



Common Event Infrastructure



Common Event Infrastructure

Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section Remarques située à la fin du présent document.

mars 2008

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

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

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

Vous pouvez également consulter les serveurs Internet suivants :

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

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

© Copyright IBM France 2008. Tous droits réservés.

© **Copyright International Business Machines Corporation 2007, 2008. All rights reserved.**

Table des matières

Avis aux lecteurs canadiens v

Chapitre 1. Infrastructure CEI (Common Event Infrastructure) 1

Modèle d'événement de base commun. 3

Chapitre 2. Configuration de l'infrastructure d'événement commune (CEI) 5

Composants de Common Event Infrastructure 6

Configuration de Common Event Infrastructure à l'aide de la console d'administration 8

Déploiement de l'application Common Event Infrastructure. 10

Déploiement de l'infrastructure CEI sur un cluster 11

Configuration de la messagerie d'événements . . . 14

Configuration de files d'attente JMS supplémentaires 14

Configuration de la messagerie d'événements à l'aide d'un fournisseur JMS externe 15

Configuration de l'alias d'authentification JMS . . 17

Configuration de la base de données d'événements 18

Restrictions relatives à la base de données d'événements. 18

Configuration d'une base de données d'événements Derby 19

Configuration d'une base de données d'événements DB2 (Linux, UNIX et Windows). . . 21

Configuration d'une base de données DB2 sur un système z/OS 23

Configuration d'une base de données DB2 sur un système iSeries 25

Configuration d'une base de données d'événements Informix 27

Configuration d'une base de données d'événements Oracle 29

Configuration d'une base de données d'événements SQL Server. 31

Exécution manuelle des scripts de configuration de base de données. 33

Mise à niveau de la base de données d'événements depuis une version antérieure . . 41

Configuration de l'infrastructure d'événement commune entre cellules pour WebSphere Business Monitor 48

Suppression de la configuration de l'infrastructure CEI 51

Suppression de la base de données d'événements 51

Suppression de l'application Common Event Infrastructure. 52

Suppression de l'application d'entreprise de messagerie d'événements. 53

Chapitre 3. Administration de l'infrastructure CEI (Common Event Infrastructure) 55

Administration du service Evénements à l'aide de la console d'administration 55

Activation et désactivation du service

Evénements à l'aide de la console

d'administration. 55

Création d'une fabrique d'émetteurs d'événement à l'aide de la console d'administration 56

Création d'un groupe d'événements à l'aide de la console d'administration 57

Création d'un filtre d'événements à l'aide de la console d'administration 58

Administration du service Evénements à l'aide de scripts 59

Activation du service Evénements à l'aide d'un script 59

Désactivation du service Evénement à l'aide d'un script 60

Création d'une fabrique d'émetteurs à l'aide d'un script 61

Création d'un groupe d'événements à l'aide d'un script 64

Création d'un filtre d'événements à l'aide d'un script 66

Journalisation et fonction de trace des composants Common Event Infrastructure 67

Maintenance de la base de données d'événements 68

Mise à jour des statistiques relatives à la base de données d'événements DB2 68

Réorganisation des tables de bases de données d'événements DB2 69

Purge des événements de la base de données d'événements. 70

Chapitre 4. Sécurisation de l'accès aux fonctions de Common Event Infrastructure 77

Chapitre 5. Identification et résolution des incidents liés à Common Event Infrastructure 81

Problèmes lors du démarrage 81

Le service Evénements ne démarre pas (message CEIDS0058E) 81

Problèmes relatifs à l'envoi d'événements 82

Erreur lors de l'envoi de l'événement (message CEIDS0060E) 82

Erreur lors de l'envoi de l'événement (ServiceUnavailableException) 83

Erreur lors de l'envoi de l'événement (NameNotFoundException) 83

Erreur lors de l'envoi de l'événement (message CEIEM0025E).	84
Erreur lors de l'envoi de l'événement (message CEIEM0034E).	85
Événement non valide (message CEIEM0027E)	86
Mode de synchronisation non pris en charge (message CEIEM0015E)	87
Mode de transaction non pris en charge (message CEIEM0016E).	88
Problèmes relatifs à la réception ou à la recherche d'événements.	88
Erreur lors de la recherche d'événements (message CEIDS0060E)	88
Événements non stockés dans le magasin de données persistantes	89
Événements non reçus par les consommateurs (aucun message d'erreur).	90
Événements non reçus par les consommateurs (NameNotFoundException)	92
Un groupe d'événements avec des éléments de données étendus ne comprend aucun événement.	93
Erreur lors de la recherche d'un groupe d'événements (message CEIES0048E).	94
Problèmes divers	94
La demande de modèle de catalogue d'événements échoue sur un système Windows	94

configEventServiceDB2DB	96
configEventServiceDB2iSeriesDB	98
configEventServiceDB2ZOSDB.	101
configEventServiceDerbyDB	104
configEventServiceInformixDB	106
configEventServiceOracleDB	108
configEventServiceSQLServerDB	111
deployEventService	113
deployEventServiceMdb	115
setEventServiceJmsAuthAlias	117
enableEventService	118
disableEventService	120
showEventServiceStatus	121
removeEventService	122
removeEventServiceMdb	123
removeEventServiceDB2DB.	124
removeEventServiceDB2iSeriesDB	126
removeEventServiceDB2ZOSDB	127
removeEventServiceDerbyDB	129
removeEventServiceInformixDB	130
removeEventServiceOracleDB	132
removeEventServiceSQLServerDB	134
eventbucket	136
eventpurge	137

Remarques	139
----------------------------	------------

Chapitre 6. Commandes de l'infrastructure d'événement commune (CEI). 95

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. Infrastructure CEI (Common Event Infrastructure)

Le Common Event Infrastructure est une technologie pouvant être imbriquée et destinée à fournir les principaux services de gestion des événements aux applications qui en ont besoin.

Cette infrastructure sert de point d'intégration pour la consolidation et la conservation des événements bruts en provenance de diverses sources hétérogènes et la distribution de ces événements aux consommateurs d'événements. Les événements sont représentés à l'aide du modèle d'événements de base communs (Common Base Event), un format XML standard définissant la structure d'un événement. Pour plus d'informations, voir la sous-rubrique relative aux modèles Common Base Event.

A l'aide de cette infrastructure commune, divers produits qui n'interagissaient que très peu entre eux peuvent désormais intégrer leur gestion des événements, offrant ainsi une vue globale des ressources de l'entreprise et corrélant les événements de différents domaines. Par exemple, les événements générés par une application de surveillance réseau peuvent être corrélés aux événements générés par une application de sécurité. Une telle corrélation peut être difficile à accomplir lorsque chaque produit utilise sa propre approche de la gestion des événements.

Le Common Event Infrastructure simplifie la génération, la propagation, la conservation et la consommation d'événements, mais il ne définit pas les événements lui-même. C'est le rôle des développeurs d'applications et des administrateurs de définir les types d'événement, les groupes d'événements, le filtrage et la corrélation.

Composants de Common Event Infrastructure

Le Common Event Infrastructure comprend les principaux composants suivants :

Common Base Event

Le composant Common Base Event prend en charge la création des événements et l'accès à leurs données de propriété. Les sources d'événements utilisent les API de Common Base Event pour créer des événements conformément au modèle Common Base Event ; les consommateurs d'événements utilisent les API pour lire les données de propriété des événements reçus. En outre, les applications peuvent convertir les événements au format XML et prendre en charge les échanges avec d'autres outils. Le composant Common Base Event fait partie de l'environnement TPTP (Test and Performance Tools Platform) d'Eclipse.

Emetteur

Le composant émetteur prend en charge l'envoi des événements. Une fois qu'une source d'événements a créé un événement et l'a alimenté en données, elle le soumet à un émetteur. L'émetteur peut compléter automatiquement le contenu, puis il valide l'événement pour s'assurer qu'il respecte la spécification Common Base Event. Il compare également l'événement aux critères de filtrage configurables. Si l'événement est valide et qu'il n'est pas bloqué par les critères de filtrage, l'émetteur l'envoie au service d'événements. Un émetteur peut envoyer des événements au service d'événements en mode synchrone (via des appels Enterprise JavaBeans) ou en mode asynchrone (via une file d'attente Java Message Service).

Service Événements

Le service Événements désigne le conduit entre les sources d'événements et les consommateurs d'événements. Ce service reçoit les événements soumis aux émetteurs par les sources d'événements. Il les stocke dans un magasin de données persistantes, puis les distribue de manière asynchrone aux consommateurs d'événements abonnés. En outre, le service Événements prend en charge les requêtes synchrones d'événements historiques provenant du magasin persistant.

Catalogue d'événements

Le catalogue d'événements est un référentiel contenant des métadonnées d'événements. Les applications utilisent le catalogue d'événements pour extraire des informations sur les classes d'événements et le contenu autorisé.

En outre, une application ou une solution utilisant le Common Event Infrastructure peut également inclure les composants suivants (qui ne font pas partie de l'infrastructure elle-même) :

Source d'événements

Une source d'événements est une application qui utilise un émetteur pour envoyer des événements au service Événements.

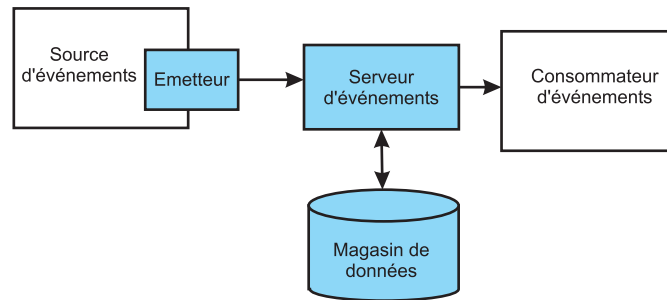
Consommateur d'événements

Un consommateur d'événements est une application qui reçoit des événements du service Événements.

Application catalogue d'événements

Une application catalogue d'événements est une application qui stocke ou extrait les métadonnées d'événement du catalogue d'événements. Il peut s'agir d'un outil de gestion ou de développement mais aussi d'une source d'événements ou d'un consommateur d'événements.

Le diagramme suivant présente le flot d'événements général entre une source d'événements et un consommateur d'événements via le Common Event Infrastructure.



Modèle d'événement de base commun

Le modèle d'événement de base commun (Common Base Event) correspond à une norme définissant une représentation commune des événements, conçue pour une utilisation par les applications métier et de gestion des entreprises. Cette norme, développée par IBM Autonomic Computing Architecture Board, prend en charge le codage de la journalisation, du traçage, de la gestion et des événements métier utilisant un format de type XML commun. Ainsi, il est possible d'effectuer une corrélation entre différents types d'événement provenant de différentes applications. Le modèle d'événement de base commun fait partie d'IBM Autonomic Computing Toolkit ; pour en savoir plus, voir <http://www.ibm.com/autonomic>.

Le Common Event Infrastructure prend actuellement en charge la version 1.0.1 de la spécification.

Le concept de base inhérent au modèle Common Base Event est la *situation*. Une situation est toute action se produisant à n'importe quel emplacement dans l'infrastructure informatique, tel un arrêt serveur, une erreur d'unité disque ou une connexion utilisateur n'aboutissant pas. Le modèle Common Base Event définit un ensemble de types de situation standard qui correspondent à la plupart des situations qui peuvent survenir (par exemple, StartSituation et CreateSituation).

Un *événement* est une notification structurée qui comporte les informations relatives à une situation. Un événement signale trois types d'information :

- La situation elle-même (événement survenu)
- L'identité du composant concerné (par exemple, le serveur qui a été arrêté)
- L'identité du composant signalant la situation (qui peut être identique à celle du composant concerné)

La spécification Common Base Event définit un événement comme un élément XML contenant des propriétés qui fournissent trois types d'informations. Ces propriétés sont codées sous forme d'attributs et de sous-éléments de l'élément racine, CommonBaseEvent.

Le format Common Base Event est extensible. En plus des propriétés d'événement standard, un événement peut contenir des éléments de données étendus, qui sont des éléments spécifiques aux applications, pouvant contenir toutes sortes d'informations relatives à la situation. L'attribut *extensionName* permet de libeller un événement avec un nom de classification (une classe d'événements), qui indique aux applications le type d'éléments de données étendus auquel elles doivent s'attendre. Le catalogue d'événements stocke les définitions d'événement qui décrivent ces classes d'événement et leur contenu autorisé.

Pour obtenir des instructions détaillées relatives au format Common Base Event, voir le document de spécification et le schéma XSD inclus dans IBM Autonomic Computing Toolkit.

Chapitre 2. Configuration de l'infrastructure d'événement commune (CEI)

Vous pouvez configurer des ressources Common Event Infrastructure ou modifier les ressources existantes, à l'aide de l'objet serveur AdminTask

A propos de cette tâche

Vous pouvez installer Common Event Infrastructure (CEI) avec une configuration par défaut totalement fonctionnelle sur une configuration de serveur autonome. N'agissez ainsi que lorsque vous créez un profil de serveur autonome à l'aide d'outils de gestion de profil. Dans tous les autres cas, utilisez la console d'administration pour configurer l'infrastructure CEI (par exemple si vous l'installez en environnement ND ou sur un cluster) afin de définir une configuration bien adaptée à votre système.

Vous pouvez également utiliser la commande wsadmin pour configurer l'infrastructure d'événement commune ou pour modifier une configuration d'infrastructure d'événement commune existante. Dans les deux cas, vous devez modifier la configuration de l'infrastructure CEI en utilisant un objet serveur AdminTask afin d'exécuter les commandes d'administration.

Après avoir modifié la configuration de l'infrastructure CEI, vous devez redémarrer le serveur ou le cluster.

Composants de Common Event Infrastructure

Les composants de Common Event Infrastructure sont installés sous la forme d'un ensemble d'applications, de services et de ressources sur le serveur.

Lorsque vous configurez Common Event Infrastructure, un nombre de composants est créé et déployé sur votre serveur.

service Common Event Infrastructure

Service installé sur le serveur permettant aux applications et aux clients d'utiliser Common Event Infrastructure. Vous pouvez visualiser la configuration du service Common Event Infrastructure dans la console d'administration, comme suit :

- Pour un serveur, sélectionnez **Serveurs > Serveurs d'applications > nom_serveur > Business Integration > Common Event Infrastructure > Common Event Infrastructure Service.**
- Pour un cluster, sélectionnez **Serveurs > Clusters > nom_cluster > Business Integration > Common Event Infrastructure > Common Event Infrastructure Service.**

Si la case intitulée "Activer le serveur d'infrastructure d'événement" est cochée, cela signifie que le service est installé et s'exécute ou qu'il sera activé après le redémarrage du serveur ou du cluster. Si cette case n'est pas cochée, cela signifie que le service n'est pas installé ou sera désinstallé après le redémarrage de votre serveur ou cluster.

Paramètres relatifs au service Événements

Ensemble de propriétés utilisées par le service Événements permettant la distribution et la conservation d'événements à l'aide des caractéristiques du magasin de données. En règle générale, aucune configuration n'est nécessaire pour cette ressource, mais vous pouvez être amené à créer des paramètres de services d'événements supplémentaires si vous souhaitez configurer plusieurs services d'événements dans la même cellule. Pour consulter les paramètres de services d'événements, cliquez sur **Intégration de services > Service Événements > Paramètres du service Événements.**

Configuration de la messagerie d'événements

Il s'agit des ressources prenant en charge la transmission d'événements en mode asynchrone vers le service Événements à l'aide de JMS (Java Messaging Service). La configuration de messagerie par défaut utilise la messagerie intégrée au serveur. En option, vous pouvez configurer un fournisseur JMS externe pour la messagerie d'événements.

Base de données d'événements

La base de données d'événements permet de stocker de façon permanente les événements reçus par le service Événements. La base de données Derby fait partie intégrante du serveur, mais il n'est pas conseillé de l'utiliser dans des environnements de production. En revanche, vous pouvez configurer une base de données d'événements externe sur les produits suivants : DB2, Oracle, SQLServer et Informix.

Plug-in de filtrage de définition d'événement

Un plug-in de filtrage permet de filtrer les événements à la source à l'aide de sélecteurs d'événements XPath. Pour configurer les propriétés de filtrage, cliquez sur **Intégration de services > Common Event Infrastructure > Fabrique d'émetteurs d'événement > Paramètres d'émetteurs d'événement.**

Fabrique d'émetteurs

Une fabrique d'émetteurs est un objet utilisé par les sources d'événements pour créer des émetteurs ; un émetteur permet d'envoyer des événements au service d'événements. Les propriétés d'une fabrique d'émetteurs affectent le comportement de tout émetteur créé à l'aide de la fabrique d'émetteurs. Pour consulter les fabriques d'émetteurs disponibles, cliquez sur **Intégration de services > Common Event Infrastructure > Fabrique d'émetteurs d'événement**.

Transmission de services d'événements

Une transmission de services d'événements est un objet définissant les propriétés qui déterminent la façon dont les émetteurs accèdent au service Événements en mode synchrone via des appels EJB ; ces propriétés sont utilisées par les fabriques d'émetteurs lors de la création d'émetteurs. Vous pouvez consulter ou modifier les transmissions de services d'événements disponibles dans les paramètres relatifs à la fabrique d'émetteurs.

Transmission JMS (Java Message Service)

Une transmission JMS est un objet définissant les propriétés qui déterminent la façon dont les émetteurs accèdent au service Événements en mode asynchrone via une file d'attente JMS ; ces propriétés sont utilisées par les fabriques d'événements lors de la création d'émetteurs. Vous pouvez consulter ou modifier les transmissions JMS disponibles dans les paramètres relatifs à la fabrique d'émetteurs.

Groupe d'événements

Un groupe d'événements est une collection logique d'événements utilisée pour classer les événements en fonction de leur contenu. Lorsqu'un consommateur d'événements envoie une requête à partir du serveur d'événements ou s'abonne à une distribution d'événements, il peut indiquer un groupe d'événements pour extraire uniquement les événements de ce groupe. Les groupes d'événements permettent également de spécifier les événements à stocker dans un magasin de données permanent. Pour consulter la liste des groupes d'événements disponibles dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Service Événements > Services Événements > *service_événement* > Groupes d'événements**.

Configuration de Common Event Infrastructure à l'aide de la console d'administration

Il est préférable de configurer Common Event Infrastructure en utilisant uniquement la console d'administration.

A propos de cette tâche

Ouvrez le panneau de la console d'administration de Common Event Infrastructure Server :

Si vous configurez un serveur, sélectionnez **Serveurs > Serveurs d'application > nom_serveur > Business Integration > Common Event Infrastructure > Common Event Infrastructure - Serveur.**

Si vous configurez un cluster, cliquez sur **Serveurs > Clusters > nom_cluster > Business Integration > Common Event Infrastructure > Common Event Infrastructure - Serveur.**

Procédure

1. Activez le déploiement de l'application d'entreprise Common Event Infrastructure en sélectionnant la case intitulée **Activer le serveur d'infrastructure d'événement**. Si le serveur a déjà été configuré, vous pouvez l'activer ou le désactiver en cochant ou en décochant la case. Si l'option d'activation est désélectionnée, Common Event Infrastructure n'a pas été configuré ou bien a été désactivé dans une configuration antérieure et le serveur n'a pas été redémarré. Un message d'information indique si Common Event Infrastructure est configuré pour cette cible de déploiement. Si le serveur a déjà été configuré, vous pouvez changer les paramètres de la source de données pour la base de données de l'événement et/ou le stockage de messages.

Remarque : Si vous cochez la case pour activer le serveur Common Event Infrastructure et si celui-ci n'a pas encore été configuré, les paramètres affichés seront utilisés pour la configuration à moins que vous ne les modifiez.

- Si c'est la première fois que vous effectuez la configuration, les tables de source de données d'événements sont créées sur la base de données commune. S'il y a déjà une configuration du serveur Common Event Infrastructure, vous devez créer une nouvelle base de données.
- Le service de messagerie est créé sous un schéma unique dans la base de données commune.

Les modifications sont appliquées après le redémarrage du serveur ou du cluster sur lequel Common Event Infrastructure a été configuré.

2. Configurez (ou modifiez les paramètres actuels de la configuration existante de) la base de données d'événements en utilisant l'une des méthodes suivantes afin de remplir les champs avec les paramètres appropriés.
 - Cliquez sur **Édit** pour obtenir un panneau de configuration de la base de données avec une liste plus complète des options.
 - Utilisez les champs de cette fenêtre pour entrer les informations, tel que souligné ci-dessous :
 - a. **Instance de base de données** – Nom de la base de données que vous allez utiliser pour stocker des événements.
 - b. **Créer des tables** – Cochez cette case si vous voulez créer les tables de base de données dans la base de données des événements.

