregistré sur un serveur d'applications. Une application Web peut "emprunter" une connexion dans le pool, utiliser cette dernière lors d'une interaction avec la base de données puis renvoyer la connexion au pool en fermant la connexion.

Le tableau suivant met en évidence les avantages de l'utilisation du regroupement de connexions :

Avantage	Description
Amélioration du temps de réponse	Dès qu'une ressource tente d'accéder à une base de données, elle doit se connecter à cette dernière. Lors du démarrage d'un pool de connexions, elle crée un nombre défini de connexions à la base de données physique. Ainsi, il n'est plus nécessaire de créer des connexions de base de données pour chaque application. En supprimant les tâches de création, de gestion et de libération des connexions, les pools de connexions peuvent améliorer de manière significative le temps de réponse pour les extractions d'ensembles de résultats.
Abstraction à partir des bases de données sous-jacentes	Le regroupement de connexions offre une couche d'abstraction provenant des bases de données sous-jacentes. Il est ainsi plus facile de changer de base de données d'application sans prendre en compte les exceptions SQL des différents fournisseurs. A la place, il vous suffit d'utiliser l'exception de regroupement de connexions du serveur d'applications.
Gestion des connexions simultanées	Si vous avez un nombre limité de connexions de base de données suite à des restrictions de licence, le regroupement de connexions peut être utilisé pour gérer les connexions de base de données simultanées.

Utilisation de DB2 Alphablox pour le regroupement de connexion de base de données relationnelle

Si vous utilisez des serveurs d'applications WebSphere et WebLogic avec DB2 Alphablox, les éléments suivants peuvent utiliser le regroupement de connexions :

- Sources de données DB2 Alphablox
- Référentiel DB2 Alphablox sous forme de base de données
- Création de cube de base de données relationnelle
- ReportBlox
- CommentsBlox
- Beans de connexion JDBC

L'ensemble des fonctions DB2 Alphablox répertoriées ci-dessus, hormis les sources de données DB2 Alphablox et le référentiel DB2 Alphablox sous forme de base de données relationnelle, tireront automatiquement parti du regroupement de connexions de base de données relationnelle s'il est disponible et correctement configuré.

Sources de données DB2 Alphablox et regroupement de connexions de base de données relationnelle

Pour utiliser le regroupement de connexions de base de données relationnelle avec DB2 Alphablox et les serveurs d'applications WebSphere ou WebLogic, l'option Application Server Data Source Adapter doit être sélectionnée dans les définitions de source de données DB2 Alphablox. Cette option d'adaptateur apparaît uniquement lorsque DB2 Alphablox est configuré pour utiliser les serveurs d'applications WebSphere et WebLogic pris en charge.

Remarque : Lors de l'utilisation des pools de connexions WebSphere ou WebLogic pour accéder aux sources de données, vous pouvez utiliser DB2 Alphablox avec toute base de données relationnelle prise en charge sur les serveurs d'applications.

Remarque : La configuration Apache Tomcat de DB2 Alphablox ne prend pas en charge le regroupement de connexions de base de données relationnelle.

Référentiel DB2 Alphablox et regroupement de connexions de base de données relationnelle

Pour activer ou désactiver le regroupement de connexions avec un référentiel DB2 Alphablox relationnel, procédez comme suit :

Activation

1. Sur votre serveur d'applications, créez une source de données DB2 Alphablox pour le pool de connexions défini. Pour plus de détails, voir les instructions ci-dessous.

2. Exécutez l'utilitaire de conversion du référentiel.

Windows :

```
<serverDirectory>/tools/convert/ConvertRepository.exe
```

Linux et UNIX :

```
<serverDirectory>/tools/convert/ConvertRepository.bat
```

3. Si DB2 Alphablox utilise WebSphere ou WebLogic, l'option 8 ("Configure Web Application Server Connection Pooling") est disponible. Sélectionnez cette option puis appuyez sur la touche Entrée.

4. Sélectionnez l'option 1 ("Turn On connection pooling") puis appuyez sur la touche Entrée.

5. Entrez le nom JNDI de la source de données définie dans le serveur d'applications.

Désactivation

1. Exécutez l'utilitaire de conversion du référentiel.

2. Si DB2 Alphablox est exécuté sur WebSphere ou WebLogic, l'option 8 ("Configure Web Application Server Connection Pooling") est disponible. Sélectionnez cette option puis appuyez sur la touche Entrée.

3. Sélectionnez l'option 1 ("Turn Off connection pooling").

Configuration du regroupement de connexions avec BEA WebLogic

Lors de l'utilisation du regroupement de connexions BEA WebLogic avec des sources de données relationnelle, BEA WebLogic requiert que vous créiez un utilisateur WebLogic pour chaque source de données afin d'accéder au pool de connexions défini.

Pour configurer correctement DB2 Alphablox pour une utilisation avec le pool de connexions WebLogic, le nom d'utilisateur et le mot de passe par défaut de la définition de source de données DB2 Alphablox doivent être un utilisateur WebLogic. Si vous n'utilisez pas de regroupement de connexions, il n'est pas nécessaire de créer d'utilisateur WebLogic pour accéder aux sources de données relationnelles.

Chapitre 18. Utilisation d'environnements groupés

Le présent chapitre décrit comment configurer et utiliser DB2 Alphablox dans des environnements groupés, ce qui permet d'utiliser des applications analytiques évolutives. Deux environnements groupés sont présentés : l'environnement groupé WebSphere et l'environnement groupé autonome WebLogic.

- «Présentation des environnements groupés», à la page 137
- «Environnements groupés WebSphere», à la page 137
- «Environnements groupés WebLogic», à la page 137
- «Commandes de la console de cluster», à la page 138

Présentation des environnements groupés

Pour augmenter l'évolutivité et la disponibilité, DB2 Alphablox peut s'exécuter sur un cluster incluant au moins deux serveurs. Plus le nombre de serveurs est important dans le serveur, plus le système peut prendre en charge un nombre important d'utilisateurs. Un cluster propose deux fonctions principales :

- **Évolutivité** : La capacité d'un cluster n'est pas limitée à une seule machine. De nouveaux serveurs peuvent être ajoutés dynamiquement au cluster afin d'augmenter la capacité
- **Haute disponibilité** : Un cluster utilise la redondance de plusieurs serveurs afin d'empêcher des erreurs dans les clients.

Remarque : Si une configuration DB2 Alphablox est configurée pour s'exécuter dans un environnement groupé, le référentiel doit se trouver dans une base de données relationnelle. Pour obtenir plus d'informations sur le référentiel DB2 Alphablox et sur sa configuration à l'aide d'une base de données relationnelle, voir «Présentation du référentiel DB2 Alphablox», à la page 119.

Environnements groupés WebSphere

Pour plus d'informations sur la configuration et l'installation de DB2 Alphablox dans un environnement groupé WebSphere, voir la section *Utilisation de DB2 Alphablox dans des environnements WebSphere groupés* du document *Guide d'installation*. Pour plus de détails sur la configuration et l'utilisation des clusters WebSphere, voir la documentation du serveur WebSphere.

Environnements groupés WebLogic

Un cluster WebLogic est un groupe de serveurs WebLogic qui fonctionnent conjointement afin de fournir évolutivité et disponibilité, qui ne sont pas disponibles avec un environnement comportant un seul serveur. Pour les clients, un cluster ressemble à un serveur unique alors qu'il s'agit en fait d'un groupe de serveurs. Ces clusters WebLogic peuvent être plusieurs instances sur la même machine physique ou des serveurs WebLogic installés sur plusieurs machines physiques.

Chaque hôte DB2 Alphablox communique avec tous les autres noeuds du cluster afin que chaque noeud indique aux autres les changements effectués dans le référentiel. Tous les changements effectués dans le référentiel (par exemple, un nouvel utilisateur) sont stockés directement dans le référentiel et chaque noeud peut lire et modifier le référentiel.

Configuration et installation de DB2 Alphablox dans des environnements groupés WebLogic

Pour plus d'informations sur la configuration et l'installation de DB2 Alphablox dans un environnement groupé WebLogic, voir la section *Utilisation de DB2 Alphablox dans des environnements WebLogic groupés* du document *Guide d'installation*. Pour plus de détails sur la configuration et l'utilisation de clusters WebLogic, consultez la documentation du serveur WebLogic.

Création d'applications dans des environnements groupés WebLogic

Pour obtenir des instructions sur la définition de nouvelles applications DB2 Alphablox dans un environnement groupé WebLogic, voir «Définition d'une application lors de l'utilisation de clusters WebLogic», à la page 37.

Utilisation de clusters verticaux WebLogic

Une configuration de clusters verticaux permet d'améliorer l'évolutivité des applications DB2 Alphablox qui fonctionnent avec Microsoft Analysis Services ou d'héberger plusieurs instances de serveur sur une seule machine pour une meilleure efficacité et une réduction des coûts.

Pour les utilisateurs de Microsoft Analysis Services, les clusters verticaux WebLogic permettent aux applications DB2 Alphablox d'ignorer la limite du processus Microsoft Analysis Services 2000 de 2 Go sur les systèmes d'exploitation Windows en exécutant plusieurs instances de DB2 Alphablox sur une seule machine. Ainsi, l'évolutivité de Microsoft Analysis Services est améliorée.

Les clusters verticaux peuvent également être utilisés sur des machines uniques et puissantes sur lesquelles vous pouvez optimiser vos investissements informatiques et réduire les incidents de maintenance liés à l'utilisation de plusieurs machines.

Pour plus d'informations sur la configuration et l'installation de DB2 Alphablox dans un environnement groupé WebLogic, voir la section *Utilisation de DB2 Alphablox dans des environnements WebLogic groupés* du document *Guide d'installation*. Pour plus de détails sur la configuration et l'utilisation de clusters WebLogic, consultez la documentation du serveur WebLogic.

Commandes de la console de cluster

En plus de l'interface utilisateur d'administration, vous pouvez également utiliser la console DB2 Alphablox pour entrer des informations sur la page **Cluster Options**. Pour afficher les paramètres de propriété du cluster en cours, entrez la chaîne suivante dans une fenêtre de console DB2 Alphablox :

```
get service cluster
```

La sortie d'écran sur la console est similaire au texte suivant :

```
get service cluster
Service Cluster Manager Properties:
```



```

IsClustered ..... false
  (Is clustering enabled [true|false])
LeadHost .....
  (Name or IP address of the lead AAS host in the cluster)
PortNum ..... 7855
  (The port number on which the lead host listens for cluster messages)
MaxHosts ..... 10
  (The maximum number of AAS hosts in the cluster)
StartupTime ..... 60
  (Time, in seconds, each normal node waits while connecting to
  the cluster)
LiveIsClustered .... false
  (Clustering is currently enabled)
LiveLeadHost .....
  (Current name or IP address of the lead AAS host.)
LivePortNum ..... 7855
  (Current port number the lead host is listening on.)
LiveMaxHosts ..... 10
  (Current maximum number of AAS hosts allowed in the cluster)
LiveStartupTime .... 60
  (Current time each normal node waits while connecting to the
  lead host)

```

Le tableau suivant répertorie les commandes de la console de cluster et inclut une description de chaque commande :

Syntaxe de la commande	Description
cluster shutdown	Arrête toutes les instances du cluster DB2 Alphablox. La commande CLUSTER SHUTDOWN peut être exécutée à partir de la console de tout hôte du cluster.
get service cluster	Répertorie les paramètres en cours des propriétés du cluster. Les propriétés commençant par "Live" (par exemple, LiveIsClustered) renvoient les valeurs réelles utilisées par le gestionnaires de clusters. Ces valeurs ne peuvent pas être configurées par l'utilisateur. Les autres propriétés peuvent être définies à l'aide d'une commande SET SERVICE CLUSTER, comme il est décrit ci-dessous.
set service cluster <propriété> <valeur>	Change la valeur de la propriété cluster en fonction de la valeur indiquée. Les valeurs valides de <i>propriété</i> sont les suivantes : IsClustered LeadHost PortNum MaxHosts StartupTime Les nouveaux paramètres sont pris en compte lorsque le gestionnaire de clusters redémarre (généralement, au démarrage de DB2 Alphablox).
set service cluster IsClustered <True False>	Définit l'état du cluster, il peut être <i>activé</i> ou <i>désactivé</i> . Lorsque la valeur est true, la mise en clusters est activée. Lorsque la valeur est false, la mise en clusters est désactivée.

Syntaxe de la commande	Description
set service cluster LeadHost <hôte_principal>	Définit le nom d'hôte principal ou l'adresse IP (<i>hôte_principal</i>). Le nom ou l'adresse IP doit être résolu via le réseau sur la machine sur laquelle l'hôte principal est installé.
set service cluster PortNum <numéro_port>	Définit le numéro de port (<i>numéro_port</i>) dans lequel l'hôte principal écoute les communications de cluster. Ce port doit être disponible sur l'hôte principal ou le démarrage du cluster n'aboutit pas.
set service cluster MaxHosts <nombre_maximal_hôtes>	Définit le nombre maximal d'hôtes (<i>nombre_maximal_hôtes</i>), incluant l'hôte principal, admis dans le cluster. Tout hôte normal qui tente de rejoindre le cluster une fois la valeur <i>nombre_maximal_hôtes</i> atteinte, se voit refuser l'entrée au cluster.
set service cluster StartupTime <durée_démarrage>	Définit le nombre de secondes (<i>durée_démarrage</i>) durant lesquelles un noeud normal tente de rejoindre le cluster lors du démarrage. Une fois la valeur <i>durée_démarrage</i> atteinte, l'hôte de noeud normal s'arrête automatiquement.
show hosts	Répertorie tous les hôtes actuellement connectés au cluster. Si la mise en clusters est désactivée, SHOW HOSTS ne renvoie aucun valeur.