Remarque : Si vous configurez Common Event Infrastructure de façon à utiliser une base de données se trouvant sur un autre serveur, vous ne pouvez pas créer les tables à l'aide de cette option. Vous devez utiliser les scripts de la base de données qui seront générés une fois la configuration terminée. Dans ce cas, cliquez sur **Editer** afin d'afficher la fenêtre reprenant le détail de la source de données qui vous donnera l'emplacement des scripts de création de la base de données.

- c. **Nom d'utilisateur** et **Mot de passe** – pour donner l'accès à la base de données d'événements.
- d. **Serveur** – Nom du serveur sur lequel est installée la base de données d'événements.
- e. **Fournisseur** – Sélectionnez un fournisseur pour votre base de données à partir du menu.

Remarque : Le champ **Schéma** n'est activé que si la base de données est créée à l'aide de DB2 sur une plateforme iSeries ou z/OS. Dans tous les autres cas, le champ Schéma sera désactivé.

Important : Si les tables existent déjà dans la base de données cible, la configuration peut échouer.

- 3. Choisissez si le bus Common Event Infrastructure sera **Local** sur le serveur ou **Distant** et situé sur un autre serveur. Si vous choisissez une connexion à distance, sélectionnez l'emplacement distant à partir du menu ou cliquez sur **Nouveau** pour créer un nouveau bus distant.
- 4. Configurez le support de la messagerie de Common Event Infrastructure.
 - Cliquez sur Editer afin d'obtenir une fenêtre de configuration de base de données disposant d'une liste plus complète des options.
 - Utiliser les champs de cette fenêtre pour entrer les informations, tel qu'indiqué ci-dessous :
 - a. **Instance de base de données** – Saisissez le nom de la base de données que vous allez utiliser pour stocker vos messages.
 - b. **Schéma** – Saisissez le nom du schéma ou acceptez le nom donné par défaut.
 - c. **Nom d'utilisateur** et **Mot de passe** – Pour donner accès à la base de données de messagerie.
 - d. **Serveur** – Nom du serveur sur lequel est installée la base de données de messagerie.
 - e. **Fournisseur** – Choisissez un fournisseur pour votre base de données à partir du menu.
- 5. Créez un alias d'authentification de messagerie pour le bus Common Event Infrastructure.
 - a. Sélectionnez **Propriétés complémentaires > Alias d'authentification JMS**.
 - b. Saisissez l'ID utilisateur et le mot de passe que vous allez utiliser lors des communications sécurisées dans le Bus d'intégration système. Vous pouvez accepter les valeurs de CEI configurées par défaut pour l'ID utilisateur et le mot de passe si la sécurité est désactivée. Si la sécurité est activée, l'ID utilisateur et le mot de passe seront utilisés à des fins d'authentification du bus. Par conséquent, vous devez modifier l'ID utilisateur et le mot de passe pour sécuriser le système.
 - c. Cliquez sur **OK**.
- 6. Cliquez sur **OK** ou **Appliquer**.

7. Redémarrez le serveur ou le cluster.

Résultats

A ce stade, les principaux éléments de Common Event Infrastructure sont configurés et fonctionnent sur le serveur ou le cluster. Cela comprend le stockage des données d'événements, le moteur de messagerie et l'application d'événements. Ce panneau facilite la configuration de Common Event Infrastructure en évitant l'utilisation de nombreuses commandes et procédures manuelles.

Après le redémarrage de votre serveur ou cluster, vous serez capable de stocker des événements composants de service émis par vos applications. Vous pouvez maintenant modifier les propriétés d'exécution du serveur Common Event Infrastructure en sélectionnant la fenêtre **Common Event Infrastructure - Cible** . Vous pouvez choisir de lancer le serveur Common Event Infrastructure au démarrage et indiquer le nom JNDI de la fabrique d'émetteur où les événements seront envoyés.

Déploiement de l'application Common Event Infrastructure

Avant d'utiliser Common Event Infrastructure, vous devez d'abord déployer les services d'événements et les ressources associées dans l'environnement d'exécution du serveur.

A propos de cette tâche

L'application d'entreprise Common Event Infrastructure comprend les composants d'exécution du service Événements et la configuration de messagerie par défaut utilisés pour la soumission d'événements asynchrones.

Pour déployer le service Événements :

Procédure

A partir de l'outil wsadmin, exécutez la commande d'administration **deployEventService** en mode par lots ou interactif. Les paramètres de la commande d'administration **deployEventService** sont les suivants :

nodeName

Nom du noeud sur lequel déployer le service Événements. Ce paramètre est facultatif. Si vous ne spécifiez pas de nom pour le noeud, celui du noeud en cours sera attribué par défaut. Si vous indiquez un nom de noeud, vous devez également indiquer le nom du serveur, à l'aide du paramètre **serverName**. Ce paramètre n'est pas valide si vous déployez le service Événements dans un cluster.

serverName

Nom du serveur sur lequel déployer le service Événements. Ce paramètre est obligatoire uniquement si vous spécifiez un noeud. Il n'est pas valide si vous déployez le service Événements dans un cluster.

clusterName

Nom du cluster sur lequel déployer le service Événements. Ce paramètre est facultatif et ne doit pas être spécifié si vous procédez à un déploiement au niveau du noeud ou du serveur.

enable

Indique si le service Événements doit être démarré automatiquement au moment du démarrage du serveur. La valeur par défaut est true.

Résultats

Une fois la commande d'administration terminée, le service Événements Common Event Infrastructure et la configuration de la messagerie par défaut sont déployés au niveau spécifié.

Si la sécurité WebSphere est activée, vous devez également configurer l'alias et le mot de passe d'authentification JMS, à l'aide de la commande d'administration **setEventServiceJmsAuthAlias**.

Si vous déployez le service Événements dans un cluster, vous devez également configurer manuellement la base de données d'événements.

Référence associée

«**deployEventService**», à la page 113

«**setEventServiceJmsAuthAlias**», à la page 117

Déploiement de l'infrastructure CEI sur un cluster

Vous pouvez déployer les ressources Common Event Infrastructure de différentes manières dans un environnement de cluster.

Déploiement de l'infrastructure CEI sur un cluster existant

Vous pouvez déployer l'application du service Événements dans un cluster existant.

A propos de cette tâche

Le déploiement de l'application du service Événements dans un cluster est à peu près le même que sur un serveur autonome. Néanmoins, dans un environnement de cluster, aucune base de données d'événements n'est configurée.

Pour déployer et configurer Common Event Infrastructure dans un environnement de cluster :

Procédure

1. Exécutez la commande d'administration **deployEventService** de la même façon que vous le feriez pour un serveur autonome, mais en spécifiant le nom du cluster. Utilisez le paramètre `clusterName` pour spécifier le cluster.
2. Sur le système du gestionnaire de déploiement, exécutez la commande d'administration relative à la configuration de la base de données. Indiquez le nom du cluster à l'aide du paramètre `clusterName`. Cette commande permet de générer le script de configuration de la base de données.
3. Copiez le script de configuration de la base de données généré sur le système relatif à la base de données.
4. Exécutez le script de configuration de la base de données sur le système relatif à la base de données afin de créer la base de données d'événements.
5. Sur le système du gestionnaire de déploiement, exécutez la commande **enableEventService** pour activer le service Événements. Utilisez le paramètre `clusterName` pour indiquer le nom du cluster.

Création d'un cluster en convertissant un serveur Common Event Infrastructure existant

Vous pouvez créer un nouveau cluster en convertissant un serveur autonome existant déjà configuré avec Common Event Infrastructure.

Avant de commencer

Avant de convertir un serveur existant, vérifiez qu'il est bien configuré pour Common Event Infrastructure. Cela inclut le déploiement de l'application du service Evénements et la configuration de la base de données d'événements.

A propos de cette tâche

Pour créer le cluster :

Procédure

1. Suivez la procédure WebSphere habituelle permettant de convertir un serveur autonome en tant que membre principal d'un nouveau cluster. Une fois le serveur converti, la procédure suivante est appliquée :
 - Les ressources Common Event Infrastructure disponibles au niveau de la portée du serveur sont transférées vers la portée du nouveau cluster.

Base de données par défaut : Si le serveur existant est configuré avec la base de données Derby par défaut, les ressources de base de données ne sont pas déplacées vers la portée du cluster. Ces ressources sont supprimées. La configuration de la base de données par défaut n'est pas prise en charge dans un cluster. Dans ce cas, le service Evénements du cluster est désactivé par défaut.
 - La liste cible relative aux applications du service Evénements déployées est modifiée de façon à supprimer le serveur converti et à ajouter le nouveau cluster.
2. Facultatif : Si le serveur converti a été configuré avec la base de données Derby par défaut, vous devez configurer une nouvelle base de données d'événements puis activer le service Evénements :
 - a. Sur le système du gestionnaire de déploiement, exécutez la commande d'administration relative à la configuration de la base de données. Indiquez le nom du cluster à l'aide du paramètre `clusterName`. Cette commande permet de générer le script de configuration de la base de données.
 - b. Copiez le script de configuration de la base de données généré sur le système relatif à la base de données.
 - c. Exécutez le script de configuration de la base de données sur le système relatif à la base de données afin de créer la base de données d'événements.
 - d. Sur le système du gestionnaire de déploiement, exécutez la commande **enableEventService** pour activer le service Evénements. Utilisez le paramètre `clusterName` pour indiquer le nom du cluster.

Création d'un cluster à l'aide d'un serveur Common Event Infrastructure existant en tant que modèle

Vous pouvez créer un nouveau cluster en définissant un serveur Common Event Infrastructure comme modèle.

Avant de commencer

Avant de créer un cluster à l'aide de cette méthode, vous devez disposer d'un serveur existant entièrement configuré pour Common Event Infrastructure. Cela inclut le déploiement de l'application du service Événements et la configuration de la base de données d'événements.

A propos de cette tâche

Pour créer le cluster :

Procédure

1. Suivez la procédure WebSphere permettant de créer un nouveau cluster en utilisant le serveur Common Event Infrastructure existant en tant que modèle pour le premier membre du cluster. Une fois le premier membre créé, la procédure suivante est appliquée :

- Les ressources Common Event Infrastructure disponibles au niveau de la portée du serveur existant sont copiées sur la portée du nouveau cluster.

Base de données par défaut : Si le serveur existant est configuré avec la base de données Derby par défaut, les ressources de base de données ne sont pas copiées sur la portée du cluster. La configuration de la base de données par défaut n'est pas prise en charge dans un cluster. Dans ce cas, le service Événements du cluster est désactivé par défaut.

- La liste cible relative aux applications du service Événements déployées est modifiée de façon à inclure le nouveau cluster.
2. Facultatif : Si le serveur existant a été configuré avec la base de données Derby par défaut, vous devez configurer une nouvelle base de données d'événements pour le cluster puis activer le service Événements :
- a. Sur le système du gestionnaire de déploiement, exécutez la commande d'administration relative à la configuration de la base de données. Indiquez le nom du cluster à l'aide du paramètre `clusterName`. Cette commande permet de générer le script de configuration de la base de données.
 - b. Copiez le script de configuration de la base de données généré sur le système relatif à la base de données.
 - c. Exécutez le script de configuration de la base de données sur le système relatif à la base de données afin de créer la base de données d'événements.
 - d. Sur le système du gestionnaire de déploiement, exécutez la commande **enableEventService** pour activer le service Événements. Utilisez le paramètre `clusterName` pour indiquer le nom du cluster.

Configuration de la messagerie d'événements

Vous pouvez modifier la configuration de messagerie utilisée pour la transmission JMS d'événements au service événements.

A propos de cette tâche

L'infrastructure de messagerie de Common Event Infrastructure est créée lorsque vous utilisez la console d'administration pour configurer Common Event Infrastructure sur un serveur. En règle générale, la configuration de messagerie utilise le fournisseur de messagerie par défaut et crée une file d'attente JMS unique pour la transmission asynchrone d'événements au service d'événements. Le cas échéant, vous pouvez modifier la configuration de la messagerie.

Configuration de files d'attente JMS supplémentaires

Si vous utilisez la configuration de messagerie d'événements par défaut, vous pouvez ajouter des files d'attente JMS supplémentaires pour la transmission d'événements au service Événements.

A propos de cette tâche

Pour des files d'attente JMS supplémentaires à l'aide de la configuration de messagerie par défaut, vous pouvez définir plusieurs files d'attente JMS acheminées à une destination de file d'attente de bus d'intégration de services. La destination de la file d'attente Common Event Infrastructure bus d'intégration de services dépend de la portée à laquelle le service Événements est déployé.

Portée	Destination de file d'attente de bus d'intégration de services
Serveur	<i>noeud.serveur.CommonEventInfrastructureQueueDestination</i>
Cluster	<i>cluster.CommonEventInfrastructureQueueDestination</i>

Pour plus d'informations sur la configuration bus d'intégration de services, voir la documentation.

Configuration de la messagerie d'événements à l'aide d'un fournisseur JMS externe

Si vous ne souhaitez pas utiliser la configuration de messagerie intégrée par défaut pour la transmission asynchrone des événements, vous pouvez configurer l'utilisation d'un fournisseur JMS (Java Messaging Service) externe.

Avant de commencer

Avant de configurer la messagerie d'événements pour l'utilisation d'un fournisseur JMS externe, vous devez créer une file d'attente et une fabrique de connexions JMS à l'aide des interfaces appropriées pour votre fournisseur JMS. Vous devez également créer un port d'écoute ou une spécification d'activation.

A propos de cette tâche

Pour configurer la messagerie d'événements à l'aide d'un fournisseur JMS externe :

Procédure

A partir de l'outil wsadmin, exécutez la commande d'administration **deployEventServiceMdb** en mode par lots ou interactif. Les paramètres de la commande **deployEventServiceMdb** sont les suivants :

applicationName

Nom de l'application du bean géré par message du service Événements à déployer. Ce paramètre est obligatoire.

nodeName

Nom du noeud sur lequel déployer le bean géré par message du service Événements. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Ce paramètre est facultatif. Par défaut, le nom du noeud en cours est utilisé. N'indiquez pas ce paramètre si vous déployez l'application dans un cluster.

serverName

Nom du serveur sur lequel déployer le bean géré par message du service Événements. Ce paramètre est obligatoire si vous déployez l'application au niveau du serveur, sinon il est facultatif. Ne spécifiez aucun nom de serveur si vous déployez l'application dans un cluster.

clusterName

Nom du cluster sur lequel déployer le bean géré par message du service Événements. Spécifiez ce paramètre uniquement si vous déployez l'application dans un cluster.

listenerPort

Nom du port d'écoute que le bean géré par message du service Événements doit utiliser pour publier des événements. Le port d'écoute spécifié doit déjà exister. Vous devez indiquer soit un port d'écoute, soit une spécification d'activation, mais pas les deux.

activationSpec

Nom JNDI de la spécification d'activation que le bean géré par message du service Événements doit utiliser pour publier des événements. La spécification d'activation doit déjà exister. Vous devez indiquer soit un port d'écoute, soit une spécification d'activation, mais pas les deux.

qcfJndiName

Nom JNDI de la fabrique de connexions de file d'attente JMS que le bean géré par message du service Evénements doit utiliser. Ce paramètre est obligatoire si vous indiquez une spécification d'activation, sinon il est facultatif. Si vous spécifiez une fabrique de connexions de file d'attente et un port d'écoute, la fabrique doit correspondre à celle qui est configurée pour le port d'écoute.

Résultats

La commande d'administration **deployEventServiceMdb** permet de déployer le bean géré par message pour le service Evénements, configuré pour le port d'écoute ou la spécification d'activation indiqué(e). Elle permet également de créer une fabrique d'émetteurs et une transmission JMS à l'aide d'une configuration JMS externe. Les applications peuvent utiliser soit la fabrique d'émetteurs par défaut (configurée pour utiliser la configuration de messagerie par défaut), soit la nouvelle fabrique (qui utilise le fournisseur JMS externe).

Que faire ensuite

Pour définir plusieurs files d'attente JMS sur le service Evénements, vous pouvez exécuter cette commande plusieurs fois en indiquant des noms d'applications d'entreprise et de files d'attente JMS différents. Chaque fois que vous exécutez le script, il déploie un bean géré par message supplémentaire et configure de nouvelles ressources pour l'utilisation de la file d'attente JMS indiquée.

Configuration de l'alias d'authentification JMS

Si la sécurité WebSphere est activée et que vous souhaitez utiliser des messages JMS asynchrones pour soumettre des événements au service Événements, vous devez configurer l'alias d'authentification JMS.

A propos de cette tâche

Pour configurer l'alias d'authentification JMS :

Procédure

A partir de l'outil wsadmin, exécutez la commande d'administration **setEventServiceJmsAuthAlias** en mode par lots ou interactif. Les paramètres de la commande **setEventServiceJmsAuthAlias** sont les suivants :

userName

Nom de l'utilisateur à utiliser pour l'alias d'authentification JMS. Ce paramètre est obligatoire.

password

Mot de passe de l'utilisateur à utiliser pour l'alias d'authentification JMS. Ce paramètre est obligatoire.

nodeName

Nom du noeud sur lequel vous souhaitez mettre à jour ou créer l'alias d'authentification JMS. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Ne spécifiez aucun nom de noeud si vous configurez l'alias d'authentification dans un cluster.

serverName

Nom du serveur sur lequel vous souhaitez mettre à jour ou créer l'alias d'authentification JMS. Ce paramètre est obligatoire uniquement si vous spécifiez un noeud. Il n'est pas valide si vous configurez l'alias d'authentification dans un cluster.

clusterName

Nom du cluster sur lequel vous souhaitez mettre à jour ou créer l'alias d'authentification JMS. Indiquez ce paramètre uniquement si vous configurez l'alias d'authentification dans un cluster. Si vous spécifiez un nom de cluster, ne spécifiez aucun nom de noeud ni de serveur.

Résultats

L'alias d'authentification JMS utilisé par les objets du service Événements est mis à jour au niveau indiqué. Si l'authentification n'existe pas, elle est créée à l'aide des valeurs spécifiées.

Configuration de la base de données d'événements

Vous pouvez configurer la source de données d'événements à l'aide de commandes qui sont spécifiques à chaque produit de base de données pris en charge.

A propos de cette tâche

La base de données d'événements est indispensable à la prise en charge des événements persistants. Si vous n'avez pas utilisé le panneau de configuration Common Event Infrastructure dans la console d'administration, vous avez toujours la possibilité de créer la base de données d'événements à l'aide des commandes décrites ici.

Restrictions relatives à la base de données d'événements

Certaines restrictions s'appliquent aux configurations de la base de données d'événements à l'aide de certains logiciels de base de données.

Reportez-vous au tableau suivant pour consulter la liste des restrictions pouvant s'appliquer à votre environnement.

Tableau 1. Restrictions relatives à la base de données d'événements

Type de base de données	Restrictions
Oracle	<ul style="list-style-type: none">• Si vous utilisez un jeu de caractères Unicode, vous devez respecter les contraintes imposées par le pilote JDBC Thin d'Oracle 10g en ce qui concerne la longueur des valeurs de chaîne. Une erreur Oracle ORA-01461 peut se produire lorsque des événements contenant des valeurs longues (attributs de message, etc.) sont stockés dans la base de données d'événements. Pour plus de détails sur ces restrictions, reportez-vous à la documentation d'Oracle 10g. Afin d'éviter ce type d'erreur, utilisez le pilote OCI d'Oracle 10g ou le pilote Thin d'Oracle 9i.• Sous Oracle Database, une chaîne vide équivaut à une valeur NULL. Si vous indiquez une chaîne vide comme valeur d'un attribut d'événement, elle sera convertie en valeur NULL lors de son stockage dans la base de données d'événements oracle.

Tableau 1. Restrictions relatives à la base de données d'événements (suite)

Type de base de données	Restrictions
Informix	<ul style="list-style-type: none"> Le pilote JDBC 3.0 (ou version ultérieure) est requis. Les versions antérieures de ce pilote n'offraient pas une prise en charge complète des transactions XA requises. Les scripts de configuration de base de données et de suppression générés par la commande d'administration configEventServiceInformixDB requièrent la commande dbaccess pour exécuter des scripts SQL. Il se peut que cette commande soit uniquement disponible sur le serveur Informix. Par conséquent, si le serveur Informix se trouve sur un système autre que celui du serveur WebSphere, les scripts de configuration de base de données devront être copiés sur le serveur Informix et s'exécuter localement.
SQL Server	<ul style="list-style-type: none"> La base de données SQL Server doit être configurée en vue d'utiliser le mode d'authentification mixte. Les connexions sécurisées ne sont pas prises en charge. Les procédures stockées XA doivent être installées. Ces procédures stockées sont fournies avec le pilote JDBC depuis Microsoft Corporation. Le fichier sqljdbc.dll doit figurer dans un répertoire spécifié dans l'instruction PATH. Ce fichier est fourni avec le pilote JDBC depuis Microsoft Corporation. Le service DTC (Distributed Transaction Coordinator) doit être démarré.

Configuration d'une base de données d'événements Derby

Vous pouvez configurer une base de données d'événements Derby au niveau de la portée serveur ou cluster sur un système Linux, UNIX ou Windows.

A propos de cette tâche

Il existe deux types de bases de données Derby exploitables pour les bases de données d'événements : Derby Embedded et Derby Network. Les deux sont fournies avec WebSphere Application Server, mais disposent de fonctionnalités limitées inadaptées à un environnement de production. C'est pourquoi il est préférable de n'utiliser Derby comme base de données d'événements que dans un but de test et de développement. Pour plus d'informations sur les bases de données Derby, voir la documentation de WebSphere Application Server (lien figurant en bas de page).

Derby Embedded ne peut être utilisé qu'avec un serveur autonome. Par conséquent, si vous deviez fédérer votre serveur autonome à un cluster ou à un environnement ND, il faut reconfigurer totalement votre source de données d'événements à l'aide d'un autre produit de base de données. Il démarrera automatiquement au démarrage du serveur.

Derby Network peut-être utilisé dans un environnement de cluster ou ND, cependant, il est également préférable d'éviter son utilisation sur des systèmes de production réels. Vous devez démarrer la base de données manuellement lors de son utilisation avec un serveur.

Pour configurer une base de données d'événements Derby :

Procédure

1. Démarrez l'outil wsadmin.
2. L'objet AdminTask permet d'exécuter la commande d'administration **configEventServiceDerbyDB** en mode de traitement par lots ou en mode interactif. Les paramètres minimaux requis pour la commande **configEventServerDerbyDB** sont les suivants :

createDB

Indique si la commande d'administration doit créer et exécuter les scripts de configuration de la base de données. Spécifiez la valeur `true` ou `false`. Si ce paramètre est défini sur `false`, les scripts sont créés mais ne sont pas exécutés. Vous devez ensuite exécuter les scripts de configuration de la base de données pour terminer la configuration de la base de données.

nodeName

Nom du noeud contenant le serveur sur lequel la source de données du service Evénements doit être créé. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Vous devez indiquer l'un des éléments suivants :

- Nom du noeud et nom du serveur
- Nom du cluster

NomServeur

Nom du serveur sur lequel la source de données du service Evénements doit être créée.

clusterName

Nom du cluster sur lequel la source de données du service Evénements doit être créée. Si vous indiquez un nom de cluster, n'indiquez pas de noms de noeud et de serveur.

D'autres paramètres peuvent être requis pour votre environnement. Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, reportez-vous à l'aide de la commande d'administration **configEventServiceDerbyDB**.

Résultats

La commande d'administration crée la source de données requise au niveau de la portée spécifiée ; si vous avez indiqué la valeur `true` pour le paramètre `createDB`, la commande exécute également le script de configuration de base de données généré pour créer la base de données.

Les scripts de configuration de base de données générés sont stockés par défaut dans le répertoire `racine_profil/databases/event/noeud/serveur/dbscripts/derby`. (Dans un environnement de déploiement réseau, ces scripts sont stockés dans le répertoire des profils du gestionnaire de déploiement.) Si vous avez indiqué une valeur pour le paramètre `outputScriptDir` facultatif, les scripts sont stockés à cet emplacement à la place. Vous pouvez utiliser ces scripts pour configurer manuellement la base de données d'événements à tout moment.

Configuration d'une base de données d'événements DB2 (Linux, UNIX et Windows)

Vous pouvez configurer une base de données d'événements externe à l'aide de DB2 Universal Database sur un système Linux, UNIX ou Windows.

A propos de cette tâche

Pour configurer une base de données d'événements DB2 sur un système Linux, UNIX ou Windows :

Procédure

1. Démarrez l'outil wsadmin.
2. L'objet AdminTask permet d'exécuter la commande d'administration **configEventServiceDB2DB** en mode de traitement par lots ou interactif. Les paramètres minimaux requis pour la commande **configEventServiceDB2DB** sont les suivants :

createDB

Indique si la commande d'administration doit créer et exécuter les scripts de configuration de la base de données. Spécifiez la valeur true ou false. Si ce paramètre est défini sur false, les scripts sont créés mais ne sont pas exécutés. Vous devez ensuite exécuter les scripts de configuration de la base de données pour terminer la configuration de la base de données.

nodeName

Nom du noeud contenant le serveur sur lequel la source de données du service Evénements doit être créé. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Vous devez indiquer l'un des éléments suivants :

- Nom du noeud et nom du serveur
- Nom du cluster

NomServeur

Nom du serveur sur lequel la source de données du service Evénements doit être créée.

clusterName

Nom du cluster sur lequel la source de données du service Evénements doit être créée. Si vous indiquez un nom de cluster, n'indiquez pas de nom de noeud ou de serveur.

jdbcClassPath

Chemin d'accès du pilote JDBC. Indiquez uniquement le chemin d'accès au fichier de pilote ; ne spécifiez pas le nom du fichier.

dbHostName

Nom hôte du serveur sur lequel est installée la base de données.

dbUser

ID utilisateur DB2 à utiliser lors de la création de la base de données d'événements. L'ID utilisateur indiqué doit disposer de privilèges permettant de créer et de supprimer des bases de données.

dbPassword

Mot de passe DB2 à utiliser.

D'autres paramètres peuvent être requis pour votre environnement. Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, reportez-vous à l'aide de la commande d'administration **configEventServiceDB2DB**.

Résultats

La commande d'administration crée la source de données requise au niveau de la portée spécifiée ; si vous avez indiqué la valeur `true` pour le paramètre `createDB`, la commande exécute également le script de configuration de base de données généré pour créer la base de données.

Les scripts de configuration de base de données générés sont stockés par défaut dans le répertoire `racine_profil/databases/event/noeud/serveur/dbscripts/db2` directory. (Dans un environnement de déploiement réseau, ces scripts sont stockés dans le répertoire des profils du gestionnaire de déploiement.) Si vous avez indiqué une valeur pour le paramètre `outputScriptDir` facultatif, les scripts sont stockés à cet emplacement à la place. Vous pouvez utiliser ces scripts pour configurer manuellement la base de données d'événements à tout moment.

Configuration d'une base de données DB2 sur un système z/OS

Vous pouvez configurer une base de données d'événements sur un système z/OS à l'aide du logiciel de base de données DB2.

Avant de commencer

Pour configurer la base de données DB2 à partir d'un client distant, vous devez auparavant avoir installé DB2 Connect avec les fixpacks les plus récents.

A propos de cette tâche

Pour configurer la base de données d'événements :

Procédure

- Linux UNIX Windows Si vous configurez la base de données d'événements z/OS à partir d'un système client Linux, UNIX, ou Windows, procédez comme suit pour créer et classer la base de données :
 - Sur le système z/OS, utilisez le menu d'administration DB2 pour créer un sous-système.
 - Facultatif : Créez le groupe de stockage à utiliser avec la base de données d'événements. Vous pouvez également utiliser un groupe de stockage existant (par exemple, `sysdeflt`).
 - Activez les pools de mémoire tampon de 4 Ko, 8 Ko et 16 Ko à utiliser avec le base de données d'événements.
 - Accordez les droits d'accès appropriés à l'ID utilisateur que la source de données doit utiliser. Cet ID doit également pouvoir accéder à la base de données et au groupe de stockage créés et être autorisé à créer des tables, des espaces tables et des index pour la base de données.
 - Cataloguez la base de données distante. Exécutez les commandes suivantes dans un script ou dans une fenêtre de ligne de commande DB2 :

```
catalog tcpip node zosnode remote nom_hôte server port_IP system
sous-système_base_de_données
catalog database nom_base_de_données as nom_base_de_données
at node zosnode authentication DCS
```

Pour plus d'informations sur le catalogage des noeuds et des bases de données, reportez-vous à la documentation relative à la documentation DB2 Connect.
 - Vérifiez que vous pouvez établir une connexion avec le sous-système distant. Pour ce faire, vous pouvez exécuter la commande suivante :

```
db2 connect to
sous-système user id_utilisateur using
mot_de_passe
```
 - Effectuez une liaison avec la base de données hôte. Exécutez les commandes suivantes :

```
db2 connect to nom_base_de_données user id_utilisateur using mot_de_passe
db2 bind db2_root/bnd/@ddcsmvs.lst blocking all sqlerror continue message
mvs.msg grant public
db2 connect reset
```

Pour plus d'informations sur la liaison d'un client à une base de données hôte, reportez-vous à la documentation relative à DB2 Connect.
- Sur le système WebSphere, lancez l'outil wsadmin.

3. L'objet AdminTask permet d'exécuter la commande d'administration **configEventServiceDB2ZOSDB** en mode de traitement par lots ou interactif. Les paramètres minimaux requis pour la commande **configEventServiceDB2ZOSDB** sont les suivants :

createDB

Linux **UNIX** **Windows** Indique si la commande d'administration doit créer et exécuter les scripts de configuration de la base de données. Ce paramètre s'applique uniquement si vous exécutez la commande d'administration à partir d'un système client Linux, UNIX ou Windows. Spécifiez la valeur true ou false.

Si ce paramètre est défini sur false ou si vous exécutez la commande sur le système z/OS, les scripts sont créés mais ne sont pas exécutés. Vous devez ensuite exécuter les scripts de configuration de la base de données pour terminer la configuration de la base de données.

nodeName

Nom du noeud contenant le serveur sur lequel la source de données du service Evénements doit être créé. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Vous devez indiquer l'un des éléments suivants :

- Nom du noeud et nom du serveur
- Nom du cluster

NomServeur

Nom du serveur sur lequel la source de données du service Evénements doit être créée.

clusterName

Nom du cluster sur lequel la source de données du service Evénements doit être créée. Si vous indiquez un nom de cluster, n'indiquez pas de noms de noeud et de serveur.

jdbcClassPath

Chemin d'accès du pilote JDBC. Indiquez uniquement le chemin d'accès au fichier de pilote ; ne spécifiez pas le nom du fichier.

dbHostName

Nom hôte du serveur sur lequel est installé la base de données.

dbUser

ID utilisateur DB2 à utiliser lors de la création de la base de données d'événements. L'ID utilisateur indiqué doit disposer de privilèges permettant de créer et de supprimer des bases de données.

dbPassword

Mot de passe DB2 à utiliser.

D'autres paramètres peuvent être requis pour votre environnement. Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, reportez-vous à l'aide de la commande d'administration **configEventServiceDB2ZOSDB**.

Résultats

La commande d'administration crée la source de données requise au niveau de la portée spécifiée ; si vous exécutez la commande sur un client DB2 Linux, UNIX ou Windows et que vous avez indiqué la valeur true pour le paramètre createDB, la commande exécute également le script de configuration de base de données généré pour créer la base de données. Sur un système z/OS, vous devez utiliser SPUI (SQL Processor Using File Input) pour exécuter les fichiers DDL générés. Les fichiers DDL sont stockés dans le répertoire *racine_profil/databases/event/noeud/serveur/db2zos/ddl*.

Les scripts de configuration de base de données générés sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/db2zos*. (Dans un environnement de déploiement réseau, ces scripts sont stockés dans le répertoire des profils du gestionnaire de déploiement.) Si vous avez indiqué une valeur pour le paramètre outputScriptDir facultatif, les scripts sont stockés à cet emplacement à la place. Vous pouvez utiliser ces scripts pour configurer manuellement la base de données d'événements à tout moment.

Que faire ensuite

Une fois la base de données configurée, vous pouvez utiliser la console d'administration pour tester la configuration de la base de données. Pour ce faire, accédez à la source de données JDBC appropriée et sélectionnez l'option **Tester la connexion**.

Configuration d'une base de données DB2 sur un système iSeries

Vous pouvez configurer une base de données d'événements sur un système iSeries en utilisant un logiciel de base de données DB2 .

A propos de cette tâche

Si vous utilisez un serveur iSeries local pour configurer un serveur iSeries distant, vous devez spécifier une entrée de base de données distante sur le serveur local en tant qu'alias à la base de données cible. Pour configurer la base de données d'événements :

Procédure

1. Démarrez l'outil wsadmin.
2. L'objet AdminTask permet d'exécuter la commande d'administration **configEventServiceDB2iSeriesDB** en mode de traitement par lots ou en mode interactif. Les paramètres minimaux requis pour la commande **configEventServiceDB2iSeriesDB** sont les suivants :

createDB

Indique si la commande d'administration doit créer et exécuter les scripts de configuration de la base de données. Spécifiez la valeur true ou false. Si ce paramètre est défini sur false, les scripts sont créés mais ne sont pas exécutés. Vous devez ensuite exécuter les scripts de configuration de la base de données pour terminer la configuration de la base de données.

Restriction : La commande d'administration ne peut exécuter automatiquement le script de configuration de la base de données que sur un système iSeries. Si vous exécutez la commande sur un système client, une erreur apparaîtra.

nodeName

Nom du noeud contenant le serveur sur lequel la source de données du service Evénements doit être créé. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Vous devez indiquer l'un des éléments suivants :

- Nom du noeud et nom du serveur
- Nom du cluster

serverName

Nom du serveur sur lequel la source de données du service Evénements doit être créée.

clusterName

Nom du cluster sur lequel la source de données du service Evénements doit être créée. Si vous indiquez un nom de cluster, n'indiquez pas de nom de noeud ou de serveur.

toolboxJdbcClassPath

Chemin d'accès au pilote JDBC DB2 d'IBM Toolbox for Java. Utilisez ce paramètre uniquement si vous souhaitez utiliser le pilote Toolbox for Java plutôt que le pilote JDBC natif. Indiquez uniquement le chemin d'accès au fichier de pilote ; ne spécifiez pas le nom du fichier.

nativeJdbcClassPath

Le chemin d'accès au pilote JDBC natif de DB2 for iSeries. Utilisez ce paramètre uniquement si vous souhaitez utiliser le pilote JDBC natif plutôt que le pilote Toolbox Java. Indiquez uniquement le chemin d'accès au fichier de pilote ; ne spécifiez pas le nom du fichier.

dbHostName

Nom hôte du serveur sur lequel est installée la base de données. Ce paramètre est requis si vous utilisez le pilote JDBC Toolbox for Java.

dbUser

ID utilisateur DB2 à utiliser lors de la création de la base de données d'événements. L'ID utilisateur indiqué doit disposer de privilèges permettant de créer et de supprimer des bases de données.

dbPassword

Mot de passe DB2 à utiliser.

D'autres paramètres peuvent être requis pour votre environnement. Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, reportez-vous à l'aide de la commande d'administration **configEventServiceDB2iSeriesDB**.

Résultats

Cette commande d'administration génère des scripts permettant de créer la base de données et les sources de données requises au niveau de la portée spécifique. Ces scripts sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/db2iseries*. Si vous avez indiqué une valeur pour le paramètre `outputScriptDir` facultatif, les scripts sont stockés à cet emplacement à la place. Vous pouvez utiliser ces scripts pour configurer manuellement la base de données d'événements à tout moment.

Si vous avez exécuté la commande d'administration de configuration de la base de données sur le système client, vous devez transférer les scripts générés au système iSeries puis les exécuter pour créer les ressources requises.

Une fois la base de données configurée, vous pouvez utiliser la console d'administration pour tester la configuration de la base de données. Pour ce faire, accédez à la source de données JDBC appropriée et sélectionnez l'option **Tester la connexion**.

Configuration d'une base de données d'événements Informix

Vous pouvez configurer une base de données d'événements externes IBM Informix Dynamic Server sur un système Linux, UNIX ou Windows.

A propos de cette tâche

Pour configurer la base de données d'événements Informix :

Procédure

1. Démarrez l'outil wsadmin.
2. L'objet AdminTask permet d'exécuter la commande d'administration **configEventServiceInformixDB** en mode par lots ou interactif. Les paramètres minimaux requis pour la commande **configEventServiceInformixDB** sont les suivants :

createDB

Indique si la commande d'administration doit créer et exécuter les scripts de configuration de la base de données. Spécifiez la valeur `true` ou `false`. Si ce paramètre est défini sur `false`, les scripts sont créés mais ne sont pas exécutés. Vous devez ensuite exécuter les scripts de configuration de la base de données pour terminer la configuration de la base de données.

Privilèges : Si vous spécifiez `true` pour ce paramètre, vérifiez que votre ID utilisateur dispose des privilèges suffisants pour créer des bases de données, des espaces de base de données, des tables, des vues, des index et des procédures stockées Informix.

NomNoeud

Nom du noeud contenant le serveur sur lequel la source de données du service d'événements doit être créée. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Vous devez indiquer l'un des éléments suivants :

- Nom du noeud et nom du serveur
- Nom du cluster

NomServeur

Nom du serveur sur lequel la source de données du service d'événements doit être créée.

Nomcluster

Nom du cluster sur lequel la source de données du service d'événements doit être créée. Si vous indiquez un nom de cluster, n'indiquez pas de noms de noeud et de serveur.

cheminClassejdbc

Chemin d'accès au pilote JDBC. Indiquez uniquement le chemin d'accès au fichier de pilote ; ne spécifiez pas le nom du fichier.

RépInformixbd

Répertoire dans lequel est installé le logiciel de base de données Informix. Ce paramètre est requis uniquement si vous avez spécifié true pour le paramètre createDB.

NomHôtebd

Nom d'hôte du système sur lequel le serveur de base de données est installé.

NomServeurBD

Le nom du serveur Informix (par exemple, ol_servername).

Utilisateurbd

ID utilisateur du schéma de base de données Informix qui détiendra les tables de base de données d'événements. Il doit s'agir d'un ID utilisateur ayant les privilèges requis pour créer des bases de données et des espaces de base de données. La source de données WebSphere utilise cet ID utilisateur pour authentifier la connexion à la base de données Informix.

MotdePassebd

Mot de passe de l'ID utilisateur du schéma spécifié.

D'autres paramètres peuvent être requis pour votre environnement. Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, reportez-vous à l'aide de la commande d'administration **configEventServiceInformixDB**.

Résultats

La commande d'administration crée la source de données requise au niveau de la portée spécifiée ; si vous avez indiqué la valeur true pour le paramètre createDB, la commande exécute également le script de configuration de base de données généré pour créer la base de données.

Les scripts de configuration de base de données générés sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/informix*. (Dans un environnement de déploiement réseau, ces scripts sont stockés dans le répertoire des profils du gestionnaire de déploiement.) Si vous avez indiqué une valeur pour le paramètre outputScriptDir facultatif, les scripts sont stockés à cet emplacement à la place. Vous pouvez utiliser ces scripts pour configurer manuellement la base de données d'événements à tout moment.

Exécution des scripts : Les scripts de configuration de base de données et de suppression générés par la commande d'administration **configEventServiceInformixDB** requièrent la commande **dbaccess** pour exécuter des scripts SQL. Il se peut que cette commande soit uniquement disponible sur le serveur Informix. Par conséquent, si le serveur Informix est sur un système différent du serveur, les scripts de configuration de la base de données peuvent ne pas être copiés sur le serveur Informix et tourner localement.

Configuration d'une base de données d'événements Oracle

Vous pouvez configurer une base de données d'événements externe à l'aide d'Oracle Database sur un système Linux, UNIX ou Windows.

Avant de commencer

Avant de configurer une base de données d'événements Oracle, vous devez créer la base de données. Le SID Oracle doit déjà exister lorsque vous exécutez la commande de configuration de la base de données d'événements. La valeur SID par défaut pour la base de données d'événements est événement.