Chapitre 19. Commandes de la console DB2 Alphablox

La gestion de DB2 Alphablox peut être effectuée au moyen de la console associée ou à partir des pages Web de l'onglet **Administration** de la page d'accueil de DB2 Alphablox. La plupart des activités d'administration (par exemple, la création et la modification d'utilisateurs et de groupes, etc.) configurées à l'aide de l'interface utilisateur de la page d'accueil de DB2 Alphablox peuvent également être configurées en utilisant les commandes de la console. Ce chapitre explique comment utiliser la console et fournit la liste des commandes disponibles sur celle-ci.

Accès à la console

Vous pouvez accéder à la console de DB2 Alphablox de deux façons :

- «Console HTML»
- «Console Telnet»

Console HTML

Pour accéder à la console depuis la page d'accueil de DB2 Alphablox, cliquez sur l'onglet **Administration**, sur le lien **General**, puis sur le lien **Launch DB2 Alphablox Console**. La page de la console HTML s'affiche.

Un administrateur peut également ouvrir la console dans une session Telnet, dans la fenêtre Console qui apparaît lorsque DB2 Alphablox est lancé depuis le menu **Démarrer** de Windows, ou bien dans la fenêtre dans laquelle est lancé DB2 Alphablox (sous Linux et UNIX).

Console Telnet

Pour accéder à la console Telnet, lancez une session Telnet sur la machine sur laquelle est exécuté DB2 Alphablox, en indiquant le port Telnet configuré sur la page d'administration de la **Console Telnet** (23 par défaut). Par exemple, si vous exécutez DB2 Alphablox sur une machine appelée *pastrami*, entrez la commande suivante à l'invite de commande :

```
telnet pastrami 23
```

Sous Windows, le processus d'installation crée un raccourci vers la console Telnet dans le menu **Démarrer, Programmes, DB2 Alphablox**, nom de l'instance (la valeur par défaut est **AlphabloxAnalytics**), **Console**.

Entrez le nom d'utilisateur et le mot de passe permettant l'authentification auprès de la console Telnet.

Pour fermer la session Telnet, entrez la commande suivante :

```
release
```

Syntaxe des commandes

La syntaxe des commandes DB2 Alphablox est la suivante :

COMMANDE objet [valeur(s)] COMMANDE objet [valeur(s)] objet [valeur(s)]

où :

- l'objet est :
 - SERVER (par défaut)
 - CONSOLE *ID console* (où *ID console* est l'ID d'une console active particulière, par exemple C1)
 - le nom d'un objet de serveur. Vous pouvez utiliser les caractères A-Z, a-z, 0-9, le trait de soulignement et des caractères spéciaux (par exemple, des caractères accentués). L'affichage du nom est dépendant des majuscules/minuscules tandis que le nom réel authentifié n'est pas dépendant des majuscules/minuscules. Les noms *Public*, *Private* et *Properties* étant réservés, ils ne peuvent pas être utilisés en tant que nom d'objet.
- la valeur qui suit qualifie l'objet

Par exemple, dans la ligne suivante, *get* est la commande, *user* est le nom d'un objet de serveur et *admin* est le nom d'un objet utilisateur particulier.

```
get user admin
```

Lorsque vous entrez une commande suivie d'un objet, un message indique la ou les valeurs requises pour cette commande. Par exemple, si vous entrez la commande suivante :

```
create data source
```

Le message suivant s'affiche :

```
La création de la source de données requiert d'autres paramètres :  
nom_source_données nom_adaptateur SERVER nom_serveur [valeur de propriété]...
```

Les noms de propriété correspondent aux noms des pages de l'interface utilisateur. Par exemple, si vous créez une source de données DB2 OLAP Server ou Hyperion Essbase, les noms des propriétés sont *application*, *database*, *username*, *password*, *maxrows*, *maxcols* et *useasuserauth*.

Abréviation des commandes

Pour saisir les commandes plus rapidement sur la console, vous pouvez utiliser des abréviations. Par exemple, l'abréviation H correspond à la commande HELP. Si plusieurs commandes commencent par la même lettre, entrez davantage de caractères afin que l'abréviation soit unique. Par exemple, les commandes START et STATISTICS commencent par STA. En ajoutant la lettre suivante, l'abréviation devient unique (STAR pour START et STAT pour STATISTICS).

Conseil : Conseils sur l'utilisation des commandes :

- La console prend en charge les abréviations pour l'ensemble de la syntaxe de la ligne de commande. Par exemple, H C permet d'obtenir de l'aide sur la commande CREATE. G U L permet d'obtenir les propriétés relatives à l'utilisateur dont le nom commence par L.
- Il n'est pas nécessaire de mettre les chaînes entre guillemets, **sauf** le nom de l'adaptateur de base de données (par exemple, "ibm db2 olap server").
- Lorsque vous utilisez des abréviations pour les propriétés générales, utilisez les premières lettres du nom de la propriété et non de sa description. Par exemple,

pour la propriété `DefaultMessageLevel` dont la description est Niveau initial du message de la console, utilisez l'abréviation `DEFAULTM`.

Liste des commandes de la console

Vous pouvez saisir la plupart des commandes sur l'écran de la Console (accessible par le lien **Launch DB2 Alphablox Console** de la page **General** de l'onglet **Administration**) ou bien sur l'écran de commandes de DB2 Alphablox. Cependant, quelques commandes sont prises en charge uniquement sur l'un ou l'autre. De plus, toutes les commandes ne sont pas prises en charge via Telnet. Pour obtenir la liste des commandes de la console de DB2 Alphablox Cube Server, voir *DB2 Alphablox Cube Server - Guide d'administration*. Pour obtenir la liste des commandes en environnement groupé (cluster), voir «Commandes de la console de cluster», à la page 138. Pour obtenir la liste des commandes spécifiques à Extensible User Manager, voir «Commande de console Telnet pour Extensible User Manager», à la page 102.

Remarque : Les commandes suivantes de la console s'appliquent uniquement à DB2 Alphablox. Par exemple, si vous supprimez une application de DB2 Alphablox sur un serveur WebSphere, elle est supprimée uniquement de DB2 Alphablox. Pour supprimer l'application du serveur WebSphere, vous devez le faire sur la console d'administration de WebSphere Application Server.

Commande et exemple	Fonction et explications
ADD objet valeur objet valeur	Ajoute le second objet au premier
add group admin user radams	Ajoute l'utilisateur nommé radams au groupe nommé admin
CREATE objet valeur	Crée un objet de type utilisateur ou groupe.
create user radams haggis	Crée un nouvel utilisateur dont le nom est radams et le mot de passe est haggis
create data source TBC ibm db2 olap server adapter server db2erver1 application demo database basic username user1 password password1 maxrows 1000 maxcols 1000 useaasuserauth false	Crée une source de données appelée <i>TBC</i> qui accède à l'application <i>TBC</i> (démo/standard) sur le serveur IBM DB2 OLAP Server appelé <i>db2server1</i> .
DELETE objet valeur	Supprime un objet de DB2 Alphablox et du référentiel DB2 Alphablox. La suppression d'une application de DB2 Alphablox ne supprime pas l'application correspondante du serveur WebSphere.
delete user radams	Supprime l'utilisateur appelé radams de DB2 Alphablox.
delete outlinecache...	Supprime une entrée de la mémoire cache de DB2 OLAP Server ou Essbase.
EXIT	Arrête DB2 Alphablox la console et ferme l'écran de celle-ci. Cela n'est possible que depuis l'écran de commandes de DB2 Alphablox, et non depuis l'écran de la console affiché via la page d'accueil de DB2 Alphablox.
ExtUserManager	Pour obtenir la liste des commandes de la console Extensible User Manager, voir «Commande de console Telnet pour Extensible User Manager», à la page 102.

Commande et exemple	Fonction et explications
GET objet valeur get get user radams	Affiche des informations sur un objet Affiche des informations sur l'objet Server (car il s'agit de l'objet par défaut) Affiche les propriétés de l'utilisateur nommé radams
HELP objet Aide help create	Affiche les informations d'aide relatives à la commande indiquée Affiche la liste complète des commandes Affiche la description et la syntaxe de la commande create
KILL objet kill 1046976892932726235	Arrête l'instance en cours de l'objet indiqué (en général une session utilisateur) Arrête la session dont l'ID est 1046976892932726235.
LOAD objet valeur load user <NomUtilisateur> load theme load license	Charge un objet dans l'espace de travail du serveur Charge l'utilisateur <NomUtilisateur> dans l'espace de travail du serveur Charge tous les thèmes disponibles dans l'espace de travail du serveur. (Notez que la commande ne prend pas en charge le chargement d'un thème seul). Charge (met à jour) le fichier de licence (license.xml), placé dans le répertoire DB2 Alphablox bin.
LOCK	Verrouille l'accès local au serveur. Pour accéder à DB2 Alphablox après la saisie de cette commande, entrez votre nom d'utilisateur et votre mot de passe. Cela n'est possible que depuis l'écran de la console DB2 Alphablox et Telnet.
MESSAGE objet valeur(s) message review 20 message review debug 20 message C2 Testing	Envoie un message système à d'autres consoles ou fichiers journaux Envoie les 20 derniers messages à la console en cours Envoie les 20 derniers messages DEBUG à la console en cours Envoie le message "Testing" à la console C2
OBJECTS objet objects objects user	Affiche des informations sur les objets administrables Affiche la hiérarchie des objets administrables Affiche des informations sur les utilisateurs
RELEASE	Libère la console de serveur distante sur laquelle a été lancée cette commande. Cela n'est possible que depuis l'écran de la console DB2 Alphablox.
REMOVE objet objet REMOVE group admin user radams	Supprime l'objet source de l'objet cible. Supprime l'utilisateur radams du groupe admin.

Commande et exemple	Fonction et explications
REPORT objet report debug	Attribue aux messages de la console en cours le niveau DEBUG, VERBOSE, INFO, SYSTEM, WARNING, ERROR ou FATAL. Attribue aux messages le niveau debug (débogage).
RESUME objet resume user	Lance le service indiqué qui était à l'état d'attente. Lance le gestionnaire d'utilisateurs qui était à l'état d'attente.
RUN objet run abc.console run <i>createUsers.txt</i>	Exécute le fichier de commandes indiqué par la cible. Exécute le fichier de commandes abc.console. Exécute le fichier <i>createUsers.txt</i> . (Ce fichier peut être un fichier de traitement par lots contenant une séquence de commandes create user destinées à créer de nouveaux utilisateurs et définir leur mot de passe initial. Voir la remarque 2 qui contient un exemple).
SAVE objet valeur Enregistrer save user radams	Enregistre les propriétés de la cible Enregistre toutes les propriétés générales (car General est l'objet par défaut). Les modifications saisies sur la console s'appliquent à la session en cours uniquement et elles sont perdues, sauf si cette commande est lancée ou si la valeur de la propriété Enregistrer lors de la fermeture de DB2 Alphablox est oui. Enregistre les propriétés de l'utilisateur radams
SET objet valeur objet valeur set user radams password castle set smtpserver mail set resolveAliasesToBaseMembers	Attribue une valeur à l'objet. Attribue le mot de passe castle à l'utilisateur radams. Attribue le serveur de messagerie nommé <i>mail</i> au serveur SMTP.