A propos de cette tâche

Pour configurer une base de données d'événements Oracle :

Procédure

1. Démarrez l'outil wsadmin.
2. L'objet AdminTask permet d'exécuter la commande d'administration **configEventServiceOracleDB** en mode interactif ou de traitement par lots. Les paramètres minimaux requis pour la commande **configEventServiceOracleDB** sont les suivants :

createDB

Indique si la commande d'administration doit créer et exécuter les scripts de configuration de la base de données. Spécifiez la valeur `true` ou `false`. Si ce paramètre est défini sur `false`, les scripts sont créés mais ne sont pas exécutés. Vous devez ensuite exécuter les scripts de configuration de la base de données pour terminer la configuration de la base de données.

nodeName

Nom du noeud contenant le serveur sur lequel la source de données du service Evénements doit être créé. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Vous devez indiquer l'un des éléments suivants :

- Nom du noeud et nom du serveur
- Nom du cluster

serverName

Nom du serveur sur lequel la source de données du service Evénements doit être créée.

clusterName

Nom du cluster sur lequel la source de données du service Evénements doit être créée. Si vous indiquez un nom de cluster, n'indiquez pas de nom de noeud ou de serveur.

jdbcClassPath

Chemin d'accès du pilote JDBC. Indiquez uniquement le chemin d'accès au fichier de pilote ; ne spécifiez pas le nom du fichier.

oracleHome

Répertoire ORACLE_HOME. Ce paramètre est requis uniquement si vous avez spécifié `true` pour le paramètre `createDB`.

dbPassword

Mot de passe à utiliser pour l'ID utilisateur du schéma créé lors de la configuration de la base de données (l'ID utilisateur par défaut est `ceiuser`). Ce mot de passe est utilisé pour l'authentification de la connexion à la base de données Oracle.

sysUser

ID utilisateur SYSUSER Oracle. Cet ID utilisateur doit disposer de privilèges SYSDBA.

sysPassword

Mot de passe relatif à l'ID utilisateur SYSUSER spécifié.

D'autres paramètres peuvent être requis pour votre environnement. Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, reportez-vous à l'aide de la commande d'administration **configEventServiceOracleDB**.

Résultats

La commande d'administration crée la source de données requise au niveau de la portée spécifiée ; si vous avez indiqué la valeur `true` pour le paramètre `createDB`, la commande exécute également le script de configuration de base de données généré pour créer la base de données.

Les scripts de configuration de base de données générés sont stockés par défaut dans le répertoire `racine_profil/databases/event/noeud/serveur/dbscripts/oracle`. (Dans un environnement de déploiement réseau, ces scripts sont stockés dans le répertoire des profils du gestionnaire de déploiement.) Si vous avez indiqué une valeur pour le paramètre `outputScriptDir` facultatif, les scripts sont stockés à cet emplacement à la place. Vous pouvez utiliser ces scripts pour configurer manuellement la base de données d'événements à tout moment.

Configuration d'une base de données d'événements SQL Server

Vous pouvez configurer une base de données d'événements externe à l'aide de Microsoft SQL Server Enterprise sur un système Windows.

A propos de cette tâche

Pour configurer une base de données d'événements SQL Server :

Procédure

1. Sur le système du serveur de base de données SQL Server, créez le répertoire utilisé pour contenir les fichiers de base de données. Par défaut, les fichiers sont écrits dans le répertoire `c:\program files\ibm\event\ceiinst1\sqlserver_data`. Si vous devez spécifier un emplacement différent, vous devez modifier le script de configuration de base de données généré afin de modifier la valeur du paramètre `ceiInstancePrefix`, puis exécuter le script manuellement.
2. Sur le système serveur, lancez l'outil `wsadmin`.
3. L'objet `AdminTask` permet d'exécuter la commande d'administration `configEventServiceSQLServerDB` en mode interactif ou de traitement par lots. Les paramètres minimaux requis pour la commande `configEventServiceSQLServerDB` sont les suivants :

createDB

Indique si la commande d'administration doit créer et exécuter les scripts de configuration de la base de données. Spécifiez la valeur `true` ou `false`. Si ce paramètre est défini sur `false`, les scripts sont créés mais ne sont pas exécutés. Vous devez ensuite exécuter les scripts de configuration de la base de données pour terminer la configuration de la base de données.

nodeName

Nom du noeud contenant le serveur sur lequel la source de données du service Evénements doit être créé. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Vous devez indiquer l'un des éléments suivants :

- Nom du noeud et nom du serveur
- Nom du cluster

serverName

Nom du serveur sur lequel la source de données du service Evénements doit être créée. Si vous indiquez un nom de serveur, vous devez également indiquer un nom de noeud.

clusterName

Nom du cluster sur lequel la source de données du service Evénements doit être créée. Si vous indiquez un nom de cluster, n'indiquez pas de noms de noeud et de serveur.

dbServerName

Nom du serveur de la base de données SQL Server. Ce paramètre est requis uniquement si vous avez spécifié `true` pour le paramètre `createDB`.

dbHostName

Nom d'hôte du serveur sur lequel la base de données SQL Server s'exécute.

dbPassword

Mot de passe à utiliser pour l'ID utilisateur créé en tant que propriétaire des tables de base de données (l'ID utilisateur par défaut est ceiuser). La source de données WebSphere utilise ce mot de passe pour authentifier la connexion à la base de données SQL Server.

saUser

ID utilisateur disposant des privilèges adéquats pour créer et supprimer les bases de données et les utilisateurs. Ce paramètre est requis uniquement si vous avez spécifié true pour le paramètre createDB.

saPassword

Mot de passe de l'utilisateur SA spécifié.

D'autres paramètres peuvent être requis pour votre environnement. Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, reportez-vous à l'aide de la commande d'administration **configEventServiceSQLServerDB**.

Résultats

La commande d'administration crée la source de données requise au niveau de la portée spécifiée ; si vous avez indiqué la valeur true pour le paramètre createDB, la commande exécute également le script de configuration de base de données généré pour créer la base de données.

Les scripts de configuration de base de données générés sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/dbscripts/* répertoire sqlserver. Dans un environnement de déploiement réseau, ces scripts sont stockés dans le répertoire des profils du gestionnaire de déploiement. Si vous avez indiqué une valeur pour le paramètre outputScriptDir facultatif, les scripts sont stockés à cet emplacement à la place. Vous pouvez utiliser ces scripts pour configurer manuellement la base de données d'événements à tout moment.

Exécution manuelle des scripts de configuration de base de données

Vous pouvez exécuter manuellement et à tout moment les scripts générés par les commandes d'administration de configuration de base de données.

A propos de cette tâche

La configuration de la base de données est une procédure en deux étapes. La commande d'administration de configuration de base de données génère en premier lieu un script approprié pour environnement et votre base de données ; ce script généré permet alors de configurer la base de données d'événements et les sources de données. Si vous spécifiez la valeur `true` pour le paramètre `createDB` lors de l'exécution de la commande d'administration, les deux étapes surviennent automatiquement.

En revanche, si vous indiquez la valeur `false` pour le paramètre `createDB`, vous devez terminer manuellement la configuration de la base de données en exécutant le script généré sur le système cible. Vous pouvez être amené à exécuter manuellement le script dans l'un des cas suivants :

- Vous devez configurer la base de données d'événements sur un système autre que celui depuis lequel vous avez exécuté la commande d'administration.
- Vous devez recréer ultérieurement la base de données d'événements.
- Vous devez modifier les options par défaut utilisées par le script généré avant d'exécuter ce dernier.

Création manuelle d'une base de données d'événements Derby

A propos de cette tâche

Pour exécuter manuellement le script de configuration de base de données généré pour une base de données d'événements Derby :

Procédure

1. Sur le système serveur, accédez au répertoire contenant le script généré. Ces scripts sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/derby* ; si vous avez indiqué une valeur pour le paramètre `outputScriptDir` de la commande d'administration de configuration de base de données, les scripts sont stockés à cet emplacement.
2. A l'aide d'un éditeur de texte ASCII, apportez les modifications requises au script de configuration. Le nom du script varie en fonction du système d'exploitation utilisé :

- **Windows** Systèmes Windows : `cr_event_derby.bat`
- **Linux** **UNIX** Systèmes Linux et UNIX : `cr_event_derby.sh`
- **i5/OS** Systèmes iSeries : `cr_event_derby`

3. Facultatif : Si vous configurez la base de données sur un système iSeries, démarrez Qshell Interpreter.
4. Exécutez le script de création de base de données à l'aide de la syntaxe suivante (le cas échéant, pensez à indiquer l'extension de fichier) :

```
cr_event_derby -p chemin_profil  
[-s nom_serveur|-c nom_cluster]
```

Les paramètres utilisés sont les suivants :

-p *chemin_profil*

Chemin d'accès au répertoire des profils WebSphere. Ce paramètre est obligatoire.

-s *nom_serveur*

Nom du serveur. Ce paramètre est requis si vous configurez la base de données au niveau du serveur.

-c *nom_cluster*

Nom du cluster. Ce paramètre est requis si vous configurez la base de données au niveau du cluster.

Par exemple, la commande suivante permet de créer la base de données Derby au niveau du serveur `serveur1`, à l'aide du profil `profil1` :

```
cr_event_derby -p c:\WebSphere\appserver\profiles\myprofile -s serveur1
```

5. Redémarrez le serveur. Pour un noeud fédéré, vous devez également arrêter et redémarrer l'agent de noeud à l'aide des commandes **stopNode** et **startNode**.

Que faire ensuite

Une fois la base de données configurée, vous pouvez utiliser la console d'administration pour tester la configuration de la base de données. Pour ce faire, accédez à la source de données JDBC appropriée et sélectionnez l'option **Tester la connexion**.

Création manuelle d'une base de données d'événements DB2 sur un système Linux, UNIX ou Windows

A propos de cette tâche

Pour exécuter manuellement le script de configuration de base de données généré pour une base de données d'événements DB2 sur un système Linux, UNIX ou Windows :

Procédure

1. Sur le système serveur, accédez au répertoire contenant le script généré. Ces scripts sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/db2* ; si vous avez indiqué une valeur pour le paramètre `outputScriptDir` de la commande d'administration de configuration de base de données, les scripts sont stockés à cet emplacement.
2. A l'aide d'un éditeur de texte ASCII, apportez les modifications requises au script de configuration. Le nom du script varie en fonction du système d'exploitation utilisé :

- **Windows** Systèmes Windows : `cr_event_db2.bat`

- **Linux** **UNIX** Systèmes Linux et UNIX : `cr_event_db2.sh`

3. Exécutez le script de création de base de données à l'aide de la syntaxe suivante (le cas échéant, pensez à indiquer l'extension de fichier) :

```
cr_event_db2 [client|server] utilisateur_base_de_donnees  
[mot_de_passe_base_de_donnees]
```

Les paramètres utilisés sont les suivants :

client|server

Indique s'il s'agit d'une base de données client ou serveur. Vous devez spécifier la valeur **client** ou **serveur**.

utilisateur_base_de_donnees

ID utilisateur de la base de données. Ce paramètre est obligatoire.

mot_de_passe_base_de_donnees

Mot de passe de l'utilisateur de la base de données. Si vous n'indiquez pas de mot de passe pour une base de données client, vous êtes invité à le faire.

Par exemple, la commande suivante permet de créer la base de données d'événements DB2 pour une base de données client, à l'aide de l'ID utilisateur `db2admin` et du mot de passe `motdepasse` :

```
cr_event_db2 client db2admin motdepasse
```

4. Redémarrez le serveur. Pour un noeud fédéré, vous devez également arrêter et redémarrer l'agent de noeud à l'aide des commandes **stopNode** et **startNode**.

Que faire ensuite

Une fois la base de données configurée, vous pouvez utiliser la console d'administration pour tester la configuration de la base de données. Pour ce faire, accédez à la source de données JDBC appropriée et sélectionnez l'option **Tester la connexion**.

Création manuelle d'une base de données d'événements DB2 sur un système z/OS

A propos de cette tâche

Pour exécuter manuellement le script de configuration de base de données généré pour une base de données d'événements DB2 sur un système z/OS, à l'aide d'un système client Linux, UNIX ou Windows :

Procédure

1. Sur le système serveur, accédez au répertoire contenant le script généré. Ces scripts sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/db2zos* ; si vous avez indiqué une valeur pour le paramètre `outputScriptDir` de la commande d'administration de configuration de base de données, les scripts sont stockés à cet emplacement.
2. A l'aide d'un éditeur de texte ASCII, apportez les modifications requises au script de configuration. Le nom du script varie en fonction du système d'exploitation utilisé :

- **Windows** Systèmes Windows : `cr_event_db2zos.bat`
- **Linux** **UNIX** Systèmes Linux et UNIX : `cr_event_db2zos.sh`

3. Exécutez le script de création de base de données à l'aide de la syntaxe suivante (le cas échéant, pensez à indiquer l'extension de fichier) :

```
cr_event_db2zos [dbName=nom_base_de_données] utilisateur_base_de_données  
[mot_de_passe_base_de_données]
```

Les paramètres utilisés sont les suivants :

nom_base_de_données

Nom de la base de données à utiliser. Ce paramètre est facultatif. Si vous ne spécifiez pas de nom de base de données, un nom est généré.

utilisateur_base_de_données

ID utilisateur de la base de données à indiquer. Ce paramètre est obligatoire.

mot_de_passe_base_de_données

Mot de passe de l'utilisateur de la base de données. Si vous n'indiquez pas de mot de passe, la base de données DB2 vous invite à le faire.

Par exemple, la commande suivante permet de créer une base de données d'événements DB2 nommée `event`, à l'aide de l'ID utilisateur `db2admin` et du mot de passe `motdepasse` :

```
cr_event_db2zos dbName=client db2admin motdepasse
```

4. Redémarrez le serveur. Pour un noeud fédéré, vous devez également arrêter et redémarrer l'agent de noeud à l'aide des commandes **stopNode** et **startNode**.

Que faire ensuite

Une fois la base de données configurée, vous pouvez utiliser la console d'administration pour tester la configuration de la base de données. Pour ce faire, accédez à la source de données JDBC appropriée et sélectionnez l'option **Tester la connexion**.

Création manuelle d'une base de données d'événements DB2 sur un système iSeries

A propos de cette tâche

Pour exécuter manuellement le script de configuration de base de données généré pour une base de données d'événement DB2 sur un système iSeries :

Procédure

1. Sur le système serveur, accédez au répertoire contenant le script généré. Ces scripts sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/db2iseries* ; si vous avez indiqué une valeur pour le paramètre `outputScriptDir` de la commande d'administration de configuration de base de données, les scripts sont stockés à cet emplacement.
2. A l'aide d'un éditeur de texte ASCII, apportez les modifications requises au script `cr_event_db2iseries`.
3. Démarrez Qshell Interpreter.
4. Exécutez le script de création de base de données à l'aide de la syntaxe suivante :

```
cr_event_db2iseries utilisateur_base_de_donnees mot_de_passe_base_de_donnees
```

Les paramètres utilisés sont les suivants :

utilisateur_base_de_donnees

ID utilisateur de la base de données. Ce paramètre est obligatoire.

mot_de_passe_base_de_donnees

Mot de passe de l'utilisateur de la base de données. Ce paramètre est obligatoire.

Par exemple, la commande suivante permet de créer la base de données d'événements DB2 à l'aide de l'ID utilisateur `db2admin` et du mot de passe `motdepasse` :

```
cr_event_db2iseries db2admin motdepasse
```

5. Redémarrez le serveur. Pour un noeud fédéré, vous devez également arrêter et redémarrer l'agent de noeud à l'aide des commandes **stopNode** et **startNode**.

Que faire ensuite

Une fois la base de données configurée, vous pouvez utiliser la console d'administration pour tester la configuration de la base de données. Pour ce faire, accédez à la source de données JDBC appropriée et sélectionnez l'option **Tester la connexion**.

Création manuelle d'une base de données d'événements Informix

Vous pouvez exécuter manuellement et à tout moment les scripts générés par les commandes d'administration de configuration de base de données.

A propos de cette tâche

Pour exécuter manuellement les scripts de configuration de base de données générés pour une base de données d'événements Informix :

Procédure

1. Sur le système serveur, accédez au répertoire contenant le script généré. Ces scripts sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/informix* ; si vous avez indiqué une valeur pour le paramètre `outputScriptDir` de la commande d'administration de configuration de base de données, les scripts sont stockés à cet emplacement.
2. A l'aide d'un éditeur de texte ASCII, apportez les modifications requises au script de configuration. Le nom du script varie en fonction du système d'exploitation utilisé :
 - **Windows** Systèmes Windows : `cr_event_informix.bat`
 - **Linux** **UNIX** Systèmes Linux et UNIX : `cr_event_informix.sh`
3. Exécutez le script de création de base de données, sans définir aucun paramètre.
4. Redémarrez le serveur. Pour un noeud fédéré, vous devez également arrêter et redémarrer l'agent de noeud à l'aide des commandes **stopNode** et **startNode**.

Que faire ensuite

Une fois la base de données configurée, vous pouvez utiliser la console d'administration pour tester la configuration de la base de données. Pour ce faire, accédez à la source de données JDBC appropriée et sélectionnez l'option **Tester la connexion**.

Création manuelle d'une base de données d'événements Oracle

A propos de cette tâche

Pour exécuter manuellement le script de configuration de base de données pour une base de données d'événements Oracle :

Procédure

1. Sur le système serveur, accédez au répertoire contenant le script généré. Ces scripts sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/oracle* ; si vous avez indiqué une valeur pour le paramètre `outputScriptDir` de la commande d'administration de configuration de base de données, les scripts sont stockés à cet emplacement.
2. A l'aide d'un éditeur de texte ASCII, apportez les modifications requises au script de configuration. Le nom du script varie en fonction du système d'exploitation utilisé :

- **Windows** Systèmes Windows : `cr_event_oracle.bat`
- **Linux** **UNIX** Systèmes Linux et UNIX : `cr_event_oracle.sh`

3. Exécutez le script de création de base de données à l'aide de la syntaxe suivante (le cas échéant, pensez à indiquer l'extension de fichier) :

```
cr_event_oracle mot_de_passe utilisateur_système  
mot_de_passe_système [sid=sid] [oracleHome=base_oracle]
```

Les paramètres utilisés sont les suivants :

mot_de_passe

Mot de passe relatif à l'ID utilisateur du schéma. Ce paramètre est obligatoire.

utilisateur_système

ID utilisateur disposant de privilèges SYSDBA au sein de la base de données Oracle (correspond généralement à l'utilisateur système). Ce paramètre est obligatoire.

mot_de_passe_système

Mot de passe relatif à l'ID utilisateur système spécifié. Si cet ID utilisateur n'utilise pas de mot de passe, saisissez `none`.

sid=sid

Identificateur système (SID) Oracle. Ce paramètre est facultatif.

oracleHome=base_oracle

Répertoire de base Oracle. Ce paramètre est facultatif. Si vous n'indiquez pas de valeur, un chemin d'accès généré est utilisé.

Par exemple, la commande suivante permet de créer la base de données d'événements Oracle à l'aide de l'ID utilisateur du schéma `auser` et de l'ID utilisateur du système `sys` :

```
cr_event_oracle auser sys syspassword sid=event oracleHome=c:\oracle
```

4. Redémarrez le serveur. Pour un noeud fédéré, vous devez également arrêter et redémarrer l'agent de noeud à l'aide des commandes **stopNode** et **startNode**.

Que faire ensuite

Une fois la base de données configurée, vous pouvez utiliser la console d'administration pour tester la configuration de la base de données. Pour ce faire, accédez à la source de données JDBC appropriée et sélectionnez l'option **Tester la connexion**.

Création manuelle d'une base de données d'événements SQL Server

A propos de cette tâche

Pour exécuter manuellement le script de configuration de base de données généré pour une base de données d'événements SQL Server :

Procédure

1. Sur le système serveur, accédez au répertoire contenant le script généré. Ces scripts sont stockés par défaut dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/sqlserver* ; si vous avez indiqué une valeur pour le paramètre `outputScriptDir` de la commande d'administration de configuration de base de données, les scripts sont stockés à cet emplacement.
2. A l'aide d'un éditeur de texte ASCII, apportez les modifications requises au script `cr_event_mssql.bat`.
3. Exécutez le script de création de base de données à l'aide de la syntaxe suivante :

```
cr_event_mssql ID_utilisateur mot_de_passe [server=serveur]
sauser=utilisateur_SA sapassword=mot_de_passe_SA
```

Les paramètres utilisés sont les suivants :

ID_utilisateur

ID utilisateur de connexion au serveur SQL Server possédant les tables créées. Cet ID utilisateur doit être créé dans le serveur SQL de sorte qu'une connexion JDBC puisse être établie avec la base de données. (Les pilotes JDBC ne prennent pas en charge les connexions sécurisées.)

mot_de_passe

Mot de passe du nouvel ID utilisateur de connexion créé.

server=serveur

Nom du serveur contenant la base de données SQL Server. Ce paramètre est facultatif. Sa valeur par défaut est définie sur l'hôte local.

sauser=utilisateur_sa

ID utilisateur SA. Cet ID utilisateur doit disposer de privilèges adéquats pour créer des bases de données et des connexions utilisateur.

sapassword=mot_de_passe_sa

Mot de passe SA, si vous utilisez un mode d'identification mixte. Si aucun mot de passe n'a été indiqué pour l'ID utilisateur SA, spécifiez `sapassword=` sans aucune valeur. Ne tenez pas compte de ce paramètre si vous utilisez une connexion sécurisée.

Par exemple, la commande suivante permet de créer la base de données d'événements SQL Server à l'aide de l'ID utilisateur de connexion `userid` :

```
cr_event_mssql userid apassword server=myserver sauser=sa sapassword=sapassword
```

4. Redémarrez le serveur. Pour un noeud fédéré, vous devez également arrêter et redémarrer l'agent de noeud à l'aide des commandes **stopNode** et **startNode**.

Que faire ensuite

Une fois la base de données configurée, vous pouvez utiliser la console d'administration pour tester la configuration de la base de données. Pour ce faire, accédez à la source de données JDBC appropriée et sélectionnez l'option **Tester la connexion**.

Mise à niveau de la base de données d'événements depuis une version antérieure

Si vous avez effectué une migration d'une version antérieure de Common Event Infrastructure et utilisez la persistance des événements, vous devrez mettre à niveau la base de données existante.

A propos de cette tâche

La mise à niveau de la base de données d'événements est nécessaire si la migration est effectuée depuis Common Event Infrastructure version 5.1 ou antérieure.

Le processus met à niveau le schéma et les métadonnées de la base de données d'événements vers la version actuelle, tout en conservant les données d'événements existantes.

Le script de mise à niveau de la base de données entraîne la mise à niveau du schéma et des métadonnées de la base de données d'événements existante vers la version actuelle.

Versions non prises en charge : Si votre base de données d'événements utilise une version de logiciel de base de données qui n'est plus prise en charge par Common Event Infrastructure 6.0, vous devez d'abord migrer la base de données vers une version prise en charge à l'aide de la procédure adaptée au logiciel de base de données. Vous pouvez ensuite procéder à la mise à niveau de la base de données d'événements.

Mise à niveau de la base de données d'événements depuis Cloudscape vers Derby

Si vous disposez d'une base de données d'événements Cloudscape, vous devez la mettre à niveau vers la base de données Derby.

A propos de cette tâche

Pour mettre à niveau une base de données d'événements Cloudscape vers une base de données Derby :

Procédure

1. Accédez au répertoire *racine_profil/bin*.
2. Exécutez le script de migration Derby correspondant à votre système d'exploitation :

- Systèmes Windows :

```
eventMigrateDerby répertoire_base_de_données [generateDDLonly]
```

- Systèmes Linux et UNIX :

```
eventMigrateDerby.sh répertoire_base_de_données [generateDDLonly]
```

Les paramètres utilisés sont les suivants :

répertoire_base_de_données

Indique le chemin d'accès du répertoire contenant la base de données d'événements Cloudscape existante. Ce paramètre est obligatoire.

generateDDLonly

Indique si vous voulez générer le script DLL de mise à niveau de la base de données sans l'exécuter. Indiquez ce paramètre si vous souhaitez effectuer manuellement la mise à niveau de la base de données à un autre moment ou sur un autre système. Ce paramètre est facultatif. Son comportement par défaut est de générer et d'exécuter le script DDL.

Résultats

Le script de migration Derby crée une copie de sauvegarde de la base de données d'événements Cloudscape existante dans le répertoire *répertoire_base_de_données.bak*, puis il crée deux scripts DDL dans le répertoire de base de données :

- *event_newDDL.sql*
- *eventcatalog_newDDL.sql*

Si vous n'avez pas indiqué de paramètre *generateDDLonly*, le script de migration exécute automatiquement ces scripts DDL afin de terminer la mise à niveau vers Derby.

L'exemple suivant permet la mise à niveau d'une base de données d'événements Cloudscape sur un système Windows :

```
eventMigrateDerby c:\databases\cloudscapeEventDB
```

Mise à niveau d'une base de données d'événements DB2 depuis une version antérieure

Si vous disposez d'une base de données d'événements DB2 de la version 5.1 de Common Event Infrastructure sous Linux, UNIX, ou Windows, vous devez la mettre à niveau.

A propos de cette tâche

Pour mettre à niveau une base de données d'événements DB2 sur un système Linux ou UNIX, procédez comme suit :

Procédure

1. Réalisez une copie de sauvegarde de la base de données d'événements existante.
2. Accédez au répertoire *racine_profil/bin*.
3. Exécutez le script de mise à niveau DB2 correspondant à votre système d'exploitation :

- **Windows** Systèmes Windows :

```
eventUpgradeDB2 runUpgrade=[true|false] dbUser=utilisateur  
[dbName=nom] [dbPassword=mot_de_passe]  
[dbNode=noeud] [scriptDir=rép]
```

- **Linux** **UNIX** Systèmes Linux et UNIX :

```
eventUpgradeDB2.sh runUpgrade=[true|false] dbUser=utilisateur  
[dbName=nom] [dbPassword=mot_de_passe]  
[dbNode=noeud] [scriptDir=rép]
```

Les paramètres requis standard sont les suivants :

runUpgrade

Indique si vous souhaitez que le script de mise à niveau exécute automatiquement les scripts DDL générés pour effectuer la mise à niveau de la base de données. Ce paramètre est obligatoire. Indiquez *false* si vous souhaitez effectuer manuellement la mise à niveau de la base de données à un autre moment ou sur un autre système.

dbUser

Indique l'ID utilisateur DB2 à utiliser. Ce paramètre est obligatoire.

dbName

Indique le nom de la base de données DB2. Le nom par défaut de la base de données d'événements est *event*. Ce paramètre est obligatoire si vous avez spécifié *runUpgrade=true*.

dbPassword

Indique le mot de passe correspondant à l'ID utilisateur DB2 spécifié. Ce paramètre est facultatif ; si vous n'indiquez pas de mot de passe, DB2 vous invite à le faire.

dbNode

Indique le nom du noeud de base de données. Ce paramètre est obligatoire si vous exécutez le script de mise à niveau depuis un système client DB2.

scriptDir

Indique le répertoire dans lequel stocker les scripts DDL générés. Ce paramètre est facultatif ; si vous n'indiquez pas de répertoire, les scripts sont stockés dans le dossier *.\eventDBUpgrade\db2*.

Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, exécutez le script **eventUpgradeDB2** sans aucun paramètre.

Résultats

Le script de mise à niveau génère les scripts DDL nécessaires à la mise à niveau de la base de données d'événements. Si vous avez spécifié `runUpgrade=true`, les scripts DDL sont exécutés automatiquement et la mise à niveau est effectuée.

L'exemple suivant permet la mise à niveau d'une base de données DB2 sur un système Windows :

```
eventUpgradeDB2 runUpgrade=true dbUser=db2inst1 dbName=event
```

Que faire ensuite

Si vous avez spécifié `runUpgrade=false`, vous devez exécuter manuellement les scripts DDL sur le système de base de données afin de mettre à niveau cette dernière.

Mise à niveau d'une base de données d'événements DB2 pour z/OS depuis une version antérieure

Si vous disposez d'une base de données d'événements DB2 de la version 5.1 de Common Event Infrastructure sous z/OS, vous devez la mettre à niveau.

A propos de cette tâche

Pour mettre à niveau une base de données d'événements DB2 sur un système z/OS, procédez comme suit :

Procédure

1. Réalisez une copie de sauvegarde de la base de données d'événements existante.
2. Accédez au répertoire `racine_profil/bin`.
3. Exécutez le script de mise à niveau DB2 pour z/OS correspondant à votre système d'exploitation client :

- **Windows** Systèmes Windows :

```
eventUpgradeDB2ZOS runUpgrade=[true|false] dbUser=utilisateur  
[dbName=nom] [dbPassword=mot_de_passe]  
[scriptDir=rép] storageGroup=groupe  
bufferPool14K=4kbufpool bufferPool18k=8kbufpool  
bufferPool16K=16kbufpool
```

- **Linux** **UNIX** Systèmes Linux et UNIX :

```
eventUpgradeDB2ZOS.sh runUpgrade=[true|false] dbUser=utilisateur  
[dbName=nom] [dbPassword=mot_de_passe]  
[scriptDir=rép] storageGroup=groupe  
bufferPool14K=4kbufpool bufferPool18k=8kbufpool  
bufferPool16K=16kbufpool
```

Les paramètres suivants sont généralement requis :

runUpgrade

Indique si vous souhaitez que le script de mise à niveau exécute automatiquement les scripts DDL générés pour effectuer la mise à niveau de la base de données. Ce paramètre est obligatoire. Indiquez false si vous souhaitez effectuer manuellement la mise à niveau de la base de données à un autre moment ou sur un autre système.

Systèmes z/OS : Ce paramètre est ignoré sur un système z/OS natif. L'exécution automatique des scripts DDL générés est uniquement prise en charge sur un système client.

dbUser

Indique l'ID utilisateur DB2 à utiliser. Ce paramètre est obligatoire.

dbName

Indique le nom de la base de données DB2. Le nom par défaut de la base de données d'événements est event. Ce paramètre est obligatoire si vous avez spécifié runUpgrade=true.

dbPassword

Indique le mot de passe correspondant à l'ID utilisateur DB2 spécifié. Ce paramètre est facultatif ; si vous n'indiquez pas de mot de passe, DB2 vous invite à le faire.

scriptDir

Indique le répertoire dans lequel stocker les scripts DDL générés. Ce paramètre est facultatif ; si vous n'indiquez pas de répertoire, les scripts sont stockés dans le dossier `.\eventDBUpgrade\db2zos`.

storageGroup

Indique le nom du groupe de stockage. Ce paramètre est obligatoire.

bufferPool4K

Indique le nom du pool de mémoire tampon 4K. Ce paramètre est obligatoire.

bufferPool8K

Indique le nom du pool de mémoire tampon 8K. Ce paramètre est obligatoire.

bufferPool16K

Indique le nom du pool de mémoire tampon 16K. Ce paramètre est obligatoire.

Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, exécutez le script **eventUpgradeDB2ZOS** sans aucun paramètre.

Résultats

Le script de mise à niveau génère les scripts DDL nécessaires à la mise à niveau de la base de données d'événements. Si vous avez spécifié runUpgrade=true sur un système client, les scripts DDL sont exécutés automatiquement et la mise à niveau est effectuée.

L'exemple suivant permet la mise à niveau d'une base de données d'événements DB2 pour z/OS depuis un système client Windows :

```
eventUpgradeDB2ZOS runUpgrade=true dbUser=db2inst1 dbName=event  
storageGroup=sysdeflt bufferPool4K=BP9 bufferPool8K=BP8K9 bufferPool16K=BP16K9
```

Si vous avez spécifié `runUpgrade=false` ou exécuté le script de mise à niveau sur le système z/OS, vous devez exécuter manuellement les scripts DDL générés sur le système z/OS à l'aide de la fonction SPUFI (SQL Processor Using File Input). Cette étape effectue la mise à niveau de la base de données.

Mise à niveau d'une base de données d'événements Oracle depuis la version 5

Si vous disposez d'une base de données d'événements Oracle de la version 5.1 de Common Event Infrastructure, vous devez la mettre à niveau.

A propos de cette tâche

Pour mettre à niveau une base de données d'événements Oracle, procédez comme suit :

Procédure

1. Réalisez une copie de sauvegarde de la base de données d'événements existante.
2. Accédez au répertoire `racine_profil/bin`.
3. Exécutez le script de mise à niveau Oracle correspondant à votre système d'exploitation :

- Systèmes Windows :

```
eventUpgradeOracle runUpgrade=[true|false] schemaUser=utilisateur_schéma  
[oracleHome=rép] [dbName=nom]  
[dbUser=utilisateur_sys] [dbPassword=mot_de_passe]  
[scriptDir=rép]
```

- Systèmes Linux et UNIX :

```
eventUpgradeOracle.sh runUpgrade=[true|false] schemaUser=utilisateur_schéma  
[oracleHome=rép] [dbName=nom]  
[dbUser=utilisateur_sys] [dbPassword=mot_de_passe]  
[scriptDir=rép]
```

Les paramètres suivants sont généralement requis :

runUpgrade

Indique si vous souhaitez que le script de mise à niveau exécute automatiquement les scripts DDL générés pour effectuer la mise à niveau de la base de données. Ce paramètre est obligatoire. Indiquez `false` si vous souhaitez effectuer manuellement la mise à niveau de la base de données à un autre moment ou sur un autre système.

schemaUser

Indique l'ID utilisateur Oracle du propriétaire des tables de base de données. Ce paramètre est obligatoire.

oracleHome

Indique le répertoire de base Oracle. Ce paramètre est obligatoire si vous avez spécifié `runUpgrade=true`.

dbName

Indique le nom de la base de données Oracle. Le nom par défaut de la base de données d'événements est `event`. Ce paramètre est obligatoire si vous avez spécifié `runUpgrade=true`.

dbUser

Indique l'ID utilisateur du système Oracle. Ce paramètre est obligatoire si vous avez spécifié `runUpgrade=true`.

dbPassword

Indique le mot de passe correspondant à l'ID utilisateur système.
N'indiquez pas ce paramètre si l'ID utilisateur système n'est associé à aucun mot de passe.

scriptDir

Indique le répertoire dans lequel stocker les scripts DDL générés. Ce paramètre est facultatif ; si vous n'indiquez pas de répertoire, les scripts sont stockés dans le dossier `.\eventDBUpgrade\oracle`.

Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, exécutez le script **eventUpgradeOracle** sans aucun paramètre.

Résultats

Le script de mise à niveau génère les scripts DDL nécessaires à la mise à niveau de la base de données d'événements. Si vous avez spécifié `runUpgrade=true`, les scripts DDL sont exécutés automatiquement et la mise à niveau est effectuée.

L'exemple suivant permet la mise à niveau d'une base de données Oracle sur un système Windows :

```
eventUpgradeOracle runUpgrade=true schemaUser=cei  
dbName=event dbUser=sys
```

Que faire ensuite

Si vous avez spécifié `runUpgrade=false`, vous devez exécuter manuellement les scripts DDL sur le système de base de données afin de mettre à niveau cette dernière.

Configuration de l'infrastructure d'événement commune entre cellules pour WebSphere Business Monitor

Vous devez configurer la connectivité entre un serveur distant produisant des événements d'infrastructure d'événement commune et le serveur WebSphere Business Monitor.

Avant de commencer

Vous devez tout d'abord déterminer les noms de ces objets d'administration en ouvrant la console d'administration sur le serveur qui est configuré pour fournir le moteur d'exécution à WebSphere Business Monitor. Au cours de l'installation de WebSphere Business Monitor, une instance de bus d'intégration de services et un moteur de messagerie sont créés pour la transmission des événements. Vous pouvez noter le nom du bus en ouvrant la console d'administration du serveur configuré pour fournir le moteur d'exécution à WebSphere Business Monitor, puis en sélectionnant **Intégration de services > Bus**. Le nom du bus configuré pour Monitor se présente sous la forme `MONITOR.nom_cellule.Bus`.

A propos de cette tâche

Les étapes 1 à 5 doivent être exécutées sur le serveur émettant les événements, alors que les étapes 6 et 7 doivent l'être sur le serveur de contrôle.

Procédure

1. Lancez le serveur qui devra produire les événements. Vous pouvez déterminer la cellule et le nom de poste du serveur émettant les événements en sélectionnant **Serveurs > Serveurs d'applications > nom_serveur**, puis en cliquant sur l'onglet **Phase d'exécution**.
2. Créez une nouvelle instance de bus d'intégration de services.
 - a. Ouvrez la console d'administration du serveur qui émet les événements.
 - b. Sélectionnez **Intégration de services > Bus**.
 - c. Cliquez sur **Nouveau**.
 - d. Saisissez le nom du bus dans le formulaire `MONITOR.nom_cellule.Bus`.
 - e. Désélectionnez **Sécuriser**.
 - f. Cliquez sur **Appliquer**.
 - g. Sauvegardez les modifications.
3. Ajoutez le serveur émetteur d'événements comme membre de bus
 - a. Sélectionnez **Intégration de services > Bus > Monitor.nom_cellule.Bus > Topologie > Membres de bus**
 - b. Cliquez sur **Ajouter**.
 - c. Sélectionnez le serveur dans le menu.
 - d. Cliquez sur **Suivant**.
 - e. Cliquez sur **Terminer** Le serveur sera ajouté comme membre de bus.
 - f. Sauvegardez les modifications.
4. Définissez un bus externe.
 - a. Sélectionnez **Intégration de services > Bus > Monitor.nom_cellule.Bus**.
 - b. Sélectionnez **Topologie > Bus externes > Nouveau**.
 - c. Dans le champ **Nom**, indiquez le nom du bus de contrôle sur le serveur où est installé le contrôleur.

- d. Cliquez sur **Suivant**.
 - e. Sélectionnez **Direct** dans le menu **Type de routage**.
 - f. Cliquez sur **Suivant**.
 - g. Laissez les champs **Entrant** et **Sortant** de l'ID utilisateur vides.
 - h. Cliquez sur **Suivant**.
 - i. Cliquez sur **Terminer**. La création du bus externe est terminée.
 - j. Sauvegardez les modifications.
5. Définissez un lien Bus d'intégration de services.
 - a. Sélectionnez **Intégration de services > Bus > Monitor.nom_cellule.Bus**.
 - b. Sélectionnez **Topologie > Moteurs de messagerie**.
 - c. Cliquez sur le moteur de messagerie qui a été créé lorsque vous avez ajouté le serveur comme membre de bus.
 - d. Sélectionnez **Caractéristiques complémentaires > Lien Bus d'intégration de services > Nouveau**.
 - e. Entrez les valeurs appropriées dans les champs suivants :
 - **Nom** — un nom qui permette d'identifier facilement le lien SIBus.
 - **Nom du bus externe** — sélectionnez dans le menu le bus externe que vous avez créé antérieurement.
 - **Nom du moteur de messagerie distant** — vous devez ici indiquer le nom du moteur de messagerie sur le serveur de contrôle.
 - **Noeuds finaux d'amorçage** — vous devez spécifier l'adresse IP et le noeud final d'amorçage du port pour le serveur de contrôle.

Remarque : Pour déterminer le numéro de port, ouvrez la console d'administration du serveur de contrôle et sélectionnez **Serveurs > Serveurs d'applications > nom_serveur > Communications > Ports**.
 6. Créez une définition de bus externe.
 - a. Ouvrez la console d'administration du serveur de contrôle.
 - b. Sélectionnez **Intégration de services > Bus > MONITOR.nom_cellule.Bus**
 - c. Sélectionnez **Topologie > Bus externes > Nouveau**.
 - d. Entrez le nom exact du bus que vous avez créé sur le serveur émetteur d'événements lors de l'étape 2.
 - e. Cliquez sur **Suivant**.
 - f. Sélectionnez **Direct** dans le menu **Type de routage**.
 - g. Cliquez sur **Suivant**.
 - h. Laissez les champs **Entrant** et **Sortant** de l'ID utilisateur vides.
 - i. Cliquez sur **Suivant**.
 - j. Cliquez sur **Terminer**. La création du bus externe est terminée.
 - k. Sauvegardez les modifications.
 7. Créez un lien Bus d'intégration de services.
 - a. Sélectionnez **Intégration de services > Bus > Monitor.nom_cellule.Bus**.
 - b. Sélectionnez **Topologie > Moteurs de messagerie**.
 - c. Cliquez sur le moteur de messagerie qui a été créé lorsque vous avez ajouté le serveur comme membre de bus.
 - d. Sélectionnez **Caractéristiques supplémentaires > Lien Bus d'intégration de services > Nouveau**.
 - e. Entrez les valeurs appropriées dans les champs suivants :

- **Nom** — entrez le nom du lien que vous avez créé lors de l'étape 5.
- **Nom du bus externe** — sélectionnez dans le menu le bus externe que vous avez créé lors de l'étape 2.
- **Nom du moteur de messagerie distant** — vous devez ici indiquer le nom du moteur de messagerie sur le serveur de contrôle (voir l'étape 5.c).
- **Noeuds finaux d'amorçage** — vous devez spécifier l'adresse IP et le noeud final d'amorçage du port pour le serveur de contrôle.