Commande et exemple	Fonction et explications
SHOW objet	Récupère des informations pour la cible.
show topics	Affiche la liste des rubriques associées à des informations.
show server	Affiche toutes les rubriques relatives aux serveurs, y compris les utilisateurs.
show user radams	Affiche des informations sur l'utilisateur appelé radams.
show theme	Affiche le nom et la description du thème.
show outlinecache	Affiche toutes les entrées de la mémoire cache DB2 OLAP Server ou Essbase.
show hosts	Affiche le nom de chaque machine du cluster dans le cas d'un environnement groupé.
show	Affiche la liste des objets. La commande SHOW affiche également la version de l'API de bibliothèque client DB2 OLAP Server ou Hyperion Essbase, si elle n'a pas encore été chargée.
START objet	Lance le service qui est à l'état arrêté.
start user	Lance le gestionnaire d'utilisateurs qui est à l'état arrêté.
STATISTICS objet valeur	Affiche les statistiques disponibles relatives à la cible.
statistics user	Affiche les statistique relatives à tous les utilisateurs.
statistics user radams	Affiche les statistiques de l'utilisateur radams.
STOP objet	Arrête un service en cours d'exécution ou en attente. Utilisez la commande START pour lancer un service qui est arrêté.
stop user	Remarque : N'utilisez pas la commande STOP pour arrêter Cube Manager sauf si tous les cubes DB2 Alphablox en cours d'exécution ont été arrêtés. Arrête le gestionnaire d'utilisateurs.
SUSPEND objet	Met un service à l'état d'attente. Par exemple, si vous mettez le service de gestionnaire d'utilisateurs en attente, aucune autre session utilisateur ne peut être instanciée. Utilisez la commande RESUME pour relancer un service qui se trouve en attente.
suspend user	Met le service de gestionnaire d'utilisateurs en attente

Commandes de console spécifiques à Essbase

DB2 Alphablox stocke les structures DB2 OLAP Server et Hyperion Essbase mises en mémoire cache afin d'améliorer les performances. Cette section décrit la syntaxe de deux commandes destinées à gérer la cache de structure et de la commande de résolution des noms d'alias en nom de membre de base. Les commandes incluses sont les suivantes :

- «Commandes RESOLVEALIASESTOBASEMEMBERS», à la page 147
- «Commande SHOW OUTLINECACHE», à la page 147

- «Commande DELETE OUTLINECACHE», à la page 148

Commandes RESOLVEALIASESTOBASEMEMBERS

Dans une base de données DB2 OLAP Server ou Essbase, vous pouvez configurer les alias en noms de membre. La propriété de serveur RESOLVEALIASESTOBASEMEMBERS peut être utilisée avec les signets qui enregistrent des requêtes avec des alias qui peuvent être résolus en nom de membre à différents moments. Par exemple, si vous créez un signet pour une requête qui référence un alias appelé Ce trimestre, le membre en cours dans la base de données DB2 OLAP Server ou Essbase auquel se réfère la requête sera différent de celui d'il y a six mois. Si vous souhaitez que vos signets récupèrent les données de ce trimestre, vous devez attribuer la valeur TRUE à la propriété RESOLVEALIASESTOBASEMEMBERS.

La syntaxe de la commande de console RESOLVEALIASESTOBASEMEMBERS est la suivante :

```
set resolvealiasestobasemembers true | false
```

La commande suivante renvoie l'état en cours de la propriété de serveur RESOLVEALIASESTOBASEMEMBERS :

```
get resolvealiasestobasemembers
```

```
ResolveAliasesToBaseMembers .... false
  (ResolveAliasesToBaseMembers)
```

Pour activer cette fonction, entrez la commande suivante sur la console :

```
set resolvealiasestobasemembers true
```

Conseil : Pour abrégier cette commande, entrez suffisamment de lettres afin d'obtenir un nom unique, comme suit :

```
set resolve true
```

Commande SHOW OUTLINECACHE

La syntaxe de la commande de console SHOW OUTLINECACHE pour les sources de données DB2 OLAP Server ou Hyperion Essbase est la suivante :

```
show outlinecache [essbasecachemanager [entry [entryname]]]
```

Cette commande permet d'afficher des informations sur le gestionnaire de cache ou les entrées qu'il gère. Par exemple, la commande suivante permet d'afficher toutes les entrées gérées par le gestionnaire de cache de DB2 OLAP Server ou Essbase :

```
show outlinecache essbasecachemanager entry
```

Le système envoie le message suivant :

```
Entry
  Entry MDB1.Financial.Current
  ....Total accesses: 7
  ....Current accessors: 1
```

Cette réponse indique que le gestionnaire de cache de DB2 OLAP Server ou Essbase gère une seule entrée appelée MDB1.Financial.Current. Le nombre d'accès à cette cache est sept (nombre cumulatif) et la cache utilise actuellement une connexion.

Commande DELETE OUTLINECACHE

La syntaxe de la commande DELETE OUTLINECACHE pour les sources de données DB2 OLAP Server ou Hyperion Essbase est la suivante :

```
delete outlinecache [essbasecachemanager [entry [entryname]]]
```

Cette commande permet de supprimer de la cache l'entrée indiquée. DB2 Alphablox supprime l'entrée uniquement si le nombre de connexions en cours (accessors) est égal à 0. Si une entrée est supprimée, toutes les ressources en mémoire associées à l'entrée sont libérées. Par exemple, la commande suivante permet de supprimer l'entrée nommée MDB1.Financial.Current de la mémoire cache :

```
delete outlinecache essbasecachemanager entry MDB1.Financial.Current
```

La réponse du système est la suivante, laquelle indique que l'entrée indiquée a été supprimée :

```
MDB1.Financial.Current  
Entrée de cache supprimée.
```

La commande suivante et la réponse correspondante confirment que l'entrée n'est plus dans la mémoire cache :

```
show outlinecache essbasecachemanager
```

```
EssbaseCache Manager  
Total entries: 0
```

Remarques relatives aux commandes de la console

Cette section décrit certains comportements particuliers et fournit quelques conseils pour l'utilisation de la console DB2 Alphablox. Les rubriques présentes sont les suivantes :

- «Affichage des propriétés générales», à la page 148
- «Niveau des messages», à la page 149
- «Exécution d'un fichier texte via la console», à la page 149
- «Messages du journal de DB2 Alphablox», à la page 150

Affichage des propriétés générales

La console DB2 Alphablox permet d'afficher les propriétés générales et les objets. La plupart des modifications des propriétés sont appliquées immédiatement, sinon un message indique qu'il est nécessaire de redémarrer DB2 Alphablox pour que les modifications soient prises en compte. Pour obtenir la liste complète des propriétés en cours de DB2 Alphablox, utilisez la commande GET.

Les liens présents au bas de la console permettent d'accéder aux informations fréquemment demandées :

- **Aide** présente la liste des commandes de DB2 Alphablox.
- **Etat** fournit des informations d'état sur DB2 Alphablox.
- **Messages** présente une vue déroulante du journal de DB2 Alphablox.
- **Sessions** présente les informations d'état sur toutes les sessions en cours de cette instance de DB2 Alphablox.
- **Utilisateurs** affiche la liste des utilisateurs (qu'ils soient connectés ou non).
- **Services** indique l'état du composant de DB2 Alphablox. Pour obtenir la liste des services, voir «Architecture DB2 Alphablox», à la page 9.
- **Paramètres** indique le paramétrage en cours de chaque propriété de DB2 Alphablox (de démarrage et avancé).

- **Afficher** présente la liste hiérarchique des rubriques de DB2 Alphablox.
- **Historique** présente l'historique des commandes lancées sur cette console.

Pour plus d'informations sur l'accès à la console, voir «Accès à la console», à la page 141.

Niveau des messages

Les messages générés par DB2 Alphablox peuvent être envoyés à toutes les consoles actives et au fichier journal actif. Les messages contiennent les informations suivantes :

- date et heure
- niveau du message
- texte du message
- source du message (service ou nom d'utilisateur)

Exemple de message courant :

```
01/9/99 12:25:33 PM [VERBOSE] Requête 4223: Traitement de la requête
'/servlet/AnalysisServer/console/consolistyle.css' [Session 41, Utilisateur radams]
```

Voici les différents niveaux de message, présentés par ordre de gravité (de la plus faible à la plus élevée) :

- **DEBUG** : informations de débogage.
- **VERBOSE** : tous les messages du système.
- **INFO** : événements système mineurs pour lesquels aucune action n'est nécessaire de la part de l'administrateur.
- **SYSTEM** : événements système normaux, tels que la création d'un utilisateur.
- **WARNING** : erreur récupérable, mais nécessitant éventuellement des recherches de la part de l'administrateur.
- **ERROR** : erreur non récupérable.
- **FATAL** : erreur entraînant l'arrêt du serveur.

Exécution d'un fichier texte via la console

La commande *run* permet d'exécuter sur la console un fichier texte contenant une série de commandes DB2 Alphablox. Par exemple, un fichier nommé *d:\CreateUser.txt* contient les lignes suivantes :

```
create user radams blue
create user sadams green
create user lplanting purple
create user klawrence yellow
create user dmessink orange
```

Vous pouvez exécuter ce fichier à l'aide de la console avec la commande *run* pour créer cinq nouveaux utilisateurs et définir leur mot de passe initial en exécutant la commande suivante sur la console :

```
run d:/CreateUser.txt
```

Remarque : Le nom du fichier doit être qualifié complet.

Messages du journal de DB2 Alphablox

Des messages sans gravité du type suivant peuvent apparaître dans le fichier journal de DB2 Alphablox (par défaut, Server.log) :

```
[VERBOSE] Requête 36: Fichier non trouvé '...\resources\basic_en_US.class'  
[VERBOSE] Requête 37: Fichier non trouvé '...\resources\basic_en_US.properties'  
[VERBOSE] Requête 38: Fichier non trouvé '...\resources\basic_en.class'  
[VERBOSE] Requête 39: Fichier non trouvé '...\resources\basic_en.properties' ...  
[VERBOSE] Requête 45: Fichier non trouvé '...\resources\metal_en_US.class'  
[VERBOSE] Requête 46: Fichier non trouvé '...\resources\metal_en_US.properties'  
[VERBOSE] Requête 47: Fichier non trouvé '...\resources\metal_en.class'  
[VERBOSE] Requête 48: Fichier non trouvé '...\resources\metal_en.properties'  
...
```

Dans cet exemple, la machine JVM tente de trouver le regroupement de ressources le plus adéquat pour le langage et l'emplacement de l'environnement local du client. Cela n'est pas indispensable au fonctionnement de DB2 Alphablox : vous pouvez ne pas en tenir compte sans risques.

Chapitre 20. Administration des applications Alphablox FastForward

Ce chapitre décrit comment gérer les applications Alphablox FastForward, y compris les tâches de configuration et d'administration de base.

Présentation générale

Alphablox FastForward est un modèle de structure d'application, préinstallé dans DB2 Alphablox, conçu pour le développement, le déploiement et le partage rapides de vues analytiques personnalisées à travers les organisations professionnelles. Prête à l'emploi, la structure FastForward offre des services d'application courants, dont la sécurité, la collaboration et la personnalisation. Les administrateurs d'application, en général des administrateurs OLAP, peuvent créer de nouvelles versions d'une application FastForward, publier des rapports en sélectionnant des modèles de rapport et en configurant des paramètres de rapport, puis déployer la nouvelle application sans se préoccuper du code. De plus, grâce à sa flexibilité et à son extensibilité, les développeurs JSP peuvent modifier ou étendre la structure d'application, ainsi qu'ajouter des modèles de rapport personnalisés que les administrateurs d'application peuvent ensuite configurer et déployer.

La structure d'application FastForward contient des fonctions souvent présentes dans les applications de création de rapports, ainsi que dans les applications analytiques, y compris :

- Exportation vers Microsoft Excel
- Génération de vues imprimables
- Enregistrement et partage faciles de vues personnelles des données
- Envoi des vues par e-mail à d'autres personnes
- Navigation facile entre les différentes vues

Rôles des utilisateurs FastForward

Les trois rôles principaux des utilisateurs Alphablox FastForward sont : administrateurs d'application, développeurs de modèles et utilisateurs. Une synergie solide entre ces trois groupes permettra d'assurer le succès des applications basées sur FastForward. Vous trouverez une brève description de ces trois rôles plus loin.

Administrateurs d'application

Les administrateurs d'application, en général des administrateurs OLAP, doivent pouvoir créer de nouvelles versions d'applications FastForward en définissant quelques paramètres, créer des rapports basés sur des modèles de rapport disponibles, puis déployer rapidement les solutions. Si le modèle de rapport existant utilisé ne répond pas aux besoins de l'utilisateur, l'administrateur d'application travaille avec les développeurs de modèles pour en créer un autre. Un administrateur d'application doit pouvoir travailler en appliquant ses connaissances des bases de données OLAP et en utilisant cette documentation (disponible également dans l'aide en ligne de l'administrateur en mode Tâches d'administrateur dans les applications FastForward).

Développeurs de modèles

Les développeurs de modèles sont en général des développeurs JSP dont le rôle principal est de créer des modèles de rapport personnalisés lorsque les modèles de rapport existants ne peuvent pas être utilisés par les administrateurs d'application pour configurer les rapports demandés. En concertation avec les administrateurs d'application et les utilisateurs, les développeurs de modèles doivent pouvoir modifier les modèles de rapport existants ou en créer d'autres pour répondre aux besoins des utilisateurs.

Les bibliothèques d'étiquettes de Blox, l'API Java côté serveur et l'API JavaScript côté client, ainsi que l'expérience en programmation Web, doivent permettre aux développeurs de modèles de créer des modèles adaptés à la plupart des besoins. Les développeurs doivent non seulement être familiarisés avec la création d'applications et de vues DB2 Alphablox, ils doivent également être familiarisés avec l'aide de l'utilisateur FastForward (disponible à partir du bouton Aide en mode utilisateur), ainsi qu'avec l'aide de l'administrateur (disponible à partir du bouton Aide en mode Tâches d'administrateur).