Que faire ensuite

Validez la configuration de la manière suivante :

1. Redémarrez les deux serveurs.
2. Ouvrez la console d'administration du serveur qui produit les événements.
3. Sélectionnez **Intégration de services > Bus > MONITOR.nom_cellule.Bus**.
4. Sélectionnez **Moteurs de messagerie > nom_moteur_messagerie > Lien Bus d'intégration de services**.
5. Confirmez que l'indicateur d'état a démarré.
6. Ouvrez la console d'administration du serveur de contrôle.
7. Sélectionnez **Intégration de services > Bus > MONITOR.nom_cellule.Bus**.
8. Sélectionnez **Moteurs de messagerie > nom_moteur_messagerie > Lien Bus d'intégration de services**.
9. Confirmez que l'indicateur d'état a démarré.

Suppression de la configuration de l'infrastructure CEI

Si vous devez supprimer la configuration de Common Event Infrastructure en vue de la désinstallation du serveur, vous devez d'abord supprimer les applications d'entreprise déployées et la configuration de la base de données.

A propos de cette tâche

L'installation de Common Event Infrastructure est effectuée sur l'installation de base du serveur, mais elle n'est active que si elle est déjà configurée. Cette rubrique indique comment supprimer une instance de Common Event Infrastructure précédemment configurée. Pour supprimer la configuration de Common Event Infrastructure, procédez comme suit :

Suppression de la base de données d'événements

Pour supprimer la base de données d'événements, vous pouvez utiliser la commande d'administration correspondant à ce type de base de données.

A propos de cette tâche

Pour supprimer la base de données d'événements :

Procédure

1. Démarrez l'outil wsadmin.
2. L'objet AdminTask permet d'exécuter la commande d'administration appropriée pour votre base de données d'événements :

Type de base de données	Commande
Derby	«removeEventServiceDerbyDB», à la page 129
DB2 sur les systèmes Linux, UNIX et Windows	«removeEventServiceDB2DB», à la page 124
DB2 sur les systèmes z/OS	
DB2 sur mes systèmes iSeries	«removeEventServiceDB2iSeriesDB», à la page 126
Informix	«removeEventServiceInformixDB», à la page 130
Oracle	«removeEventServiceOracleDB», à la page 132
SQL Server	«removeEventServiceSQLServerDB», à la page 134

Remarques spécifiques à la base de données :

- Sur un système z/OS, la commande d'administration permet uniquement de supprimer la source de données JDBC. Pour supprimer la base de données, vous devez utiliser la commande SPUFI pour exécuter le script de suppression de base de données généré lors de la création de la base de données. Par défaut, ce script se trouve dans le répertoire *racine_profil/databases/event/noeud/serveur/dbscripts/db2zos*.
- **i5/OS** Sur un système iSeries, la commande d'administration supprime uniquement la source de données JDBC. Pour supprimer la base de données sur le système iSeries, supprimez la collection créée pour la base de données.

Les paramètres requis (par exemple, ID utilisateur et mot de passe) varient en fonction du type de base de données. Pour obtenir la liste complète des paramètres ainsi que des informations relatives à l'utilisation, reportez-vous à l'aide de la commande d'administration.

Suppression de l'application Common Event Infrastructure

Pour supprimer manuellement l'application d'entreprise du service Événements et les ressources du serveur, vous pouvez utiliser la commande d'administration **removeEventService**.

A propos de cette tâche

Pour supprimer l'application d'entreprise du service Événements :

Procédure

1. Démarrez l'outil wsadmin.
2. L'objet AdminTask permet d'exécuter la commande d'administration **removeEventService** en mode interactif ou de traitement par lots. Les paramètres de la commande d'administration **removeEventService** sont les suivants :

nodeName

Nom du noeud sur lequel le service d'événements est déployé. Ce paramètre est facultatif. Si vous ne spécifiez pas de nom pour le noeud, celui du noeud en cours sera attribué par défaut. Si vous indiquez un nom de noeud, vous devez également indiquer le nom du serveur, à l'aide du paramètre **serverName**. Ce paramètre n'est pas valide si vous supprimez le service Événements d'un cluster.

serverName

Nom du serveur sur lequel le service d'événements est déployé. Ce paramètre est obligatoire uniquement si vous spécifiez un noeud ; il n'est pas valide si vous supprimez le service Événements d'un cluster.

clusterName

Nom du cluster sur lequel le service d'événements est déployé. Ce paramètre est facultatif et ne doit pas être spécifié si vous supprimez le service Événements d'un serveur.

Suppression de l'application d'entreprise de messagerie d'événements

Pour supprimer la configuration de messagerie du service Événements pour un fournisseur JMS externe, vous pouvez utiliser la commande d'administration `removeEventServiceMdb`.

A propos de cette tâche

Cette commande permet de supprimer le bean géré par message déployé pour la configuration JMS. Pour supprimer le bean géré par message du service Événements :

Procédure

1. Démarrez l'outil `wsadmin`.
2. L'objet `AdminTask` permet d'exécuter la commande d'administration `removeEventServiceMdb` en mode interactif ou de traitement par lots. Les paramètres relatifs à la commande d'administration `removeEventServiceMdb` sont les suivants :

applicationName

Nom de l'application du bean géré par message du service Événements déployé.

nodeName

Nom du noeud sur lequel le bean géré par message du service Événements est déployé. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Ce paramètre est facultatif. Par défaut, le nom du noeud en cours est utilisé. N'indiquez pas ce paramètre si vous supprimez l'application d'un cluster.

serverName

Nom du serveur sur lequel le bean géré par message du service Événements est déployé. Ce paramètre est requis si vous supprimez l'application d'un serveur, sinon il est facultatif. N'indiquez aucun nom de serveur si vous supprimez l'application d'un cluster.

clusterName

Nom du cluster sur lequel le bean géré par message du service Événements est déployé. Spécifiez ce paramètre uniquement si vous supprimez l'application d'un cluster.

Chapitre 3. Administration de l'infrastructure CEI (Common Event Infrastructure)

Ces rubriques décrivent les tâches d'administration que vous pouvez effectuer en vue de contrôler le fonctionnement des composants Common Event Infrastructure lors de l'exécution.

Administration du service Événements à l'aide de la console d'administration

Vous pouvez utiliser la console d'administration Web pour effectuer les tâches d'administration relatives au service Événements.

Activation et désactivation du service Événements à l'aide de la console d'administration

Vous pouvez activer et désactiver le service Événements en modifiant les propriétés de ce service dans la console d'administration du serveur.

A propos de cette tâche

Si le service Événements est activé, il démarre automatiquement au moment du démarrage du serveur.

Pour activer ou désactiver le service Événements via la console d'administration :

Procédure

1. Sélectionnez l'une des méthodes suivantes pour accéder au panneau de la console d'administration appropriée :
 - Ouvrez le panneau de la console d'administration de Common Event Infrastructure Server
 - Pour les serveurs, sélectionnez **Serveurs** → **Serveurs d'application** → *nom_serveur* → **Business Integration** → **Common Event Infrastructure** → **Common Event Infrastructure - Serveur**.
 - Pour les clusters, sélectionnez **Serveurs** → **Clusters** → *nom_cluster* → **Business Integration** → **Common Event Infrastructure** → **Common Event Infrastructure - Serveur**.
 - Vous pouvez également accomplir cette tâche en ouvrant Services du conteneur :
 - Pour les serveurs, cliquez sur **Serveurs** → **Serveurs d'application** → *nom_serveur* → **Services du conteneur** → **Common Event Infrastructure Service**.
 - Pour les cluster, cliquez sur **Serveurs** → **Clusters** → *nom_cluster* → **Membres de cluster** → *serveur* → **Services du conteneur** → **Common Event Infrastructure Service**.
2. Sélectionnez ou désélectionnez la propriété **Activer le service au démarrage du serveur**. Si la case est cochée, le service Common Event Infrastructure sera activé lors du démarrage du serveur.
3. Sauvegardez les modifications apportées à la configuration.

4. Dans un environnement de déploiement réseau uniquement, synchronisez le noeud.
5. Redémarrez les serveurs ou le cluster.

Création d'une fabrique d'émetteurs d'événement à l'aide de la console d'administration

Une fabrique d'émetteurs permet aux sources d'événements de créer des émetteurs.

A propos de cette tâche

Les propriétés d'une fabrique d'émetteurs affectent le comportement de tout émetteur créé à l'aide de la fabrique d'émetteurs. Vous pouvez utiliser la fabrique d'émetteurs par défaut ou en créer d'autres qui seront utilisées par vos sources d'événements. Vous pouvez également vouloir créer une fabrique d'émetteurs supplémentaire pour spécifier une transmission d'événements ou un mode de transaction différent. Pour afficher l'émetteur d'événements créé lors de la configuration de CEI (Common Event Infrastructure) sur le panneau Common Event Infrastructure - Cible :

1.
 - Pour un serveur unique, sélectionnez **Serveurs > Serveurs d'application > nom_serveur**.
 - Pour un cluster, sélectionnez **Serveurs > Clusters > nom_cluster**.
2. Sur l'onglet **Configuration**, sélectionnez **Business Integration > Common Event Infrastructure > Common Event Infrastructure - Cible**.
3. Vous pouvez choisir une fabrique d'émetteurs d'événement existante dans le menu de noms JNDI ou en spécifier une dans la zone de texte.

Si le serveur CEI n'est pas local, vous devez résoudre les noms JNDI sur un serveur distant. Pour plus d'informations sur l'attribution de noms au sein d'un environnement ND, reportez-vous à la documentation WebSphere Application Server. Pour créer une fabrique d'émetteurs :

Procédure

1. Dans la console d'administration du serveur, cliquez sur **Intégration de services > Common Event Infrastructure > Fabriques d'émetteurs d'événement > Nouveau**.
2. Indiquez les propriétés de la nouvelle fabrique d'émetteurs. Pour plus de détails sur ces propriétés, voir l'aide en ligne de la page relative aux paramètres des fabriques d'émetteurs.
3. Sauvegardez les modifications apportées à la configuration.
4. Dans un environnement de déploiement réseau uniquement, synchronisez le noeud.
5. Redémarrez le serveur.

Résultats

Vous pouvez désormais utiliser la fabrique d'émetteurs configurée pour créer des émetteurs.

Création d'un groupe d'événements à l'aide de la console d'administration

Un groupe d'événements définit un ensemble logique d'événements en fonction du contenu de leurs données de propriétés. Un groupe d'événements peut être utilisé lors de l'interrogation d'événements du service d'événement. De façon facultative, il peut être associé à une destination JMS dans le cadre d'une distribution d'événements asynchrone.

A propos de cette tâche

Pour créer un groupe d'événements :

Procédure

1. **Facultatif** : Configurez une ou plusieurs destinations JMS pour le groupe d'événements. Un groupe d'événements peut être associé à une rubrique JMS et à une ou plusieurs files d'attente JMS. Consultez la documentation de votre fournisseur JMS pour en savoir plus sur la création de destinations et de fabriques de connexions JMS et leur liaison à un espace de noms JNDI.

Sécurité : Si la sécurité WebSphere est activée, un alias d'authentification doit être spécifié dans la configuration de la destination JMS.

2. Créez le groupe d'événements. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Services Événement > Services Événements > service_Evénements > Groupes d'événements > Nouveau**.
3. Indiquez les propriétés du nouveau groupe d'événements, y compris le sélecteur d'événements et les destinations JMS facultatives.
4. Sauvegardez les modifications apportées à la configuration.
5. Dans un environnement de déploiement réseau uniquement, synchronisez le noeud.
6. Redémarrez le serveur.

Résultats

Les consommateurs d'événements peuvent à présent indiquer le groupe d'événements lors de l'interrogation d'événements. Si la distribution d'événements est activée dans les paramètres du service Événements, les événements appartenant au groupe d'événements sont également publiés sur toutes les destinations JMS définies dans le groupe d'événements. Les consommateurs d'événements peuvent alors recevoir des événements de façon asynchrone en souscrivant aux destinations appropriées.

Création d'un filtre d'événements à l'aide de la console d'administration

Un filtre d'événements définit les propriétés qu'utilise le module d'extension de filtrage par défaut, utilisé par les émetteurs pour filtrer les événements à la source.

A propos de cette tâche

Un filtre d'événements peut être spécifié en tant qu'élément de la configuration d'une fabrique d'émetteurs. Tout émetteur créé à l'aide de cette fabrique utilise ensuite le filtre spécifié pour déterminer l'événement à envoyer au service Événements.

Pour créer un filtre d'événements :

Procédure

1. Dans la console d'administration du serveur, cliquez sur **Intégration de services > Common Event Infrastructure > Fabriques d'émetteur d'événements > fabrique_émetteur > Filtres d'événements > Nouveau**.
2. Indiquez les propriétés du nouveau filtre d'événements. Pour plus de détails sur ces propriétés, voir l'aide en ligne de la page relative aux paramètres des filtres d'événements.
3. Sauvegardez les modifications apportées à la configuration.
4. Dans un environnement de déploiement réseau uniquement, synchronisez le noeud.
5. Redémarrez le serveur.

Résultats

Les émetteurs d'événement peuvent à présent utiliser le filtre configuré pour déterminer les événements à envoyer au service Événements.

Administration du service Événements à l'aide de scripts

Vous pouvez utiliser les interfaces de scriptage pour effectuer les tâches d'administration relatives au service Événements.

Activation du service Événements à l'aide d'un script

Vous pouvez activer le service Événements en exécutant une commande d'administration AdminTask à l'aide de l'outil wsadmin.

A propos de cette tâche

Si le service Événements est activé, il démarre automatiquement au moment du démarrage du serveur.

Pour activer le service Événements à l'aide de l'outil wsadmin :

Procédure

1. Démarrez l'outil wsadmin.
2. Exécutez la commande d'administration **enableEventService** à l'aide de l'objet AdminTask :

- A l'aide de Jacl :

```
# enable event service at server scope
$AdminTask enableEventService { -nodeName noeud1 -serverName serveur1 }

# enable event service at cluster scope
$AdminTask enableEventService { -clusterName cluster1 }
```

- A l'aide de Jython :

```
# enable event service at server scope
AdminTask.enableEventService([' -nodeName noeud1 -serverName serveur1 ]')

# enable event service at cluster scope
AdminTask.enableEventService([' -clusterName cluster1 ]')
```

Les paramètres de la commande **enableEventService** sont les suivants :

nodeName

Nom du noeud sur lequel activer le service Événements. Ce paramètre est facultatif. Par défaut, le nom du noeud en cours est utilisé. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Ne spécifiez aucun noeud si vous activez le service Événements dans un cluster.

serverName

Nom du serveur sur lequel activer le service Événements. Ce paramètre est obligatoire si vous spécifiez un nom de noeud. Ne spécifiez aucun serveur si vous activez le service Événements dans un cluster.

clusterName

Nom du cluster sur lequel activer le service Événements. Ce paramètre est obligatoire si vous activez le service Événements dans un cluster.

3. Redémarrez le serveur.

Désactivation du service Événement à l'aide d'un script

Vous pouvez désactiver le service Événements en exécutant une commande d'administration AdminTask à l'aide de l'outil wsadmin.

A propos de cette tâche

Si le service Événements est désactivé, il ne démarre pas automatiquement au moment du démarrage du serveur.

Pour désactiver le service Événements à l'aide de l'outil wsadmin :

Procédure

1. Démarrez l'outil wsadmin.
2. Exécutez la commande d'administration **disableEventService** à l'aide de l'objet AdminTask :

- A l'aide de Jacl :

```
# disable event service at server scope
$AdminTask disableEventService { -nodeName noeud1 -serverName serveur1 }
```

```
# disable event service at cluster scope
$AdminTask disableEventService { -clusterName cluster1 }
```

- A l'aide de Jython :

```
# disable event service at server scope
AdminTask.disableEventService([' -nodeName noeud1 -serverName serveur1 '])
```

```
# disable event service at cluster scope
AdminTask.disableEventService([' -clusterName cluster1 '])
```

Les paramètres de la commande **disableEventService** sont les suivants :

nodeName

Nom du noeud sur lequel désactiver le service Événements. Ce paramètre est facultatif. Par défaut, le nom du noeud en cours est utilisé. Si vous indiquez un nom de noeud, vous devez également indiquer un nom de serveur. Ne spécifiez aucun noeud si vous désactivez le service Événements dans un cluster.

serverName

Nom du serveur sur lequel désactiver le service Événements. Ce paramètre est obligatoire si vous spécifiez un nom de noeud. Ne spécifiez aucun serveur si vous désactivez le service Événements dans un cluster.

clusterName

Nom du cluster sur lequel désactiver le service Événements. Ce paramètre est obligatoire si vous désactivez le service Événements dans un cluster.

3. Redémarrez le serveur.

Création d'une fabrique d'émetteurs à l'aide d'un script

Vous pouvez créer une fabrique d'émetteurs à l'aide d'un script Jacl ou Jython.

Avant de commencer

Avant d'exécuter cette tâche, assurez-vous que l'outil wsadmin est en cours d'exécution. Pour plus d'informations, reportez-vous à la documentation WebSphere Application Server.

A propos de cette tâche

Pour configurer une nouvelle fabrique d'émetteurs, suivez la procédure ci-dessous.

Procédure

1. Définissez l'ID du fournisseur Common Event Infrastructure.

- A l'aide de Jacl :

```
set providerid [$AdminConfig getid \  
/Cell:mycell/Node:mynode/Server:myserver/ \  
EventInfrastructureProvider:/]
```
- A l'aide de Jython :

```
providerid =  
AdminConfig.getid('/Cell:mycell/Node:mynode/Server:myserver/EventInfrastructureProvider/')  
print providerid
```

Exemple :

```
EventInfrastructureProvider(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#  
EventInfrastructureProvider_1)
```

2. Définissez les attributs requis.

- A l'aide de Jacl :

```
set Name [list name "EmitterName"]  
set JndiName [list jndiName "Nom JNDI de la nouvelle fabrique d'émetteurs"]  
set Description [list description "Description"]  
set Category [list category "Catégorie"]  
  
# set TransactionMode to true to send each event in a new transaction  
set TransactionMode [list preferredTransactionMode "false"]  
  
# set SynchronizationMode to true to use synchronous event transmission  
# as the preferred synchronization mode  
set SynchronizationMode [list preferredSynchronizationMode "true"]  
  
# leave blank if synchronous transmission is not supported  
set SyncJNDIName [list synchronousTransmissionProfileJNDIName \  
"Nom JNDI du profil de transmission synchrone"]  
  
# leave blank if asynchronous transmission is not supported  
set AsyncJNDIName [list asynchronousTransmissionProfileJNDIName \  
"Nom JNDI du profil de transmission asynchrone"]  
  
set FilteringEnabled [list filteringEnabled "false"]  
  
# leave blank if filtering is not enabled  
set FilterJNDIName [list filterFactoryJNDIName \  
"Nom JNDI du filtre d'événement"]
```

```

# custom properties include compatibility mode
set CompatibilityMode [list [list name compatibilityMode] \
  [list description ""] \
  [list required false] \
  [list type java.lang.Boolean] \
  [list value "false" ] \
set resProp [list [list resourceProperties [list \
  $CompatibilityMode ]]]

```

- A l'aide de Jython :

```

Name = ['name', 'EmitterName']
JndiName = ['jndiName', 'Nom JNDI de la nouvelle fabrique d'émetteurs']
Description = ['description', 'Description']
Category = ['category', 'Catégorie']

```

```

# set TransactionMode to true to send each event in a new transaction
TransactionMode = ['preferredTransactionMode', 'false']

```

```

# set SynchronizationMode to true to use synchronous event transmission
# as the preferred synchronization mode
SynchronizationMode = ['preferredSynchronizationMode', 'true']

```

```

# leave blank if synchronous transmission is not supported
SyncJNDIName = ['synchronousTransmissionProfileJNDIName', \
  'Nom JNDI du profil de transmission synchrone']

```

```

# leave blank if asynchronous transmission is not supported
AsyncJNDIName = ['asynchronousTransmissionProfileJNDIName', \
  'Nom JNDI du profil de transmission asynchrone']

```

```

FilteringEnabled = ['filteringEnabled', 'false']

```

```

# leave blank if filtering is not enabled
FilterJNDIName = ['filterFactoryJNDIName', 'Nom JNDI du filtre d'événement']

```

```

# custom properties include compatibility mode
compatibilityName = ['name', 'compatibilityMode']
compatibilityDescription = ['description', '']
compatibilityRequired = ['required', 'false']
compatibilityType = ['type', 'java.lang.Boolean']
compatibilityValue = ['value', 'false']
CompatibilityMode = [compatibilityName, compatibilityDescription, compatibilityRequired, \
  compatibilityType, compatibilityValue]
customProperties = ['propertySet', [['resourceProperties', [CompatibilityMode]]]]

```

3. Définissez les propriétés de la nouvelle fabrique d'émetteurs.

- A l'aide de Jacl :

```

set properties "[list $Name $JndiName $Description $Category \
  $TransactionMode $SynchronizationMode $AsyncJNDIName \
  $SyncJNDIName $FilteringEnabled $FilterJNDIName \
  [list propertySet $resProp]]"

```

- A l'aide de Jython :

```
properties = [Name,JndiName,Description,Category,TransactionMode,
SynchronizationMode,AsyncJNDIName,SyncJNDIName,FilteringEnabled, FilterJNDIName, customProperties]
print properties
```

Exemple :

```
[[['name', 'EmitterName'], ['jndiName', 'Nom JNDI de la nouvelle
fabrique d'émetteurs'], ['description', 'Description'],
['category', 'Catégorie'], ['preferredTransactionMode',
'false'], ['preferredSynchronizationMode', 'true'],
['asynchronousTransmissionProfileJNDIName', 'Nom JNDI du profil
de transmission asynchrone '],
['synchronousTransmissionProfileJNDIName', 'Nom JNDI du profil
de transmission synchrone '], ['filteringEnabled',
'false'], ['filterFactoryJNDIName', 'Nom JNDI du filtre
d'événement'], ['propertySet', [['resourceProperties', [[['name',
'compatibilityMode'], ['description', ''], ['required', 'false'],
['type', 'java.lang.Boolean'], ['value', 'false']]]]]]]]]
```

4. Créez la fabrique d'émetteurs.

- A l'aide de Jacl :

```
set emitterProf [$AdminConfig create EmitterFactoryProfile \
$providerid $properties]
```

- A l'aide de Jython :

```
print AdminConfig.create('EmitterFactoryProfile', providerid, properties)
```

Exemple :

```
EmitterName(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#EmitterFactoryProfile_1)
```

5. Sauvegardez les modifications apportées à la configuration.
6. Dans un environnement de déploiement réseau uniquement, synchronisez le noeud.
7. Redémarrez le serveur.