Utilisateurs

Les utilisateurs, en général des analystes et d'autres professionnels dans l'organisation, doivent pouvoir se connecter à une application FastForward et utiliser les rapports publiés pour analyser la problématique de l'entreprise. En fonction de l'interactivité disponible dans une application basée sur FastForward, les utilisateurs peuvent manipuler les données, explorer les hiérarchies de données, modifier les types de graphique, ajouter des commentaires, et bien plus. Une fois les vues modifiées pour qu'elles répondent à des questions particulières, les utilisateurs peuvent conserver leurs vues actuelles, en créant des rapports qu'ils enregistrent sous l'onglet Privé pour une utilisation ultérieure ou en partageant les rapports sous l'onglet Groupes avec des groupes définis d'utilisateurs de l'application.

Pour chaque rapport, les utilisateurs disposent en général de plusieurs autres options disponibles dans la barre d'outils de l'application, située au-dessus des rapports. Outre l'enregistrement des rapports pour l'analyse en ligne, l'option d'exportation vers Excel permet également aux utilisateurs d'exporter des vues vers les tableurs Microsoft Excel pour une analyse ultérieure hors ligne. Les utilisateurs peuvent alors imprimer une copie d'une vue donnée à l'aide de l'option Aperçu avant impression. De plus, s'ils le souhaitent, ils peuvent ouvrir un message e-mail contenant un lien vers la vue actuelle, ajouter des commentaires, et l'envoyer à d'autres utilisateurs de l'application.

Si les rapports nécessaires ne sont pas disponibles dans leurs applications, les utilisateurs demandent de nouveaux rapports directement aux administrateurs d'application.

Configuration système requise pour les applications FastForward

Outre la configuration spécifiée dans la section relative à la configuration système requise dans le guide d'installation, notez qu'Alphablox FastForward fonctionne avec le client DHTML uniquement.

Création d'applications Alphablox FastForward

Les étapes suivantes vous permettent de créer de nouvelles versions d'applications Alphablox FastForward lorsque vous vous connectez avec les droits administrateur :

1. Ouvrez votre navigateur Web et affichez les pages d'administration de DB2 Alphablox. Par défaut, la page d'applications s'affiche.
2. Dans la page d'applications, lancez Alphablox FastForward.
3. Dans la barre d'outils de l'application, cliquez sur le bouton Tâches d'administrateur.
4. La boîte de dialogue Application modèle s'ouvre et présente deux options, Créer et Continuer. Cliquez sur le bouton Créer.
5. Dans la boîte de dialogue Nouvelle application qui s'affiche, renseignez les zones suivantes :

Zone d'entrée	Description
Nom du contexte	Entrez un nom de contexte d'application devant être utilisé comme le nom de répertoire. Ce nom ne doit pas contenir d'espaces.
Nom d'affichage	Le nom qui s'affiche dans la page d'applications. Ce nom peut contenir des espaces.
Description	Facultatif. Entrez une courte description de l'application.
Rôle administrateur	Obligatoire. Définit la valeur AdministratorRole dans le fichier config.xml de l'application (situé dans le répertoire WEB-INF de l'application). Le paramètre par défaut est AlphabloxAdministrator.

Important : Ne pas configurer ou utiliser l'application Alphablox FastForward existante. Créer toujours une nouvelle version. Ce modèle d'application sera écrasé par chaque mise à niveau du serveur, et vos modifications seront perdues.

Modification des rôles d'administrateur

Par défaut, une application Alphablox FastForward attribue le rôle AdministratorRole à AlphabloxAdministrator. Pour modifier le rôle pouvant administrer une application, définissez-le dans la boîte de dialogue Nouvelle application ci-dessus, ou complétez les étapes suivantes :

1. Ouvrez le fichier config.xml de l'application FastForward, situé à l'emplacement suivant :
`<applicationDirectory>/WEB-INF/config.xml`
2. Dans le fichier config.xml, modifiez le rôle attribué dans la section suivante, en remplaçant AlphabloxAdministrator par le nouveau rôle désigné pour administrer cette application :
`<param> <param-name>AdministratorRole</param-name>
<param-value><![CDATA[AlphabloxAdministrator]]></param-value> </param>`

Pour en savoir plus sur la définition de rôles et la création de rôles, reportez-vous au Chapitre 10, «Définitions des rôles», à la page 59

Administration des applications FastForward

L'application FastForward Administration propose trois fonctions générales :

- Gestion des rapports publiés
- Gestion de l'apparence et de la sensation de base de l'application FastForward
- Affichage et suppression du journal d'application

Des informations sur l'administration des applications FastForward sont également disponibles en cliquant sur le bouton Aide dans la barre d'outils de l'application lorsque vous vous connectez en mode Tâches d'administrateur.

Catégories et sécurité d'accès aux rapports

Les applications FastForward proposent aux utilisateurs trois catégories d'accès de rapports : Publié, Privé et De groupe.

Rapports publiés

Les rapports publiés sont gérés à partir de l'écran d'administration comme décrit ci-dessous. Les rapports publiés, s'ils ne sont pas gérés, sont accessibles à tous les utilisateurs ayant accès à l'application FastForward. En revanche, l'administrateur peut contrôler l'accès aux rapports publiés en paramétrant la sécurité des dossiers de rapports. Si un utilisateur a accès à un dossier donné, il peut accéder à tout rapport contenu dans ce dossier. L'administrateur affecte des accès à un dossier en associant le dossier à un ou à plusieurs groupes DB2 Alphablox.

Rapports privés et de groupe

Les rapports d'accès privés et de groupe sont créés par les utilisateurs à partir des rapports précédemment publiés. Les rapports d'accès privés sont uniquement accessibles à l'utilisateur qui a créé le rapport. Les rapports d'accès de groupe sont accessibles aux utilisateurs membres des groupes d'utilisateurs DB2 Alphablox, et un utilisateur membre d'un groupe peut créer des rapports accessibles au groupe.

Agencement et commandes

Toutes les applications basées sur Alphablox FastForward ont une présentation similaire et comportent également le menu de navigation sur la gauche et la barre d'outils d'application en haut à droite. Les rapports accessibles apparaissent dans la fenêtre de rapport, située sous la barre d'outils de l'application.

Menu d'exploration

Ce menu contient les options suivantes :

Bouton	Description
Créer un dossier	Crée un dossier appelé Nouveau dossier. L'écran Modifier le dossier apparaît afin d'apporter des modifications au dossier.
Créer un rapport	Affiche l'écran Modifier le rapport pour qu'un nouveau rapport puisse être modifié. Le rapport doit être enregistré pour qu'il apparaisse dans l'arborescence des rapports sur la gauche.
Supprimer le rapport	Affiche de manière contextuelle une boîte de dialogue vous demandant de confirmer la suppression d'un rapport.
Copier	Copie le rapport avec Copie de précédant le nom du rapport. Vous pouvez alors renommer le fichier.

Gestion des rapports

Alphablox FastForward fournit une administration de base des rapports à partir de l'interface utilisateur de l'application lorsque vous vous connectez en mode Tâches d'administrateur.

Création de rapports

Pour créer un rapport, cliquez sur le bouton Créer un rapport dans la barre d'outils. Affiche l'écran Modifier le rapport pour que vous puissiez configurer les paramètres suivants :

Zone	Description
Nom	Nom du rapport tel qu'il apparaît dans l'arborescence.
Description	Cette description apparaît lorsqu'un utilisateur passe avec la souris au-dessus d'un rapport de l'arborescence et lorsque le rapport est chargé.
Modèle	La liste déroulante du modèle indique le nom de tous les modèles de rapport disponibles dans l'application. Les modèles de rapport sont stockés dans le répertoire des modèles de l'application. Suivant le modèle de rapport spécifique, d'autres paramètres devront être sélectionnés.
Enregistrer	Enregistre le rapport. Des erreurs risquent d'apparaître si les paramètres du rapport ne peuvent pas être validés.
Aperçu	Ouvre le rapport dans une nouvelle fenêtre de navigateur. Des erreurs risquent d'apparaître si des paramètres ne peuvent pas être validés.

Modification des rapports

Pour modifier un rapport, cliquez avec le bouton gauche de la souris sur un rapport existant dans l'arborescence. L'écran Modifier le rapport s'affiche.

Suppression des rapports

Pour supprimer un rapport, cliquez avec le bouton gauche de la souris sur le rapport, puis sur le bouton Supprimer le rapport dans la barre d'outils.

Déplacement des rapports

Pour déplacer un rapport :

1. Placez le pointeur de la souris sur l'icône en regard du rapport à déplacer, appuyez et maintenez enfoncé le bouton gauche de la souris.
2. Amenez le pointeur sur l'emplacement approprié dans l'arborescence et relâchez le bouton de la souris.

Gestion des dossiers

L'administration des dossiers est une tâche de base que vous pouvez réaliser dans l'interface utilisateur de l'application.

Création de dossiers

Pour créer un dossier, cliquez sur le bouton Créer un dossier dans la barre d'outils. L'écran Modifier le dossier apparaît avec les zones modifiables suivantes :

Zone	Description
Nom	Nom du rapport tel qu'il apparaît dans l'arborescence.
Description	La valeur qui apparaît lorsqu'un utilisateur passe avec la souris au-dessus d'un dossier de l'arborescence.
Groupes d'accès	Liste déroulante des groupes d'utilisateurs DB2 Alphablox disponibles. Plusieurs groupes d'utilisateurs peuvent être sélectionnés. Si aucun groupe n'est sélectionné, tous les utilisateurs peuvent accéder au rapport (et un avertissement s'affiche).
Enregistrer	Enregistre le dossier.

Modification des dossiers

Pour modifier un dossier, cliquez avec le bouton gauche de la souris sur un rapport existant dans l'arborescence. L'écran Modifier le rapport, dans lequel vous pouvez apporter des modifications s'affiche.

Suppression des dossiers

Pour supprimer un rapport, cliquez avec le bouton gauche de la souris sur le rapport, puis sur le bouton Supprimer le rapport dans la barre d'outils.

Déplacement des dossiers

Pour déplacer un dossier, procédez comme suit :

1. Placez le pointeur de la souris sur l'icône en regard du dossier à déplacer, appuyez et maintenez enfoncé le bouton gauche de la souris.
2. Faites glisser le dossier vers l'emplacement approprié dans l'arborescence et relâchez le bouton de la souris.

Gestion des propriétés de l'application

Pour gérer les propriétés de l'application, cliquez sur le bouton Paramètres généraux dans l'angle supérieur droit de la page. Les paramètres suivants peuvent être modifiés :

Zone	Description
Titre	Le titre de l'application. Ce titre s'affiche en haut de l'application de l'utilisateur ainsi qu'en haut de la page d'administration.
Logo	Présente le nom du fichier affiché sous forme de logo dans l'application de l'utilisateur.
Télécharger l'image du logo	Télécharge une nouvelle image de logo à partir d'un fichier accessible en local (format GIF ou JPEG). Notez que le téléchargement ne s'effectue pas en réalité tant que vous n'avez pas cliqué sur Enregistrer.
Thème	Liste déroulante de sélection des thèmes installés. Notez que le modèle d'application est uniquement conçu pour le thème fastforward.
Pied de page	Change l'affichage du texte en bas de la page de l'application de l'utilisateur.
E-mail de l'administrateur	Définit l'adresse e-mail utilisée pour le retour de l'application. Si cette adresse n'est pas définie, le bouton de retour apparaît désactivé sur la page de l'application de l'utilisateur.
Enregistrer	Enregistre toutes les modifications apportées aux zones de modification ci-dessus.

Utilisation du journal de l'application

La structure FastForward Application possède son propre fichier journal permettant de faire état d'informations et d'anomalies au niveau de l'application. Ce fichier journal peut être affiché et effacé en cliquant sur le bouton Afficher le fichier journal.

Annexe. Termes et concepts OLAP

Cette section contient les termes et les concepts qui sont critiques pour comprendre l'analyse multidimensionnelle (MDA) et OLAP (Online Analytical Processing). Nous commençons par les termes utilisés dans l'analyse bidimensionnelle, puis définissons leurs contreparties étendues dans l'analyse multidimensionnelle. Ensuite, nous illustrons les termes liés utilisés dans les bases de données OLAP.

Pour d'autres définitions OLAP, consultez également le glossaire de DB2 Alphabloc et le glossaire de l'organisme consacré à la technologie OLAP (OLAP Council Glossary).

- «Analyse bidimensionnelle»
- «Analyse multidimensionnelle»
- «Termes de base de données OLAP», à la page 162

Analyse bidimensionnelle

Les utilisateurs qui exécutent une analyse bidimensionnelle de données avec des outils, tels que des tableurs et des rapports sont familiarisés avec les termes suivants, répertoriés dans le tableau ci-dessous :

- Une **ligne** contient un ensemble de données liées. Le tableau contient deux lignes de données.
- Des **intitulés de ligne** apparaissent sur la gauche des valeurs de données. Les intitulés de ligne sont en bleu marine.
- Une **colonne** contient également un ensemble de données liées. Le tableau contient quatre colonnes de données.
- Des **intitulés de colonne** apparaissent au-dessus des valeurs de données. Les intitulés de colonne sont en rouge foncé.
- Un **point de données** (appelé également une **cellule**) est l'intersection d'une ligne et d'une colonne. Les points de données ont un fond gris.
- Une **valeur de données** est l'élément qui se trouve à un point de données spécifique. Les nombres 10 à 30 sont des valeurs de données.