Création d'un groupe d'événements à l'aide d'un script

Vous pouvez créer un groupe d'événements à l'aide d'un script Jacl ou Jython.

Avant de commencer

Avant d'exécuter cette tâche, assurez-vous que l'outil wsadmin est en cours d'exécution. Pour plus d'informations, reportez-vous à la documentation WebSphere Application Server.

A propos de cette tâche

Pour configurer un nouveau groupe d'événements, suivez la procédure ci-dessous.

Procédure

1. Définissez l'ID du fournisseur Common Event Infrastructure.

- A l'aide de Jacl :

```
set providerid [$AdminConfig getid \  
/Cell:mycell/Node:mynode/Server:myserver/ \  
EventInfrastructureProvider:/]
```

- A l'aide de Jython :

```
providerid = \  
AdminConfig.getid('/Cell:mycell/Node:mynode/Server:myserver/EventInfrastructureProvider:') \  
print providerid
```

Exemple :

```
EventInfrastructureProvider(cells/mycell/nodes/mynode/servers/myserver| \  
resources-cei.xml#EventInfrastructureProvider_1)
```

2. Récupérez la liste des groupes d'événements.

- A l'aide de Jacl :

```
set eventGroupProfileId [lindex [$AdminConfig list EventGroupProfileList $providerid] 0]
```

- A l'aide de Jython :

```
eventGroupProfileId = AdminConfig.list('EventGroupProfileList',providerid)
```

3. Définissez les attributs requis.

- A l'aide de Jacl :

```
set name [ list eventGroupName "EventGroupName" ] \  
 \  
# escape all '[' characters in the event selector string. \  
set selectorString [ list eventSelectorString "Sélecteur d'événements" ] \  
 \  
# leave blank if events should not be published using JMS \  
set JNDIName [ list topicJNDIName "Nom JNDI de rubrique" ] \  
 \  
# leave blank if events should not be published using JMS \  
set connectionFactoryJNDIName [ list topicConnectionFactoryJNDIName \  
"Nom JNDI de la fabrique de connexions de rubrique" ] \  
 \  
set persistEventsFlag [ list persistEvents "true" ] \  
 \  
# custom properties include compatibility mode \  
set CompatibilityMode [list [list name compatibilityMode] \  
[list description "" ] \  
[list required false] \  
[list type java.lang.Boolean] \  
[list value "false" ] ] \  
set resProp [list [list resourceProperties [list \  
$CompatibilityMode ]]]
```


- A l'aide de Jython :

```
Name = ['eventGroupName', 'EventGroupName']
SelectorString = ['eventSelectorString', 'Sélecteur d'événements']

# leave blank if events should not be published using JMS
JNDIName = ['topicJNDIName', 'Nom JNDI de rubrique']

# leave blank if events should not be published using JMS
ConnectionFactoryJNDIName = ['topicConnectionFactoryJNDIName', \
    'Nom JNDI de la fabrique de connexions de rubrique']

PersistEventsFlag = ['persistEvents', 'true']

# custom properties include compatibility mode
compatibilityName = ['name', 'compatibilityMode']
compatibilityDescription = ['description', '']
compatibilityRequired = ['required', 'false']
compatibilityType = ['type', 'java.lang.Boolean']
compatibilityValue = ['value', 'false']
CompatibilityMode = [compatibilityName, compatibilityDescription, compatibilityRequired, \
    compatibilityType, compatibilityValue]
customProperties = ['propertySet', [['resourceProperties', [CompatibilityMode]]]]
```

4. Définissez les propriétés du nouveau groupe d'événements.

- A l'aide de Jacl :

```
set properties [ list $name $selectorString $JNDIName \
    $connectionFactoryJNDIName $persistEventsFlag [list propertySet \
    $resProp]]
```

- A l'aide de Jython :

```
properties = [Name,SelectorString,JNDIName,ConnectionFactoryJNDIName, \
    PersistEventsFlag,customProperties]
```

Exemple :

```
[[['eventGroupName', 'EventGroupName'], ['eventSelectorString',
'Sélecteur d'événements'], ['topicJNDIName', 'Nom JNDI
de rubrique'], ['topicConnectionFactoryJNDIName', 'Nom JNDI
de fabrique de connexions de rubrique'], ['persistEvents','true'],
['propertySet', [['resourceProperties', [[['name',
'compatibilityMode'], ['description', ''], ['required', 'false'],
['type', 'java.lang.Boolean'], ['value', 'false']]]]]]]]
```

5. Créez le groupe d'événements.

- A l'aide de Jacl :

```
set result [ $AdminConfig create EventGroupProfile $eventGroupProfileId $properties ]
```

- A l'aide de Jython :

```
print AdminConfig.create('EventGroupProfile', eventGroupProfileId, properties)
```

Exemple :

```
(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#EventGroupProfile_1)
```

6. Sauvegardez les modifications apportées à la configuration.

7. Dans un environnement de déploiement réseau uniquement, synchronisez le noeud.

8. Redémarrez le serveur.

Création d'un filtre d'événements à l'aide d'un script

Vous pouvez créer un filtre d'événements à l'aide d'un script Jacl ou Jython.

Avant de commencer

Avant d'exécuter cette tâche, assurez-vous que l'outil wsadmin est en cours d'exécution. Pour plus d'informations, reportez-vous à la documentation WebSphere Application Server.

A propos de cette tâche

Pour configurer un nouveau filtre d'événements, suivez la procédure ci-dessous.

Procédure

1. Définissez l'ID du fournisseur Common Event Infrastructure.

- A l'aide de Jacl :

```
set providerid [$AdminConfig getid \  
/Cell:mycell/Node:mynode/Server:myserver/ \  
EventInfrastructureProvider:/]
```
- A l'aide de Jython :

```
providerid =  
AdminConfig.getid('/Cell:mycell/Node:mynode/Server:myserver/EventInfrastructureProvider:/')  
print providerid
```

Exemple :

```
EventInfrastructureProvider(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#  
EventInfrastructureProvider_1)
```

2. Définissez les attributs requis.

- A l'aide de Jacl :

```
set Name [list name "EventFilterName"]  
set JndiName [list jndiName "Nom JNDI du nouveau filtre d'événements"]  
set Description [list description "Description du nouveau  
filtre d'événements"]  
set Category [list category "Catégorie du filtre d'événements"]
```

```
# escape all '[' characters in the configuration string  
# (for example, "CommonBaseEvent\[@severity=50\]"  
set filterConfigurationString [list filterConfigurationString \  
"Set filter configuration string here"]
```

- A l'aide de Jython :

```
Name = ['name', 'EventFilterName']  
JndiName = ['jndiName', 'Nom JNDI du nouveau filtre d'événements']  
Description = ['description', 'Description du filtre d'événements']  
Category = ['category', 'Catégorie du filtre d'événements']  
FilterConfigurationString = ['filterConfigurationString', \  
'Chaîne de configuration du filtre']
```

3. Définissez les propriétés du nouveau filtre d'événements.

- A l'aide de Jacl :

```
set properties [list $name $jndiName $description $category \  
$filterConfigurationString]
```
- A l'aide de Jython :

```
properties = [Name,JndiName,Description,Category,FilterConfigurationString]  
print properties
```

Exemple :

```
[[['name', 'EventFilterName'], ['jndiName', 'Nom JNDI du nouveau
filtre d'événements'], ['description', 'Description du filtre
d'événements'], ['category', 'Catégorie du filtre
d'événements'], ['filterConfigurationString', 'Chaîne de configuration
du filtre']]
```

4. Créez le filtre d'événements.

• A l'aide de Jacl :

```
set filterProf [$AdminConfig create FilterFactoryProfile \
$providerid $properties]
```

• A l'aide de Jython :

```
print AdminConfig.create('FilterFactoryProfile', providerid, properties)
```

Exemple :

```
EventFilterName(cells/mycell/nodes/mynode/servers/myserver|resources-cei.xml#FilterFactoryProfile_1)
```

5. Sauvegardez les modifications apportées à la configuration.

6. Dans un environnement de déploiement réseau uniquement, synchronisez le noeud.

7. Redémarrez le serveur.

Journalisation et fonction de trace des composants Common Event Infrastructure

Vous pouvez activer la journalisation et la fonction de trace afin de déboguer des incidents à l'aide de Common Event Infrastructure.

A propos de cette tâche

Les composants Common Event Infrastructure utilisent la structure de journalisation JSR47 Java, disponible dans les environnements serveur et client Common Event Infrastructure. Pour plus d'informations sur l'utilisation de la structure de journalisation, reportez-vous à la documentation sur la résolution des incidents de WebSphere Application Server Network Deployment, version 6.1.

Le tableau suivant indique le nom des enregistreurs chronologiques automatiques par les composants de l'infrastructure CEI.

Tableau 2. Nom des enregistreurs chronologiques automatiques

Composant	Nom de l'enregistreur chronologique automatique
Nom de l'enregistreur chronologique automatique racine	com.ibm.events
Catalogue d'événements	com.ibm.events.catalog
Sous-composants du service Evénements	com.ibm.events.access com.ibm.events.bus com.ibm.events.distribution com.ibm.events.server
Module d'extension du magasin de données par défaut	com.ibm.events.datastore
Emetteur d'événements	com.ibm.events.emitter
Assistant de notifications	com.ibm.events.notification
Configuration	com.ibm.events.configuration com.ibm.events.admintask

Tableau 2. Nom des enregistreurs chronologiques automatiques (suite)

Composant	Nom de l'enregistreur chronologique automatique
Configuration de la base de données	com.ibm.events.install.db
Migration	com.ibm.events.migration
Utilitaires divers	com.ibm.events.til

Maintenance de la base de données d'événements

Si vous utilisez une base de données d'événements externe, vous devez régulièrement procéder à sa maintenance en exécutant les scripts fournis.

Mise à jour des statistiques relatives à la base de données d'événements DB2

Pour activer une base de données DB2 sous Linux, UNIX ou Windows en vue d'optimiser des requêtes et récupérer l'espace disponible, vous pouvez mettre à jour les statistiques de la base de données à l'aide du script **runstats**.

A propos de cette tâche

Il est recommandé de régulièrement mettre à jour des statistiques de la base de données DB2, et particulièrement dans les cas suivants :

- Des événements ont été supprimés de la base de données, à l'aide des interfaces relatives à la suppression des événements du service Événements ou via l'utilitaire de purge d'événements du plug-in de stockage des données par défaut
- Un grand nombre d'événements ont été insérés dans la base de données
- Les tables ont été réorganisées à l'aide du script **reorg**
- Des index ont été ajoutés ou supprimés d'une table

Le script **runstats** est disponible dans le répertoire *racine_profil/event/nom_noeud/nom_serveur* ou dans le répertoire *nom_cluster/dbscripts/db2*.

Procédure

Pour mettre à jour les statistiques de la base de données, exécutez l'une des commandes suivantes :

- **Windows** Sur les systèmes Windows :
`runstats.bat utilisateur_base_de_données [password=mot_de_passe_base_de_données]`
- **Linux** **UNIX** Sur les systèmes Linux et UNIX :
`runstats.sh utilisateur_base_de_données [password=mot_de_passe_base_de_données]`

Les paramètres utilisés sont les suivants :

utilisateur_base_de_données

ID utilisateur de la base de données à indiquer. Ce paramètre est obligatoire.

mot_de_passe_base_de_donnees

Mot de passe de la base de données. Ce paramètre est facultatif. Si vous ne spécifiez pas de mot de passe sur la ligne de commande, la base de données DB2 vous invite à le faire.

Par exemple, l'exécution de la commande suivante permet de mettre à jour les statistiques de la base de données DB2 sur un système Windows, à l'aide de l'ID utilisateur de base de données dbadmin et du mot de passe motdepasse :

```
mrunstats.bat dbadmin motdepasse
```

Réorganisation des tables de bases de données d'événements DB2

Une fois que les événements sont purgés ou supprimés d'une base de données d'événements DB2 sous Linux, UNIX ou Windows, vous pouvez réorganiser les tables de bases de données à l'aide du script **reorg**.

A propos de cette tâche

Le script **reorg** est disponible dans le répertoire *racine_profil/event/nom_noeud/nom_serveur* ou dans le répertoire *nom_cluster/dbscripts/db2*.

Procédure

Pour réorganiser les tables de la base de données des événements, exécutez l'une des commandes suivantes :

- **Windows** Sur les systèmes Windows :

```
reorg.bat alias_base_de_donnees utilisateur_base_de_donnees  
[password=mot_de_passe_base_de_donnees]
```
- **Linux** **UNIX** Sur les systèmes Linux et UNIX :

```
reorg.sh alias_base_de_donnees utilisateur_base_de_donnees  
[password=mot_de_passe_base_de_donnees]
```

Les paramètres utilisés sont les suivants :

alias_base_de_donnees

Alias de la base de données. La base de données d'événements doit être cataloguée sur le client DB2 ; si vous exécutez le script sur le serveur DB2, la base de données est déjà cataloguée.

utilisateur_base_de_donnees

ID utilisateur de la base de données à indiquer. Ce paramètre est obligatoire.

mot_de_passe_base_de_donnees

Mot de passe de la base de données. Ce paramètre est facultatif. Si vous ne spécifiez pas de mot de passe sur la ligne de commande, la base de données DB2 vous invite à le faire.

Par exemple, la commande suivante permet de réorganiser les tables de la base de données d'événements sur un système Windows, à l'aide de l'alias de base de données eventdb, de l'ID utilisateur dbadmin et du mot de passe motdepasse :

```
reorg.bat eventdb dbadmin motdepasse
```

Une fois le script **reorg** exécuté, vous devez mettre à jour les statistiques de la base de données à l'aide du script **runstats**. Pour plus d'informations, reportez-vous à la rubrique «Mise à jour des statistiques relatives à la base de données d'événements DB2», à la page 68.

Purge des événements de la base de données d'événements

Vous pouvez exécuter les scripts fournis pour purger rapidement un nombre important d'événements de la base de données d'événements.

A propos de cette tâche

Le module d'extension du magasin de données par défaut fournit un ensemble d'utilitaires que vous pouvez exécuter pour effectuer une purge périodique d'un grand nombre d'anciens événements de la base de données d'événements. Ces utilitaires sont différents de la commande du service Événements `eventpurge`, qui supprime les événements correspondant aux critères spécifiés.

Bases de données Derby : L'utilitaire de purge de base de données n'est pas pris en charge pour une base de données d'événements Derby.

La capacité de purge de la base de données fait appel au concept de *compartiments*. Un compartiment est un ensemble de tables utilisé pour stocker les événements dans la base de données d'événements. Le magasin de données par défaut en utilise deux :

- Le compartiment actif est celui qui contient les événements les plus récents ; les nouveaux événements sont stockés dans ce compartiment. Le compartiment actif ne peut pas être purgé à l'aide de l'utilitaire de purge de base de données.
- Le compartiment inactif contient des événements plus anciens. Les événements stockés dans ce compartiment peuvent être interrogés, supprimés ou modifiés, mais aucun événement récent n'y est stocké. Le compartiment inactif peut être purgé par l'utilitaire de purge de base de données.

Chaque événement est stocké dans un seul compartiment. Du point de vue d'un consommateur d'événements, il n'y a aucune différence entre le compartiment actif et le compartiment inactif : il peut interroger, modifier ou supprimer un événement sans savoir dans quel compartiment cet événement est stocké. L'avantage de cette approche est que le compartiment inactif peut être purgé à l'aide d'interfaces spécifiques de bases de données sans affecter le compartiment actif. Le trafic normal d'événements peut continuer même si une opération de purge est en cours.

Une fois le compartiment inactif purgé, vous pouvez échanger les compartiments de sorte que le compartiment actif devienne inactif et inversement. L'échange de compartiments n'est possible que lorsque le compartiment inactif est vide.

Bien que les nouveaux événements soient généralement stockés dans le compartiment actif, il peut arriver dans certains cas qu'ils le soient dans le compartiment inactif, immédiatement après l'échange de compartiments. Le module d'extension du magasin de données détermine régulièrement lequel des compartiments est actif, mais tant que la vérification n'a pas eu lieu, certains événements peuvent être placés dans le compartiment inactif. De la même façon, les événements envoyés au sein d'un lot sont stockés dans le même compartiment, même si ce dernier devient inactif alors que le lot est en cours de traitement.

Si vous voulez utiliser la fonction de purge rapide, c'est vous qui devez déterminer la fréquence d'échange des compartiments et de purge du compartiment inactif, en fonction du trafic des événements, de l'espace de stockage, des exigences d'archivage, ou autre.

Affichage ou modification du statut du compartiment actif de la base de données d'événements

Le statut de compartiment actif indique le compartiment actuellement actif et le compartiment actuellement inactif.

Procédure

Pour afficher ou modifier le statut du comportement actif, utilisez la commande `eventbucket` :

```
eventbucket [-status] [-change]
```

Cette commande dispose des options suivantes :

-status

Utilisez cette options pour afficher des informations sur la configuration du compartiment actuel, y compris les paramètres du compartiment actif et la fréquence de vérification du compartiment (fréquence à laquelle de module d'extension de magasin de données identifie le compartiment actif).

-change

Utilisez cette option pour échanger les compartiments actif et inactif. Le compartiment inactif doit être vide avant de pouvoir utiliser cette option.

Purge du compartiment inactif de la base de données d'événements

La méthode utilisée pour purger le compartiment inactif dépend du logiciel de base de données utilisé.

A propos de cette tâche

Base de données Derby : L'utilitaire de purge rapide n'est pas pris en charge pour une base de données d'événements Derby.

Purge du compartiment inactif d'une base de données d'événements DB2(systèmes Linux, UNIX ou Windows) :

Sur les systèmes Linux, UNIX et Windows, l'utilitaire de purge d'une base de données DB2 est implémenté en tant que fichier batch ou script shell.