Tableau de ventes bidimensionnel

	Trim. 1 Unités	Trim. 2 Unités	Trim. 3 Unités	Trim. 4 Unités
Coca Light	10	15	20	25
Coca	12	18	24	30

Analyse multidimensionnelle

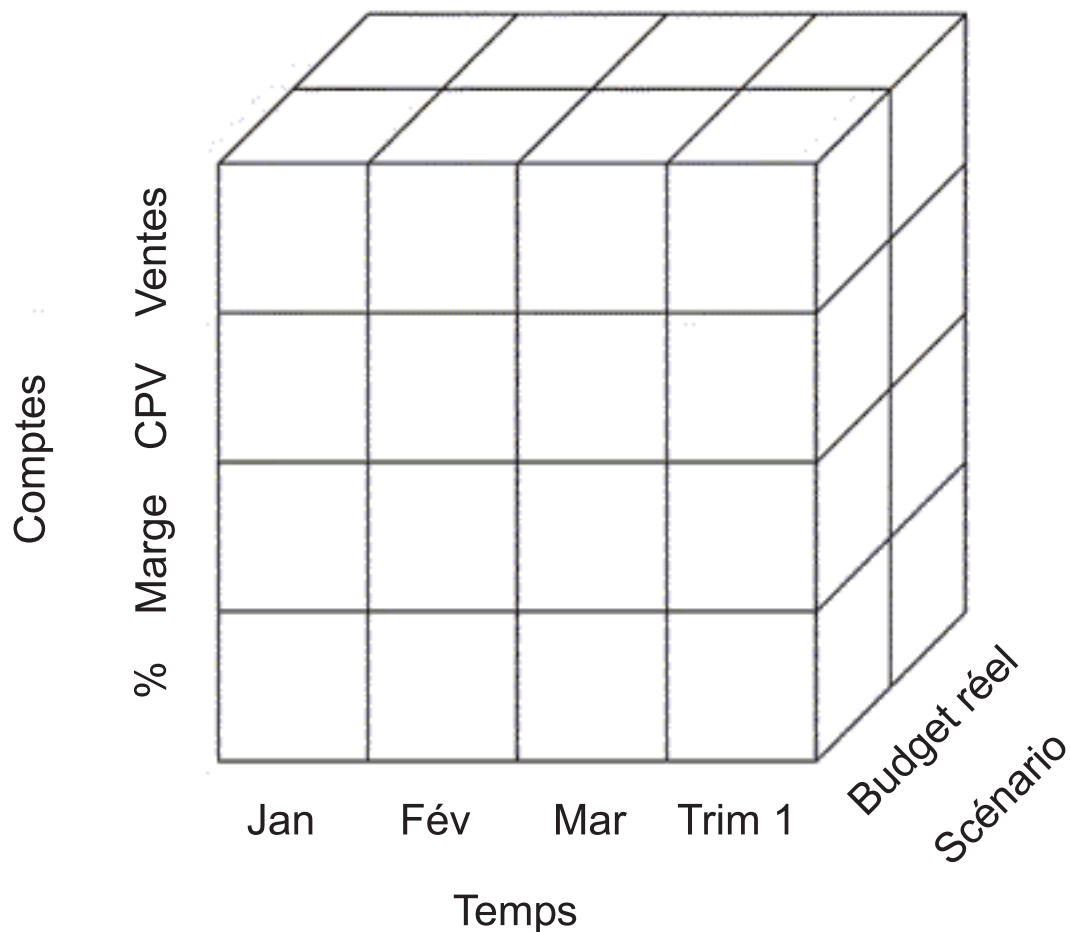
Dans l'**analyse multidimensionnelle**, ces termes et concepts prennent une autre envergure au niveau de la complexité. L'analyse multidimensionnelle implique plus de lignes et de colonnes, ainsi que d'intersections. Bien que le moyen d'affichage des données multidimensionnelles consiste souvent d'une grille bidimensionnelle, les fonctions de l'analyse multidimensionnelle impliquent des dimensions, des hiérarchies, des membres, des titres, des valeurs et des instances, ainsi que des lignes, des colonnes et des points de données.

Les définitions suivantes s'appliquent aux données multidimensionnelles et elles sont illustrées dans la matrice multidimensionnelle ci-dessous :

- Une **dimension** est un attribut structurel d'un cube de données, et est composée de membres hiérarchiques liés. Par exemple, une dimension temporelle peut être composée des membres suivants : années, trimestres, mois et semaines. Une dimension géographique peut être composée des membres suivants : régions, pays et villes.
- Un **membre de dimension** est un élément dans une dimension. Par exemple, Trimestre et Mois sont des membres de la dimension Temps.
- Une **hiérarchie de dimension** organise les membres de dimension en relations parent/enfant. Par exemple, Mois est un enfant de (appartient à) Trimestre ; Trimestre est un enfant d'Année, et ainsi de suite.
- Un **titre de dimension** est le nom d'identification de la dimension, tel que Temps ou Géographie.
- Un **titre de membre de la dimension** est le nom d'identification du membre de la dimension, tel que Mois ou Région.
- Une **valeur de membre de la dimension** est une instance d'un membre de dimension ; 1998 est une valeur d'Année du membre de la dimension.
- Un **point de données** est l'intersection de plusieurs dimensions. Les points de données dans la matrice multidimensionnelle de ventes ci-dessous ont un fond gris.
- Une **valeur de données** se trouve à un point de données. Par exemple, dans la matrice multidimensionnelle de ventes, chaque nombre dans la zone en gris est une valeur de données.

Cube de données

Cette illustration montre des données mises en forme en un cube à trois dimensions, chacune comprenant plusieurs membres :



Dimension

Membre

Comptes

Ventes, coût des ventes (CPV), Marge, Pourcentage de marge

Temps

Trimestre 1, comprenant : Janvier, Février et Mars

Scénario

Budget, Réel

Matrice multidimensionnelle des ventes

L'illustration ci-dessous montre une vue hiérarchique, multidimensionnelle de données de ventes, avec deux membres de la dimension Temps (Année et Trimestre) sur l'axe de la colonne et deux membres de la dimension Inventaire (Catégorie et Produit) sur l'axe de la ligne.

		Axe de la colonne					
		Trim. 1		Fév		Mar	
		Jan	Réel	Budget	Réel	Budget	Réel
Axe de la ligne	Comptes	Budget	Réel	Budget	Réel	Budget	Réel
	Ventes	300	350	300	325	325	325
	CPV	175	185	175	175	185	190
	Marge	125	165	125	150	140	135
	Pourcentage	42	47	42	46	43	42

Termes de base de données OLAP

Les termes suivants aident à identifier, organiser et extraire des données dans une base de données OLAP. Par exemple, lorsqu'un utilisateur explore vers le bas depuis Trim. 1, l'action OLAP consiste à extraire les "enfants" de Trim. 1 (Jan, Fév et Mar).

Modèle de hiérarchie de la dimension	Terme et définition
<pre> graph TD Année[Année] --> Trim1[Trim 1] Année --> Trim2[Trim 2] Année --> Trim3[Trim 3] Année --> Trim4[Trim 4] Trim1 --> Jan[Jan] Trim1 --> Fév[Fév] Trim1 --> Mar[Mar] Trim2 --> Avr[Avr] Trim2 --> Mai[Mai] Trim2 --> Juin[Juin] Trim3 --> Juil[Juil] Trim3 --> Aou[Aou] Trim3 --> Sep[Sep] Trim4 --> Oct[Oct] Trim4 --> Nov[Nov] Trim4 --> Déc[Déc] </pre>	<p>Racine : la position la plus élevée dans la hiérarchie. Dans l'exemple, la racine est Année.</p> <p>Descendant : un membre au moins d'une génération en dessous de Racine. Toutes les autres dimensions dans l'exemple sont des descendants de Racine. Jan est également un descendant de Trim. 1 et Année.</p> <p>Enfant : un membre exactement d'une génération en dessous d'une autre. Jan est un enfant de Trim. 1.</p> <p>Enfant : un membre exactement d'une génération au-dessus d'une autre. Trim. 1 est le parent de Jan.</p> <p>Soeur/frère : un membre de la même génération qu'un autre. Jan, Fév et Mar sont des soeurs/frères.</p> <p>Ancêtre : un membre d'une génération ou plus au-dessus d'une autre. Trim. 1 et Année sont des ancêtres de Jan.</p>

Remarque : La lettre majuscule "I" utilisée comme préfixe de ces termes (IPARENT, IENFANT, etc.) indique "inclusion". Par exemple, dans une exploration vers le haut Parent, le résultat ne montrerait que les données du parent. Dans une exploration vers le haut IPARENT, le résultat inclurait les données du parent et de l'enfant.

Glossaire

Les termes répertoriés dans ce glossaire concernent DB2 Alphablox. En outre, vous pouvez vous reporter aux annexes relatives aux «Termes et concepts OLAP», à la page 159. D'autres glossaires, plus complets, sont disponibles sur Internet, comme les sites A Web of Online Dictionaries et WhatIs.

2D : Se rapporte à un affichage bidimensionnel des informations, sous forme de lignes et de colonnes. Il peut s'agir de simples listes, rapports et feuilles de calcul bidimensionnelles.

3D : Se rapporte à un affichage multidimensionnel des informations, à l'aide de l'outil dimension de page, de ligne et de colonne. Gridblox présente ce type d'affichage.

A

accès à n-niveaux : Accès aux données par le biais de plusieurs niveaux de traitement informatique. Les architectures client-serveur de base utilisent un accès à deux niveaux : le client accède aux données stockées sur le serveur ; ce dernier transfère ensuite les résultats au client. Les architectures Intranet impliquent généralement un accès à trois niveaux : un navigateur client accède aux données via un serveur Web, qui à son tour transmet la demande d'accès aux données au serveur de la base de données. Appelé également *accès multiniveau*.

accès multiniveau : Voir accès à n-niveaux.

ACL : Liste ACL (liste de contrôle d'accès). Liste d'utilisateurs dont l'accès à un service en particulier est autorisé.

agencement de cadres : Ensemble de deux ou plusieurs sections contrôlable indépendamment (cadres) dans la fenêtre d'un navigateur, chaque section pouvant afficher un fichier HTML séparé. Le fichier affiché dans un cadre peut contenir des liens aux fichiers qui pourront apparaître dans un autre (ou le même) cadre. Généralement, un cadre contient les outils utilisateur de navigation et de sélection, alors que les autres cadres affichent les résultats des actions d'un utilisateur.

ancêtre : Dans une hiérarchie, membre d'une dimension dont le niveau de génération est supérieur à un autre membre. Voir ancêtre dans Termes et concepts OLAP pour voir un exemple.

API : Interface de programmation d'application utilisée par les programmeurs pour écrire le code source. L'API se compose des noms des bibliothèques

et des types et du nombre d'arguments qu'elles utilisent. Dans DB2 Alphablox, l'API se compose des méthodes Blox et des propriétés exposées dans Java et JavaScript.

applet : Voir l'applet Java.

applet Java : Programme écrit dans Java, langage développé par Sun Microsystems. Les applets Java sont intégrées aux pages Web et s'exécutent lorsque le navigateur affiche la page.

Application DB2 Alphablox : Combinaison de (1) fichiers JSP contenant un Blox, (2) d'informations d'accès pour les données utilisées par l'application et (3) ensemble de propriétés permettant de définir l'affichage des pages. L'application est hébergée sur un serveur Web et apparaît dans une fenêtre du navigateur en réponse à une requête utilisateur.

authentification : Processus de vérification de l'identité d'une personne ou d'un processus. De manière générale, l'utilisateur doit entrer un ID et un mot de passe valides pour répondre aux exigences du processus d'authentification.

axe : Coordonnée utilisée pour organiser et afficher les données. Un axe peut afficher une ou plusieurs dimensions. Pour les données multidimensionnelles, un axe se rapporte également à l'arête d'un cube.

- Un graphique linéaire ou de Gantt affiche également les données sur un axe horizontal (X) et un axe vertical (Y).
- Une grille permet de présenter les données multidimensionnelles à l'aide de lignes, de colonnes et d'axes de page. En outre, DB2 Alphablox peut afficher un axe appelé "Autre" qui indique les dimensions non utilisées à l'heure actuelle. Le fait de placer une dimension sur l'axe Autre la supprime de la vue courante mais elle reste disponible.

axe de la colonne : Orientation permettant de visualiser des données multidimensionnelles. L'axe de la colonne est l'un des trois axes visualisables (les deux autres étant l'axe de la ligne et l'axe de la page). Une colonne affiche les valeurs des données correspondant aux membres d'une dimension. Le fait de placer une dimension sur l'axe d'une colonne permet aux utilisateurs de visualiser les valeurs des données correspondant aux membres de la dimension. Par exemple, le fait de placer la dimension Temps sur l'axe de la colonne permet aux utilisateurs de visualiser les identificateurs d'un membre (tels que l'année et le trimestre) et les valeurs des données associées (telles que le volume des ventes).

axe de la ligne : Orientation permettant de visualiser des données multidimensionnelles. L'axe de la ligne est l'un des trois axes visualisables (les autres étant la page et la colonne). Une ligne affiche les valeurs des données pour les membres d'une dimension. Le fait de placer une dimension sur l'axe des lignes permet aux utilisateurs de visualiser des données pour les membres de cette dimension. Par exemple, placer la dimension Produit sur l'axe des lignes permet aux utilisateurs de visualiser les identificateurs d'un membre (tels que l'identificateur du produit ou son alias) et les valeurs des données associées (telles que les volumes des ventes).

axe de la page : Orientation permettant de visualiser des données multidimensionnelles. Une page définit une tranche au niveau d'une base de données multidimensionnelle. Le fait de placer une dimension sur l'axe des pages permet aux utilisateurs de sélectionner un membre de cette dimension et de ne visualiser que les données relatives à ce membre. Par exemple, si vous placez la dimension Marché sur l'axe des pages, les utilisateurs peuvent sélectionner New York à partir de cette dimension et ne visualiser que les données relatives à New York.

B

base de données d'application : Base de données hébergée sur un serveur devant prendre en charge les applications spécifiques. Une base de données d'application peut être extraite d'une base de données de production afin de permettre aux applications d'accéder rapidement aux données associées. DB2 Alphabloc utilise des bases de données multidimensionnelles et relationnelles pour organiser les données d'application.

base de données existante : Base de données, se trouvant généralement sur un ordinateur principal et qui utilise une technologie propriétaire. Il s'agit généralement d'un fichier sécurisé contenant des informations importantes pour l'entreprise.

base de données multidimensionnelle : Voir base de données OLAP.

base de données OLAP : Base de données à traitement analytique en ligne. Base de données utilisant des techniques d'indexation à des fins de traitement de requête optimal des données sommaires et des vues multidimensionnelles de ces données. DB2 OLAP Server est un exemple de base de données OLAP.

Blox : Composants logiciels réutilisables permettant un assemblage rapide des applications pour des réseaux intranet et Internet et des solutions d'analyse intégrées. Pour savoir comment assembler un Blox au sein d'une application, voir le *Developer's Guide*. Pour des informations détaillées sur chaque Blox, voir le *Developer's Reference*.

C

catalogue : Niveau organisationnel dans la structure d'une source de données. Dans DB2 OLAP Server ou Essbase, il s'agit d'une "application." De manière générale, les cubes de données sont regroupés dans des schémas, qui sont à leur tour regroupés dans des catalogues.

CGI : Common Gateway Interface. Interface de programmation utilisée pour exécuter des programmes sur des serveurs Web (HTTP). CGI permet à la structure de transférer des données d'un serveur au programme de passerelle d'un serveur (chargé du traitement) et de renvoyer les résultats au client depuis le programme passerelle vers le serveur Web. CONTR. ISAPI ou NSAPI.

classe CSS : Technologie CSS utilisée pour définir une extension ou remplacer le style normal d'un élément.

client complet : Client qui effectue des opérations de traitement lourd à l'aide d'un logiciel d'application client devant être installé sur le poste client. CONTR. client léger.

client léger : Client dont le volume du traitement effectué sur l'ordinateur client est minimal en comparaison du volume traité dans le serveur, tout en étant dépourvu d'un logiciel d'application. En contre-partie, le logiciel est téléchargé une fois qu'il est requis. Un navigateur Web Java est un client léger. Le navigateur peut afficher des pages Web et exécuter des applets intégrées. CONTR. client complet.

code HTML : Code langage intégré à un document HTML permettant au navigateur d'afficher le contenu d'une page Web. Les balises HTML intègrent un nom et des attributs facultatifs.

connecteur de base de données (database connector) : Interface de programmation offrant un langage commun permettant d'accéder à une base de données par le biais d'un réseau. Exemples : Mise en oeuvre des applications JDBC (Java Database Connectivity) for Java et ODBC (Open Database Connectivity) for Microsoft et Apple Macintosh.

cookie : Objet contenant des informations sur une session utilisateur telles que l'identificateur de la session enregistré sur le poste client.

CSV : Acronyme de Comma-separated values. Format de fichier contenant des lignes délimitées par un caractère de retour à la ligne formant des cases dans lesquelles les valeurs des données sont séparées par des virgules.

D

DAA : Voir Dynamic Application Assembly.

descendant : Dans une hiérarchie, membre d'une dimension dont le niveau de génération est inférieur à la racine (position la plus haute dans une hiérarchie). Voir descendant dans Termes et concepts OLAP pour voir un exemple.

développer : Afficher les détails d'une valeur. CONTR. réduire.

dimension : Attribut structurel d'un cube permettant de définir sa hiérarchie. Une dimension répertorie des membres, perçus par l'utilisateur comme des données du même type. Par exemple, une dimension Temps peut inclure les membres suivants : Année, Trimestre, Mois et Semaine. Une dimension Marché peut contenir les membres suivantes : Pays, Région, Etat et Ville. Une dimension permet de classer des données dans une source de données multidimensionnelle, facilitant ainsi l'analyse multidimensionnelle.

DLL : Bibliothèque de liaison dynamique (Dynamic Link Library). Fichier d'application s'exécutant dans le cadre du processus d'une application et de l'espace d'adresse. Les fichiers DLL sont chargés lorsqu'une application démarre et restent disponible tant qu'ils sont nécessaires, supprimant ainsi le temps d'extraction d'une application requis pour une application CGI.

DSS : Système d'aide à la décisions (Decision Support System). Système d'information et de planification prenant en charge une interrogation sur mesure, des analyses et des prévisions à l'aide de sources d'informations très diverses.

Dynamic Application Assembly : Approche dans laquelle les applications sont rapidement assemblées à l'aide d'un ensemble de composants bien définis.

E

élément de données : Dans une analyse multidimensionnelle, valeur se trouvant à l'intersection de deux ou plusieurs dimensions.

enfant : Dans une hiérarchie, membre d'une dimension dont le niveau de génération est inférieur à un autre membre. Voir enfant dans Termes et Concepts OLAP pour voir un exemple.

entrepôt de données : Registre central des données collectées par des systèmes d'application d'entreprise. Un entrepôt de données est généralement hébergé sur un serveur d'entreprise. Les données issues de plusieurs applications de traitement des transactions en ligne (OLTP) sont extraites et classées pour être utilisées dans le cadre d'applications analytiques et de requêtes utilisateur.

explorer vers le bas : Se déplacer vers le bas au sein de la hiérarchie des membres d'une dimension. Par exemple, un utilisateur peut visualiser des données financières d'Amérique du Nord, puis se déplacer vers

le bas dans la section Etats-Unis pour visualiser des données des états de l'ouest américain : Californie, Nord de la Californie et enfin Silicon Valley.

explorer vers le haut : Se déplacer vers le haut au sein des membres d'une hiérarchie. Par exemple, un utilisateur peut visualiser des données financières de la Silicon Valley puis se déplacer vers le haut en allant dans Nord de la Californie, Californie, états de l'ouest américain, et enfin Etats-Unis.

extranet : Site Web d'entreprise pouvant accéder au monde extérieur (Internet) avec des autorisations d'accès spécifiques telles que des mots de passe enregistrés. Les réseaux extranet facilitent les échanges avec les partenaires, les fournisseurs, les clients et toute autre personne avec laquelle une relation prédéfinie existe.

F

Feuilles de style en cascade (CSS) : Technologie utilisée par les développeurs de site Web afin de définir l'aspect des éléments d'une page HTML. Elles ressemblent aux feuilles de style utilisées par les programmes de traitement de texte. Pour en savoir plus, voir Cascading Style Sheets: Designing for the Web by Hakon Wium Lie and Bert Bos (Addison Wesley Longman 1997), ou <http://www.w3.org/Style/>.

filtre : Il permet de limiter des données issues d'une requête. Par exemple, vous pouvez définir un filtre pour qu'il ne renvoie que les données en fonction de l'année.

G

graphique : Niveau organisationnel dans la structure d'une source de données. Dans DB2 OLAP Server ou Essbase, ce niveau porte le nom "database." (base de données). De manière générale, des cubes de données sont regroupés en schéma qui sont eux-même regroupés en catalogue.

grille : Affichage multidimensionnel d'informations qui utilise des axes de colonne, de ligne et de page. Une grille permet aux utilisateurs d'explorer vers le bas (et vers le haut), de trancher et de pivoter l'affichage des données. GridBlox affiche les données dans une grille.

groupe : Suppression des en-têtes répétitives pour des lignes multiples ou des éléments de colonne. Mais aussi : Génération automatique de valeurs totales pour des éléments de données dans le groupe. Les programmeurs peuvent définir un groupe dans des bases de données relationnelles à l'aide de la clause SQL GROUP BY. Les programmeurs définissent un groupe dans des bases de données multidimensionnelles en définissant la hiérarchie de la base de données.

H

hiérarchie : Organisation des membres d'une dimension classés par rang et subordonnés les uns aux autres dans une relation parent/enfant. Par exemple, une hiérarchie dans une dimension Temps peut être l'année, le trimestre et le mois. Chaque mois est l'enfant d'un trimestre en particulier, et chaque trimestre est l'enfant d'une année en particulier.

HTML : Acronyme de HyperText Markup Language. Langage de balisage de texte qui permet la création de documents hypertextes affichables par un navigateur Web. HTML connecte le texte qui s'affiche à l'écran à d'autre texte, images, et autres objets, tels que les applets Java, qui peuvent se trouver ailleurs sur Internet.

HTTP : Protocole de transfert hypertexte (Hypertext Transfer Protocol). Ensemble de règles permettant l'échange de fichiers (texte, images, son, vidéo et multimédia) sur Internet ou sur un réseau intranet.

I

image au format GIF : Acronyme de Graphics Interchange Format. L'un des formats graphiques pris en charge par HTML.

image au format JPEG : Joint Photographic Experts Group. L'un des formats graphiques pris en charge par HTML. Pour être conforme aux exigences DOS, l'extension du format de fichier comporte généralement une extension à trois lettres du type .eps ou .jpg.

instancier : Créer un objet d'une classe spécifique. Par exemple, lorsqu'un utilisateur appelle une application DB2 Alphablox, DB2 Alphablox instancie (crée une instance de) cette application.

interface ISAPI : Internet Server Application Program Interface. Interface de programmation d'applications de Microsoft Windows, conçue spécifiquement pour les serveurs Web qui exécutent Microsoft Internet Information Server et vous permet d'écrire une application de serveur Web pouvant s'exécuter plus rapidement qu'une application CGI (Common Gateway Interface).

Internet : Réseau informatique mondial qui utilise un ensemble commun de protocoles de communication et logiciels. Internet permet aux utilisateurs de partager des informations numériques, placées sur des pages Web, disponibles sur plusieurs plateformes. Voir aussi extranet et intranet.

intranet : Réseau d'entreprise pour une utilisation en interne offrant des services comparables à ceux fournis par Internet. Voir aussi extranet.

J

JavaScript : Langage de script qui permet d'ajouter des fonctionnalités à une page ou à un site Web, ou d'en améliorer l'apparence. Les scripts JavaScript peuvent s'exécuter sur n'importe quel type d'ordinateur client ou serveur. Comme pour les applets Java, JavaScript peut être intégré à une page Web.

JDBC : Java Database Connectivity. API Java pour l'accès aux bases de données relationnelles. JDBC utilise l'interface SQL CLI (Call-Level Interface). Cette interface permet aux programmes d'utiliser les requêtes SQL pour accéder aux bases de données sans faire appel aux interfaces propriétaires de ces bases de données. Chaque base de données utilise un module ou un pilote qui gère la requête SQL et la convertit en une requête que le système de base de données peut comprendre.

JRE : JavaSoft Java Runtime Environment, composé de Java Virtual Machine, des classes core Java et des fichiers d'accompagnement. Il s'agit du composant d'exécution de J2SE Software Development Kit (JDK). Le JRE est l'ensemble le plus petit de fichiers exécutables et de fichiers dont se compose la plateforme standard Java.

JVM : Machine virtuelle Java (Java Virtual Machine). Entité permettant d'interpréter et d'exécuter le code Java.

L

langage SQL : Acronyme de Structured Query Language. Langage qui permet à l'interface utilisateur d'accéder aux données et de les extraire des Systèmes de gestion des bases de données relationnelles (RDBMS). Voir aussi ODBC SQL et langage SQL natif.

langage SQL natif : Langage de programmation d'une base de données relationnelle associé à une base de données en particulier. Le langage SQL natif comprend des fonctions qui sont optimisés pour cette base de données. Voir SQL. CONTR. : ODBC SQL.

M

magasin de données : Registre de données regroupées à partir d'opérations et d'autres sources pour répondre aux besoins de l'analyse et de la présentation des spécialistes de l'information. Ces données peuvent être issues d'une base de données d'entreprise ou d'un entrepôt de données, et se présente généralement sous une forme connue d'un groupe de spécialistes de l'information en particulier. Un magasin de données facilite l'accès et privilégie la convivialité en vue de l'exécution d'un travail donné, tel que l'analyse des ventes.

MDA : Analyse multidimensionnelle (Multidimensional analysis). Permet d'analyser des données en navigant dans l'arborescence et en affichant un sous-ensemble de base de données au cours d'une recherche, en explorant vers le bas, en visualisant par tranche, en pivotant et en définissant des calculs à effectuer sur les données.

métadonnées : Données décrivant d'autres données. Les schémas de base de données et les dictionnaires de données contiennent des métadonnées.

middleware : Logiciel qui sert d'intermédiaire entre des applications appartenant à des systèmes hétérogènes liés en réseau. Par exemple, les pilotes JDBC sont des middleware permettant aux programmes d'accéder aux bases de données sans utiliser d'interfaces propriétaires.

MIME : Multipurpose Internet Mail Extensions. Permet aux systèmes informatiques d'échanger des informations multimédia dans des formats audio, graphique, vidéo et texte à l'aide de normes de messagerie Internet.

N

niveau : Niveau au sein d'une hiérarchie informatique. Les environnements Internet/intranet mettent généralement en place une hiérarchie à trois niveaux : le niveau 1 est un navigateur s'exécutant sur une machine client ; le niveau 2 est un serveur Web (HTTP) s'exécutant sur une machine séparée ; le niveau 3 est un serveur de bases de données s'exécutant encore sur un autre ordinateur.

NSAPI : Netscape Application Program Interface. Cette interface vous permet d'écrire une application de serveur Web pouvant s'exécuter plus rapidement qu'une application CGI (Common Gateway Interface).

O

obsolète : Utilisé pour qualifier des paramètres, des classes et des méthodes qui peuvent être amenés à disparaître à l'avenir. Le caractère d'obsolescence intervient généralement lorsqu'une classe évolue et que son interface de programmation d'application (API) est modifiée : les paramètres et les méthodes sont renommés à des fins de cohérence et de nouveaux paramètres sont ajoutés. L'ancienne API reste en vigueur tant que les assembleurs travaillent à la transition de l'ancienne API vers la nouvelle, mais dès qu'une API devient obsolète, les assembleurs ne peuvent plus l'utiliser. DB2 Alphablox écrit des messages à la Console client dès qu'un élément obsolète est repéré.

ODBC : Open Database Connectivity. Interface de programmation d'une base de données qui rend possible l'accès à toutes les données à partir de

n'importe quelle application indépendamment du gestionnaire de la base (DBMS) qui gère les données. ODBC insère une couche intermédiaire, connue sous le nom de pilote de base de données, entre l'application et le gestionnaire de la base (DBMS). La couche intermédiaire traduit les requêtes de données d'une application sous forme de commandes que le gestionnaire de base (DBMS) peut comprendre.

ODBC SQL : Langage de programmation commun permettant d'accéder aux bases de données relationnelles sur un réseau. Pour permettre un accès presque universel aux bases de données, ODBC SQL peut être dépourvu de certaines des fonctions disponibles dans un langage SQL natif d'une base de données. Voir SQL. CONTR. : langage SQL natif.

P

page d'application : Page HTML unique faisant partie d'une application DB2 Alphablox.

page HTML : Document-texte pouvant contenir un balisage HTML standard, du texte, des graphiques, des objets multimédia, et un Blox. Les pages HTML s'affichent dans des navigateurs Web.

parent : Dans une hiérarchie, membre d'une dimension ayant un niveau de génération supérieur à un autre membre. Voir parent dans Termes et concepts OLAP pour voir un exemple.

persistance : Propriété d'un langage de programmation par laquelle les objets créés et les variables continuent d'exister et conservent leurs valeurs pendant la mise en veille du programme.

pivoter : Pour déplacer une dimension d'un axe vers un autre. Les assembleurs peuvent définir un pivot par le biais de la spécification de rapport Essbase ou à l'aide de JavaScript. Les utilisateurs peuvent effectuer une permutation des axes dans l'interface utilisateur.

protocole TCP/IP : Acronyme de Transmission Control Protocol/Internet Protocol. Ensemble de protocoles de communication utilisés pour connecter des ordinateurs hôte sur Internet. Également appelé Transport Control Protocol/Internet Protocol.

R

racine : Position la plus élevée dans une hiérarchie. Se reporter à racine dans Termes et concepts OLAP pour voir un exemple.

réduire : Condenser des informations détaillées dans une valeur récapitulatif unique. CONTR. développer.

référentiel : Base de données d'informations permettant d'identifier des objets pour les rendre disponible en vue d'une utilisation ultérieure. Le

référentiel DB2 Alphablox enregistre les objets utilisateur et application, les informations de contrôle d'accès et les paramètres de configuration.

regroupement : Lors de l'affichage des données numériques, processus de formatage des données en caractères numériques (en millier par exemple) généralement séparés par une virgule ou un point.

RVB : Les trois couleurs de base, rouge, vert et bleu, sur lesquelles repose un mode de composition de couleurs fondé sur le principe des couleurs additives.

S

serveur proxy : Type de serveur intermédiaire à mi-chemin entre le client et le serveur. Le serveur proxy intercepte l'ensemble des requêtes adressées au serveur qui ne lui sont pas destinées. Si elles lui sont destinées, il les fait suivre au serveur. Dans un environnement Internet/intranet, les serveurs proxy peuvent considérablement améliorer les performances d'un serveur Web en interceptant, sauvegardant et servant des pages qui sont fréquemment demandées.

serveur Web : Processus serveur qui envoie des pages HTML en réponse aux requêtes issues des navigateurs distants. Le terme se rapporte également à l'ordinateur sur lequel le processus s'exécute.

soeur/frère : Dans une hiérarchie, membre d'une dimension de la même génération qu'un autre membre. Voir soeuf/frère dans Termes et concepts OLAP pour voir un exemple.

spécification de rapport : Dans DB2 OLAP Server ou Hyperion Essbase, définition du résultat défini pour permettre à une application d'être disponible. La spécification de rapport contient les commandes d'extraction des données qui permettent de définir les données à extraire de la base de données et les commandes de formatage qui définissent les dimensions à afficher sur les axes des pages, colonnes et lignes dans une grille.

SQL multiple : Plusieurs instructions SQL permettant de former une seule requête complexe. Termes apparentés : requête corrélée, sous-requête et requête imbriquée.

T

table de hachage : Structure de données Java qui associe les clés aux valeurs. Ces exemples sont issus de l'API de Java JavaSoft. Le premier exemple crée une table de hachage de nombres et utilise les lettres de l'alphabet comme des clés (de sorte que A pointe vers 1, B vers 2, et ainsi de suite). Le deuxième exemple extrait une valeur spécifique (2) à l'aide de sa clé (B).

```
Hashtable letters = new Hashtable();
numbers.put("A", new Integer(1));
numbers.put("B", new Integer(2));
numbers.put("C", new Integer(3));

Integer n = (Integer)numbers.get("B");
if (n != null) {
    System.out.println("B = " + n);
}
```

thème : Ensemble d'éléments de conception tels que des polices de caractères et des images qui peuvent être appliquées dans une page HTML. Il est recommandé d'utiliser le même thème sur plusieurs pages afin de garantir l'homogénéité de la présentation.

Threadsafe : Code permettant des phases de déclenchement multiples, simultanées, entrelacées ou imbriquées qui ne peuvent pas interférer entre elles ou code empêchant des exécutions multiples et simultanées de se produire grâce à une sorte d'exclusion mutuelle.

tranche : Permet de filtrer un ensemble de données en plaçant une ou plusieurs dimension(s) sur l'axe de la page. Par exemple, un cube relatif aux données de ventes peut contenir le produit sur l'axe de la ligne, le mois sur l'axe de la colonne et l'année sur l'axe de la page. Si vous choisissez "année", seules les données relatives à la dimension Année s'afficheront.

TSL (HSB en anglais) : Modèle de couleurs, acronyme de Teinte, Saturation et Luminosité.

tuple : Dans DB2 Alphablox, ensemble de membres relatif à une entité qui définit un sous-ensemble d'un cube. Dans la grille ci-dessous, les tuples sur l'axe de la colonne sont South, East et West ; les tuples sur l'axe de la ligne sont VCR QTR1, TV QTR1 et TV QTR2.

		South	Est	Ouest
VCR	QTR 1	10	20	30
TV	QTR 1	20	30	40
TV	QTR 2	30	40	50

U

unité d'exécution : Partie d'un programme qui peut s'exécuter indépendamment des autres. Les systèmes d'exploitation qui prennent en charge le traitement multifilière permet aux développeurs de concevoir des programmes dont les éléments conçus avec des unités d'exécution peuvent s'exécuter simultanément.

URL : Acronyme d'Uniform Resource Locator. Projet de norme pour la spécification d'un objet, tel qu'un fichier ou un forum, sur Internet. Les adresses URL sont utilisées dans des documents HTML pour indiquer la cible d'un lien hypertexte.

V

Vecteur : Tableau extensible d'objets appelés éléments. Le nombre d'éléments dans un Vecteur peut augmenter ou diminuer automatiquement en fonction des instructions du programme. Le nombre d'éléments est défini par la taille du Vecteur et le volume de stockage par sa capacité.

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, Japon

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. Ce document 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

IBM

DB2 OLAP Server

WebSphere

DB2 Universal Database

Alphablox et Blox sont des marques d'Alphablox Corporation.

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

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

Java, JavaBeans, JavaScript 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

A

- abréviations, commandes de la console 142
- adaptateurs de données 4
- ADD, commande 143
- alias, Essbase 147
- Apache
 - sécurité 64
- Apache HTTP Server
 - enregistrement d'applications 38
- Apache Tomcat
 - migration d'applications 21
- Application Migration Utility (Utilitaire de migration d'applications)
 - accès 21
- Application Studio
 - présentation 22
- applications
 - arborescence 31
 - authentification 63
 - changement de nom 36
 - définition 32
 - définition, clusters WebLogic 37
 - définition, modification 36
 - définition, suppression 36
 - définition avec WebSphere 34
 - enregistrement sur le serveur Apache HTTP Server 38
 - enregistrement sur Microsoft IIS 39
 - enregistrement sur Sun iPlanet 39
 - gestionnaire 11
 - présentation 8
 - types 15
- Applications, onglet 119
- applications DB2 Alphablox,
 - Voir applications*
- applications J2EE
 - importation 40
- architecture
 - DB2 Alphablox 9
- architecture à n-niveaux 8
- arrêt de DB2 Alphablox 25
- authenticate(), méthode de l'interface IUser 113
- authentification
 - activation pour DB2 Alphablox 79
 - DB2 Alphablox contre serveur Web 64
 - domaines de sécurité 63
 - modes de sécurité 61
- authorize(), méthode de l'interface IUser 113

B

- bases de données
 - adaptateurs de données 4
 - configuration du pilote relationnel Sybase JConnect 48
 - sécurité Microsoft Analysis Services 45
 - sources de données, définition 43
- Bases de données
 - termes OLAP 162
- bases de données relationnelle
 - pilotes JDBC, mise à jour 49

- bases de données relationnelle (*suite*)
 - Sybase JConnect, configuration de l'environnement pour 48
 - traçage JDBC 48
- bases de données relationnelles
 - référentiel, utilisation comme 119
- Blox
 - composants dans les applications DB2 Alphablox 17

C

- cache de structure,
 - Voir commandes relatives à la mémoire cache*
- calculs
 - extension de DB2 Alphablox 73
- chemin de classes 75
- client DHTML
 - incidents connus 29
 - Feuilles de style en cascade (CSS) 29
- clusters, DB2 Alphablox
 - présentation 137
- collections de commentaires
 - création et gestion 85
- commande CLUSTER SHUTDOWN
 - description 139
- commande de console
 - RESOLVEALIASSTOBASEMEMBERS 147
- commande GET SERVICE CLUSTER
 - description 139
 - exemple 138
- commande ISCLUSTERED 139
- commande LEADHOST 140
- commande MAXHOSTS 140
- commande PORTNUM 140
- commande SET SERVICE CLUSTER 139
- commande SHOW HOSTS
 - description 140
- commande STARTUPTIME 140
- commandes relatives à la mémoire cache
 - DB2 OLAP Server 146
 - Essbase 146
- CommentsBlox
 - création de collections de commentaires 85
- Composants Blox
 - description 5
- comptes, utilisateurs,
 - Voir utilisateurs*
- configuration système requise
 - applications FastForward 152
- connexion unique 105
- console
 - abréviations 142
 - accès 141
 - commandes de cluster 138
 - Commandes spécifiques à DB2 OLAP Server 146
 - Commandes spécifiques à Essbase 146
 - fichiers texte, exécution via la 149
 - gestionnaire 12
 - HTML 141
 - liste des commandes, DB2 Alphablox 143
 - niveau des messages 149

console (*suite*)
 syntaxe des commandes 142
 Telnet 141
Console
 accès 20
 DB2 Alphablox, arrêt à l'aide de 26
 fichier de commandes, utilisation 78
console Telnet 141
 port, valeur par défaut 80
containsGroup(), méthode de l'interface IGroup 116
containsUser(), méthode de l'interface IGroup 116
CREATE, commande 143
Créateur de modèle de blox
 lien Exemples sous l'onglet Assemblage 23
cube relationnel
 Voir Cube Server Administrator's Guide

D

DB2 Alphablox
 accès 25
 architecture 9
 arrêt 25
 clusters 137
 configuration requise,
 Voir Guide d'installation
 console,
 Voir Console
 démarrage 25
 durée d'inactivité du serveur 78
 gestionnaires définis 9
 modes d'authentification et de sécurité 61
 ports, indication 80
 présentation 1
 propriétés de démarrage, configuration 77
 propriétés personnalisées,
 Voir propriétés personnalisées
 propriétés système, configuration 79
 propriétés utilisateur,
 Voir propriétés utilisateur
 référentiel, base de données 119
 sécurité,
 Voir sécurité
 sécurité reposant sur le serveur Web, configuration 70
 services définis 9
 tâches d'administration 25
 utilisateurs, création 51
DB2 Alphablox, page d'accueil
 Applications, onglet 119
DB2 Alphablox Cube Manager
 nombre maximal de cubes, indication 81
DB2 Alphablox Cube Server
 Voir aussi Cube Server Administrator's Guide
 source de données, définition 43
DB2 OLAP Server
 bibliothèques client, mise à jour 26
 commande de la console 146
 show client library version 146
DELETE, commande 143
DELETE OUTLINECACHE, commande DB2 OLAP
 Server 148
DELETE OUTLINECACHE, commande Essbase 148
domaine JNDI
 modèle de sécurité 62
droit utilisateur Agir en tant que partie du système
 d'exploitation 45
durée d'inactivité, serveur 78

durée d'inactivité du serveur,
 Voir Alphablox Analytics

E

Essbase
 bibliothèques client, mise à jour 26
 commande de la console 146
 show client library version 146
EXIT, commande 143
exploration des répertoires, désactivation 71
extensibilité
 client DHTML 74
 modèle d'interface utilisateur de blox 74
Extensible User Manager 99
 extension de DB2 Alphablox 74
 interfaces de programmation 103

F

FastForward
 administration 154
 configuration système requise 152
 création d'applications 153
 journal d'application, utilisation 157
 paramètres généraux 157
 propriétés d'application, gestion 157
 rapport, types 154
 rôle administrateur, modification 153
 rôles utilisateur 151
Feuilles de style en cascade (CSS)
 paramètres et client DHTML 29
fichier blox.tld 32
fichier web.xml 32
fichiers de classe 23
fichiers JAR 23
fichiers journaux
 changement de nom 90
 gestion 91
 niveau des messages, définition 149
 niveaux des messages, définition 78, 79
 permettre à DB2 Alphablox d'écrire des messages vers 78
 propriétés de démarrage, définition 77
fichiers journaux, DB2 Alphablox 90
 Voir aussi fichiers journaux
 paramètres d'intervalle de remplacement des fichiers
 journaux 90
fichiers texte, exécution via la console 149
findGroup(), méthode de l'interface IUserManager 109
findUser(), méthode de l'interface IUserManager 109

G

gestion des collections de commentaires 85
gestionnaire, référentiels 119
gestionnaire d'applications 11
gestionnaire d'utilisateurs 10
 commande de console Telnet 102
 extension 74
 moteur de personnalisation 99
gestionnaire d'utilisateurs, LDAP 99
gestionnaire de clusters 12
gestionnaire de données 11
gestionnaire de référentiels 12, 119
gestionnaire de requêtes 9
gestionnaire de services 9

- gestionnaire de sessions 10
- GET, commande 144
- getEmail(), méthode de l'interface IUser 113
- getExternalProperties(), méthode de l'interface IUserManager 110
- getFullName(), méthode de l'interface IUser 114
- getName(), méthode de l'interface IGroup 116
- getName(), méthode de l'interface IUser 114
- getPassword(), méthode de l'interface IUser 114
- getPrincipleUserName(), méthode de l'interface IUserManager 110
- getPropertiesSubset(), méthode de l'interface IGroup 117
- getPropertiesSubset(), méthode de l'interface IUser 115
- groupes
 - appartenance 53
 - appartenance d'un utilisateur ou d'un groupe, modification 53
 - création 55
 - modification 57
 - rôles, modification de l'appartenance 60
 - sous-groupes, présentation 56
 - suppression 57

H

- hasExternalEditor(), méthode de l'interface IUserManager 110
- HELP, commande 144

I

- isUserInRole(), méthode de l'interface IUser 115

J

- JDBC
 - pilote, mise à jour de la version 49
 - traçage 48
- journaux
 - journal d'application, FastForward 157

K

- KILL, commande 144

L

- LDAP-based User Manager 99
- LDAP-based User Manager, configuration 99
- liens d'en-tête
 - configuration 33
- liste de clients autorisés, indiquer 79
- Listes ACL,
 - Voir* rôles
- listes de contrôle d'accès,
 - Voir* rôles
- LOAD, commande 144
- LOCK, commande 144
- logiciel de blocage de fenêtre en incrustation 29

M

- MESSAGE, commande 144
- Microsoft Analysis Services
 - authentification, configuration 45

- Microsoft Analysis Services (*suite*)
 - droits utilisateur Windows, ajout 45
 - performances 130
 - regroupement de connexions 130
 - source de données, définition 43

- Microsoft IIS
 - comptes utilisateur, génération auto 70
 - configuration de la sécurité 65
 - droits de l'utilisateur anonyme, restriction 67
 - droits sur les répertoires, définition 71
 - enregistrement d'applications 39
 - exploration des répertoires, désactivation 71
 - paramètres de sécurité, activer 66

- Microsoft SQL Server
 - source de données, définition 43
 - types d'authentification 44

- mise en cluster
 - gestionnaire de clusters 12
- mise en clusters
 - démarrage 137
 - liste de commandes, mise en clusters 138

- modèle de sécurité
 - Alphablox 8.4.1 62
 - Domaine JNDI 62
 - Module de connexion Alphablox 62

- mon profil
 - accès à 52

- Mon profil
 - lien 23
- moteur de personnalisation,
 - Voir* Extensible User Manager

N

- nom d'application 32
- nom d'instance
 - conversion du référentiel, spécification 121
 - indication pour DB2 Alphablox 77

O

- OBJECTS, commande 144
- Onglet Administration 20
- onglet Applications 133, 137
- Onglet Applications 19
- Onglet Assemblage 22

P

- page d'accueil DB2 Alphablox
 - onglet Applications 133, 137
- Page d'accueil de DB2 Alphablox
 - Onglet Administration 20
 - Onglet Applications 19
 - Onglet Assemblage 22
- paramètres de chemin de classes
 - pilotes JDBC, modification 50
- paramètres généraux, FastForward 157
- pilotes, JDBC, mise à jour 49
- pilotes JDBC
 - ajout d'éléments supplémentaires 49
 - paramètres du chemin de classes, modification 50
- Portal Theme Utility (Utilitaire de thèmes de portail)
 - accès 21
- ports
 - indication 80

- présentation
 - DB2 Alphablox 1
- propriétés
 - démarrage, configuration 77
 - personnalisées,
 - Voir* propriétés personnalisées
 - système 79
 - utilisateur 52
- propriétés de démarrage
 - configuration 77
 - durée d'inactivité du serveur 78
 - journal DB2 Alphablox, activation 78
 - niveau des messages, console DB2 Alphablox, valeur par défaut 78
 - niveau des messages, valeur par défaut 78
 - nom d'instance 77
 - nom du fichier de commandes 78
 - nom du fichier journal 78
- propriétés personnalisées
 - définies 82
 - propriétés d'applications, définition 84
 - propriétés d'applications, modification 84
 - propriétés d'applications, suppression 85
 - propriétés utilisateur, définition 82
 - propriétés utilisateur, modification 83
 - propriétés utilisateur, suppression 83
 - valeurs par défaut 82
- propriétés système
 - configuration 79
 - préfixe URL du serveur Web, définition 79
- propriétés utilisateur
 - définition 82
 - modification 52
 - personnalisées, modification 83

R

- rapports, FastForward 154
- rapports PDF
 - création d'un processeur PDF distant 88
- référentiel, DB2 Alphablox
 - gestionnaire 12
 - présentation 119
 - utilitaire de conversion du registre 120
- Référentiel, DB2 Alphablox
 - avantages du référentiel relationnel 119
 - configuration 120
 - configuration d'une instance, afin qu'elle utilise un référentiel existant 125
 - type, vérification 120
- référentiel DB2 Alphablox,
 - Voir aussi* Gestion de la configuration, Alphablox Analytics
 - Voir aussi* référentiel, Alphablox Analytics
 - Voir aussi* Référentiel, Alphablox Analytics
 - Voir* Référentiel Alphablox Analytics
- refresh(), méthode de l'interface IGroup 117
- refresh(), méthode de l'interface IUser 115
- regroupement de connexions
 - BEA WebLogic 135
 - DB2 OLAP Server 129
 - Hyperion Essbase 129
 - Microsoft Analysis Services 130
 - référentiel DB2 Alphablox 134
 - relationnel, utilisation DB2 Alphablox 133
 - sources de données DB2 Alphablox 134
 - sources de données relationnelles 133
- RELEASE, commande 144

- REMOVE, commande 144
- répertoire WEB-INF 32
- REPORT, commande 145
- resume(), méthode de l'interface IUserManager 110
- RESUME, commande 145
- rôles
 - appartenance, modification 59
 - définition 59
 - groupes et utilisateurs, modification 53, 60
 - suppression 60
- RUN, commande
 - syntaxe 145

S

- SAVE, commande 145
- scripts SQL Sybase 48
- sécurité
 - Voir aussi* rôles
 - activation pour DB2 Alphablox 79
 - admin contre droits d'utilisateur 63
 - Authentification Windows à l'aide de Microsoft IIS 65
 - comptes utilisateur, génération automatique 70
 - contenu des répertoires, accès 71
 - domaines et accès aux applications 63
 - implémentations personnalisées 104
 - Microsoft Analysis Services 45
 - modes d'authentification 61
 - Options Sun iPlanet 64
 - propriétés système, définition 79
 - rôles, utilisation 59
 - Web, définition 70
 - Web contre DB2 Alphablox 64
- serveur SMTP, indication 80
- serveurs Web
 - configuration de la sécurité Microsoft IIS 65
 - DB2 Alphablox, configuration de la sécurité 70
 - enregistrement d'applications 38
 - préfixe URL, indiqué 79
- service, en tant que DB2 Alphablox 47
- SET, commande 145
- setCaseSensitiveGroups(), méthode de l'interface IUserManager 111
- setCaseSensitiveUsers(), méthode de l'interface IUserManager 111
- SHOW, commande 146
- SHOW HOSTS, commande
 - description 146
- SHOW OUTLINECACHE, commande DB2 OLAP Server 147
- SHOW OUTLINECACHE, commande Essbase 147
- source de données DB2 UDB, définition 43
- source de données Oracle, définition 43
- source de données Sybase, définition 43
- sources de données
 - définition 43
 - définition, changement 44
 - définition, suppression 45
 - sécurité Microsoft Analysis Services 45
- sous-groupes
 - création 55
 - définis 56
- SQL Server, Microsoft
 - types d'authentification 44
- start(), méthode de l'interface IUserManager 111
- START, commande 146
- STATISTICS, commande 146
- stop(), méthode de l'interface IUserManager 112

- STOP, commande 146
- Sun iPlanet
 - enregistrement d'applications 39
 - options de sécurité 64
- suspend(), méthode de l'interface IUserManager 112
- SUSPEND, commande 146

T

- termes et concepts OLAP 159
- thème
 - valeur par défaut, indication 80

U

- Utilisateur Alphablox 63
- Utilisateur authentifié Alphablox 63
- utilisateur de confiance, défini 65
- Utilisateur invité
 - restriction de l'accès aux applications 63
- utilisateurs
 - appartenance à un groupe 53
 - création 51
 - génération automatique des comptes 70, 79
 - modification 52
 - profil, modification 52
 - propriétés, modification 52
 - propriétés personnalisées, modification 83
 - rôles, modification de l'appartenance 60
 - suppression 53
- utilitaire, conversion du registre 119, 120
- Utilitaire de bibliothèque de client Essbase
 - utilisation 26
- utilitaire de conversion, registre 119, 120
- utilitaire de conversion du registre 119
- Utilitaire de conversion du registre
 - arguments 127
 - base de données en fichier 124
 - démarrage 121
 - fichier en base de données 123
 - opération, descriptions 126
 - syntaxe, ligne de commande 126
 - utilisation 120

W

- WebLogic
 - regroupement de connexions 135
- WebSphere
 - applications, définition avec 34
- WebSphere Portal Server
 - thèmes 21
- Windows
 - droits utilisateur, configuration pour Microsoft Analysis Services 45
 - service, configuration lors de l'utilisation de Microsoft Analysis Services 47



Numéro de programme : 5724-L14

GC11-2170-00



Spine information:



IBM DB2 Alphablox

IBM DB2 Alphablox - Guide d'administration

Version 8.4