Procédure

Pour purger le compartiment inactif, exécutez l'une des commandes suivantes :

- Pour purger le compartiment inactif, exécutez la commande suivante :

```
fastpurge aliasBD utilisateurBD [password=motdepasseBD] [copydir=repCopie]
```

Les paramètres de cette commande sont les suivants :

aliasBD

Alias de la base de données. La base de données d'événements doit être cataloguée sur le client DB2 ; si vous exécutez le script sur le serveur DB2, la base de données est déjà cataloguée.

utilisateurBD

ID utilisateur de la base de données à utiliser lors de la connexion à la base de données d'événements.

motdepasseBD

Mot de passe à utiliser avec l'ID utilisateur spécifié. Ce paramètre est facultatif. Si vous ne spécifiez pas de mot de passe, la base de données DB2 vous invite à le faire.

repCopie

Chemin vers le répertoire à utiliser pour les fichiers générés par l'utilitaire de chargement. Ce paramètre est obligatoire uniquement si vous avez activé la reprise aval pour la base de données d'événements (avec les paramètres de configuration LOGRETAIN ou USEREXIT). Par défaut, la base de données d'événements n'utilise pas la reprise aval.

Purge du compartiment inactif d'une base de données d'événements DB2 (systèmes z/OS) :

Sur les systèmes z/OS, l'utilitaire de purge d'une base de données d'événements DB2 est implémentée à l'aide de l'utilitaire de charge DB2.

A propos de cette tâche

Pour purger le compartiment inactif :

Procédure

1. Exécutez la commande eventbucket pour identifier le compartiment inactif (compartiment 0 ou 1).
2. Téléchargez le fichier de contrôle d'utilitaire approprié. Ces fichiers sont générés lors de la configuration de la base de données. Ils se trouvent dans le répertoire *racine_profil/event/dbscripts/db2zos*. Téléchargez l'un des fichiers suivants :
 - Si le compartiment 0 est inactif, téléchargez le fichier fastpurge00.ctl.
 - Si le compartiment 1 est inactif, téléchargez le fichier fastpurge01.ctl.

Format de téléchargement : Le fichier de contrôle doit être téléchargé sous un format d'enregistrement fixe et avec une longueur d'enregistrement logique de 80.

3. Sur l'hôte z/OS, go accédez au menu ISPF DB2I Primary Option et sélectionnez l'option **Utilities**.
4. Spécifiez les informations suivantes :

Zone	Valeur
Function	EDITJCL
Utility	LOAD
Statement Data Set	Nom de l'ensemble de données contenant le fichier de contrôle téléchargé
LISTDEF	NO
Template	NO

5. Appuyez sur Entrée pour passer à l'écran suivant.
6. Dans la zone de saisie recdsn, entrez le nom de l'ensemble de données contenant le fichier de contrôle téléchargé.

7. Appuyez sur Entrée. Le script JCL permettant de purger le compartiment inactif est généré.
8. Appuyez sur Entrée pour effacer les messages de sortie.
9. Editez le script JCL généré comme nécessaire.
10. Soumettez le script JCL.

Résultats

Le script JCL généré peut être réutilisé. Une fois créés les scripts permettant de purger les deux compartiments (0 et 1), vous n'avez plus besoin de répéter toute cette procédure.

Purge du compartiment inactif d'une base de données d'événements DB2 event database (systèmes iSeries) :

Sur les systèmes iSeries, l'utilitaire de purge d'une base de données d'événements DB2 est implémenté en tant que procédure stockée.

A propos de cette tâche

Pour purger le compartiment inactif :

Procédure

1. Connectez-vous au système iSeries à l'aide d'une application émulant un terminal 5250, par exemple, IBM Personal Communications.
2. Connectez-vous avec un ID utilisateur disposant de droits suffisants pour exécuter la procédure stockée de purge rapide. Il est recommandé d'utiliser l'ID utilisateur qui a servi à créer la base de données d'événements.
3. Exécutez la commande **strsql**.
4. Entrez l'instruction SQL suivante :

```
call collection/fast_purge()
```

collection est le nom de la collection contenant la base de données d'événements. La connexion par défaut est event :

```
call event/fast_purge()
```

Purge du compartiment inactif d'une base de données d'événements Oracle :

L'utilitaire de purge pour une base de données d'événements Oracle est mis en oeuvre en tant que procédure mémorisée.

Avant de commencer

Pour utiliser l'utilitaire pour une base de données Oracle, vous devez avoir installé SQL*Plus sur le client Oracle, et le client doit être configuré pour communiquer avec la base de données Oracle (le fichier tnsnames.ora doit être configuré correctement). Voir la documentation Oracle pour plus d'informations.

Procédure

Pour purger le compartiment inactif, exécutez la procédure mémorisée à l'aide de SQL*Plus :

```
sqlplus chaîne_connexion @fastpurge.sql
```

Le fichier `fastpurge.sql` figure au même emplacement que les autres scripts générés lors de la configuration de la base de données (emplacement par défaut `racine_profil/databases/event/nom_noeud/nom_serveur` ou `nom_cluster/dbscripts/oracle`).

Le paramètre `chaîne_connexion` correspond à la chaîne de connexion d'Oracle. Utilisez le même ID utilisateur de base de données pour créer la base de données d'événements.

Purge du compartiment inactif d'une base de données d'événements Informix :

L'utilitaire de purge d'une base de données d'événements Informix est implémenté en tant que procédure stockée.

Avant de commencer

Pour pouvoir utiliser l'utilitaire pour une base de données Informix, vous devez exécuter la commande Informix `dbaccess` dans un environnement dont la source a été correctement définie pour les variables d'environnement Informix. Pour plus d'informations, reportez-vous à la documentation Informix.

Procédure

1. Pour purger le compartiment inactif, exécutez la procédure stockée à l'aide de `dbaccess` :

```
dbaccess - fastpurge.sql
```

Le fichier `fastpurge.sql` figure au même emplacement que les autres scripts générés lors de la configuration de la base de données (emplacement par défaut `:racine_profil/databases/event/noeud/serveur/dbscripts/informix`).

2. Exécutez le script `eventpurgepool` pour purger le pool de connexions :

- **Windows** Sur les systèmes Windows : `eventpurgepool.bat`
- **Linux** **UNIX** Sur les systèmes Linux ou Unix : `eventpurgepool.sh`

Purge du compartiment actif d'une base de données d'événements SQL :

L'utilitaire de purge pour une base de données d'événements SQL Server est mis en oeuvre en tant que procédure mémorisée.

Avant de commencer

Pour utiliser l'utilitaire pour une base de données SQL Server, vous devez avoir installé l'utilitaire `osql` sur le client SQL Server. Pour plus d'informations, reportez-vous à la documentation SQL Server.

Remarque : N'utilisez pas l'utilitaire `isql` pour exécuter la procédure mémorisée de purge de la base de données.

Procédure

Pour purger le compartiment inactif, exécutez la procédure mémorisée à l'aide de l'utilitaire `osql` :

```
osql -Snom_serveur -UutilisateurBD -PmotdepasseBD -dBDévénements -Q"fast_purge"
```

Le fichier `fast_purge` figure au même emplacement que les autres scripts générés lors de la configuration de la base de données (emplacement par défaut : `:racine_profil/databases/event/noeud/serveur/dbscripts/sqlserver`).

Les paramètres de cette commande sont les suivants :

nom_serveur

Nom du serveur de base de données.

utilisateurBD

ID utilisateur de la base de données à utiliser lors de la connexion à la base de données d'événements.

motdepasseBD

Mot de passe à utiliser avec l'ID utilisateur spécifié. Ce paramètre est facultatif. Si vous ne spécifiez pas de mot de passe, l'utilitaire `osql` vous le demandera.

BDÉvénements

Nom de la base de données d'événements (généralement événement).

Modification de la fréquence de vérification du compartiment de la base de données d'événements

La fréquence de vérification du compartiment spécifie la fréquence à laquelle le plug-in du magasin de données détermine lequel des compartiments est actif. Cette valeur est spécifiée en tant que propriété personnalisée dans les paramètres relatifs au magasin de données.

A propos de cette tâche

La fréquence de vérification du compartiment par défaut est de 5 minutes (300 secondes). Une fréquence plus importante réduit la probabilité que les événements soient stockés dans le compartiment inactif après un échange, mais peut altérer les performances.

Pour modifier la fréquence de vérification du compartiment :

Procédure

1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Service Événements > Services Événements > service_événements > Magasin de données d'événement**.
2. Modifiez la valeur de la propriété `BucketCheckInterval` pour spécifier une fréquence de vérification du compartiment en secondes.

Résultats

Les modifications seront appliquées au prochain démarrage du serveur.

Chapitre 4. Sécurisation de l'accès aux fonctions de Common Event Infrastructure

Vous pouvez utiliser la sécurité déclarative au niveau des méthodes WebSphere pour restreindre l'accès aux fonctions de Common Event Infrastructure.

Le Common Event Infrastructure définit sept rôles de sécurité, dont chacun est associé à un groupe de fonctions. Ces rôles de sécurité contrôlent l'accès aux interfaces et aux commandes de programmation.

Le tableau suivant décrit les rôles de sécurité et les types d'utilisateurs associés à chaque rôle.

Tableau 3. Rôles de sécurité et types d'utilisateurs

Rôle de sécurité	Types d'utilisateurs
eventCreator	<p>Sources d'événements devant soumettre des événements à un émetteur, au moyen d'appels EJB synchrones. Ce rôle permet d'accéder aux interfaces suivantes :</p> <ul style="list-style-type: none">• Emitter.sendEvent()• Emitter.sendEvents()• Commande eventemit <p>Si l'émetteur est configuré pour utiliser des appels EJB synchrones pour la transmission des événements, le rôle eventCreator donne accès uniquement à la soumission d'événements. En revanche, si l'émetteur utilise des messages JMS asynchrones, vous devez utiliser la sécurité JMS pour restreindre l'accès à la destination utilisée pour soumettre les événements.</p>
eventUpdater	<p>Destinataires d'événements devant mettre à jour les événements stockés dans la base de données d'événements. Ce rôle permet d'accéder aux interfaces suivantes :</p> <ul style="list-style-type: none">• EventAccess.updateEvents()• EventAccess.eventExists()• EventAccess.queryEventByGlobalInstanceId()• EventAccess.queryEventsByAssociation()• EventAccess.queryEventsByEventGroup()• Commande eventquery
eventConsumer	<p>Destinataires d'événements devant interroger les événements stockés dans la base de données d'événements. Ce rôle permet d'accéder aux interfaces suivantes :</p> <ul style="list-style-type: none">• EventAccess.eventExists()• EventAccess.queryEventByGlobalInstanceId()• EventAccess.queryEventsByAssociation()• EventAccess.queryEventsByEventGroup()• Commande eventquery

Tableau 3. Rôles de sécurité et types d'utilisateurs (suite)

Rôle de sécurité	Types d'utilisateurs
eventAdministrator	<p>Destinataires d'événements devant interroger, mettre à jour et supprimer des événements stockés dans la base de données d'événements. Ce rôle permet d'accéder aux interfaces suivantes :</p> <ul style="list-style-type: none"> • EventAccess.purgeEvents() • EventAccess.eventExists() • EventAccess.queryEventByGlobalInstanceId() • EventAccess.queryEventsByAssociation() • EventAccess.queryEventsByEventGroup() • EventAccess.updateEvents() • Emitter.sendEvent() • Emitter.sendEvents() • Commande eventquery • Commande eventpurge • Commande eventemit • Commande eventbucket
catalogReader	<p>Applications du catalogue d'événements devant extraire des définitions d'événements du catalogue. Ce rôle permet d'accéder aux interfaces suivantes :</p> <ul style="list-style-type: none"> • EventCatalog.getAncestors() • EventCatalog.getChildren() • EventCatalog.getDescendants() • EventCatalog.getEventDefinition() • EventCatalog.getEventDefinitions() • EventCatalog.getEventExtensionNamesForSourceCategory() • EventCatalog.getEventExtensionToSourceCategoryBindings() • EventCatalog.getParent() • EventCatalog.getRoot() • EventCatalog.getSourceCategoriesForEventExtension() • Commande eventcatalog (option -listdefinitions) • Commande eventcatalog (option -listcategories) • Commande eventcatalog (option -exportdefinitions)
catalogAdministrator	<p>Applications du catalogue d'événements devant créer, mettre à jour, supprimer ou extraire des définitions d'événements figurant dans le catalogue. Ce rôle permet d'accéder à toutes les méthodes de l'interface EventCatalog et à toutes les fonctions de la commande eventcatalog. Etant donné que les modifications du catalogue peuvent générer des événements, ce rôle donne également accès aux interfaces de soumission d'événements.</p>

Le bean géré par message du service Événements s'exécute via l'identité de l'utilisateur du serveur. Si vous utilisez la transmission JMS asynchrone pour soumettre des événements au service Événements et si vous avez activé la sécurité basée sur les méthodes, vous devez mapper cette identité utilisateur au rôle eventCreator.

Sécurité : Si votre source d'événements s'exécute avec la sécurité Java 2 activée et que vous souhaitez générer vos propres GUID (globally unique identifiers), vous devez modifier le fichier de règles pour permettre un traitement correct. Ajoutez les entrées suivantes :

```
permission java.io.FilePermission "${java.io.tmpdir}${/}guid.lock",  
    "read, write, delete";  
permission java.net.SocketPermission "*", "resolve";
```

Chapitre 5. Identification et résolution des incidents liés à Common Event Infrastructure

Ces rubriques offrent des informations de résolution des incidents relatives au service Événements et basées sur la tâche ou l'activité effectuée lorsque ce problème s'est produit.

Problèmes lors du démarrage

Le service Événements ne démarre pas (message CEIDS0058E)

Le service Événements ne démarre pas et envoie le message CEIDS0058E au fichier journal WebSphere.

Cause

Le service Événements utilise des instructions SQL avec le nom d'utilisateur. Cette erreur signale que le nom d'utilisateur employé par le service Événements pour se connecter à la base de données d'événements n'est pas identique à l'ID utilisateur employé pour créer la base de données.

Solution

L'ID utilisateur employé pour se connecter à la base de données d'événements doit être identique à celui utilisé pour créer cette base de données. Pour remédier à cette erreur :

1.
 - Pour un serveur unique, sélectionnez **Serveurs > Serveurs d'application > nom_serveur**.
 - Pour un cluster, sélectionnez **Serveurs > Clusters > nom_cluster**.
2. Sur l'onglet **Configuration**, sélectionnez **Business Integration > Common Event Infrastructure > Common Event Infrastructure - Serveur**.
3. Modifiez l'ID et le mot de passe utilisateur spécifiés afin qu'ils correspondent à ceux spécifiés pour créer la base de données.
4. Sauvegardez les modifications apportées à la configuration.
5. Redémarrez le serveur.

Problèmes relatifs à l'envoi d'événements

Erreur lors de l'envoi de l'événement (message CEIDS0060E)

Ma source d'événements rencontre une erreur lors de la tentative d'envoi d'un événement, et le message CEIDS0060E s'affiche dans le fichier journal WebSphere.

Cause

Le service Événements utilise des métadonnées stockées dans la base de données d'événements afin de mapper des éléments et des attributs Common Base Event aux colonnes et aux tables de base de données. Ces informations sont lues dans la base de données lorsque l'application tente pour la première fois d'utiliser le service Événements suite au démarrage.

Les tables de métadonnées sont renseignées lors de la création de la base de données d'événements. Cette erreur survient si les tables ne contiennent pas les métadonnées requises lors de l'exécution.

Solution

Pour corriger ce problème, vous devez recréer les métadonnées requises. Lors de la création de la base de données d'événements, la commande d'administration de configuration de base de données génère également un script de base de données pouvant être utilisé pour renseigner à nouveau les métadonnées ultérieurement. Le nom de ce script dépend du type de base de données :

Type de base de données	Nom du script
Derby	ins_metadata.derby
DB2	ins_metadata.db2
Informix	ins_metadata.sql
Oracle	ins_metadata.ora
SQL Server	ins_metadata.mssql
DB2 UDB for iSeries	ins_metadata.db2

Par défaut, le script est créé dans le répertoire *racine_profil/dbscripts/nomnoeud_base de données_CEI*. Vous pouvez exécuter ce script à tout moment.

Pour recréer les métadonnées, utilisez le processeur SQL qui convient pour exécuter le script :

- **Windows** **Linux** **UNIX** DB2 : db2
- **Windows** **Linux** **UNIX** Oracle : SQL*Plus
- **Windows** **Linux** **UNIX** Informix : dbaccess
- **Windows** SQL Server : osql
- **i5/OS** DB2 UDB for iSeries : exécutez le script dans QShell

Une fois les métadonnées à nouveau renseignées, redémarrez le serveur.

Erreur lors de l'envoi de l'événement (ServiceUnavailableException)

Mon application source d'événement rencontre une erreur lors de la tentative d'envoi d'un événement au serveur d'événements. Le fichier journal signale une exception ServiceUnavailableException avec le message "Un échec de communication est survenu lors de la tentative d'obtention d'un contexte initial avec l'URL du fournisseur."

Cause

Ce problème indique que l'application source d'événement ne parvient pas à se connecter au serveur. Ceci peut être dû à une ou plusieurs des conditions suivantes :

- Le serveur n'est pas en cours de fonctionnement.
- L'application source d'événement n'est pas configurée pour utiliser l'URL du fournisseur JNDI correcte.

Solution

Pour corriger ce problème, procédez comme suit :

1. Pour vérifier l'état du serveur, accédez au répertoire `racine_profil/bin` et exécutez la commande **serverStatus** :
`serverStatus nom_serveur`
2. Si le serveur n'est pas en cours d'exécution, utilisez la commande **startServer** pour le démarrer :
`startServer nom_serveur`
3. Vérifiez le nom d'hôte et le port RMI (Remote Method Invocation) relatifs au serveur et confirmez que les mêmes valeurs sont spécifiées dans l'URL JNDI configurée pour l'application source d'événement. Si le serveur CEI se trouve sur un autre serveur, le nom JNDI doit être résolu avec cette cible de déploiement distante.

Erreur lors de l'envoi de l'événement (NameNotFoundException)

Mon application source d'événement rencontre une erreur lors de la tentative d'envoi d'un événement au service Evénements. Le fichier journal signale une exception NameNotFoundException avec un message semblable à "Premier composant du nom events/configuration/emitter/Default introuvable."

Cause

Ce problème indique que le service Evénements n'est pas disponible. Ceci peut être dû à une ou plusieurs des conditions suivantes :

- Le service Evénements n'a pas été déployé.
- Le service Evénements est désactivé.

Solution

Pour déployer le service Evénements :

1. Démarrez l'outil `wsadmin`.
2. L'objet `AdminTask` permet d'exécuter la commande d'administration `deployEventService`.

3. Redémarrez le serveur.

Pour activer le service Événements à l'aide de l'outil wsadmin :

1. Démarrez l'outil wsadmin.
2. L'objet AdminTask permet d'exécuter la commande d'administration enableEventService.
3. Redémarrez le serveur.

Pour activer le service Événements à l'aide de la console d'administration :

1. Cliquez sur **Applications** → *serveur* → **Services du conteneur** → **Common Event Infrastructure Service**.
2. Sélectionnez la propriété Activer le service au démarrage du serveur.
3. Cliquez sur **OK** pour sauvegarder les modifications.
4. Redémarrez le serveur.

Erreur lors de l'envoi de l'événement (message CEIEM0025E)

Mon application source d'événement rencontre une erreur lors de la tentative d'envoi d'un événement au serveur d'événements. Le fichier journal indique le message DuplicateGlobalInstanceIdException.

Cause

Ce problème indique que l'émetteur a soumis l'événement, mais le service Événements l'a rejeté car un autre événement existe déjà avec le même ID d'instance global. Chaque événement doit posséder un ID d'instance global unique, spécifié par la propriété globalInstanceId.

Solution

Pour corriger ce problème, procédez de l'une des façons suivantes :

- Assurez-vous que votre application source d'événement génère un ID d'instance global unique pour chaque événement.
- N'indiquez aucune valeur pour la propriété globalInstanceId relative à vos événements soumis. L'émetteur génère alors automatiquement un ID unique pour chaque événement.

Erreur lors de l'envoi de l'événement (message CEIEM0034E)

Ma source d'événement rencontre une erreur lors de la tentative d'envoi d'un événement au service Événements. Le fichier journal signale une exception EmitterException avec le message "La recherche JNDI d'une file d'attente JMS a échoué car le nom JNDI défini dans le profil de l'émetteur n'est pas lié à l'interface JNDI."

Cause

Ce problème indique que la configuration de la transmission JMS utilisée par l'émetteur spécifie une ou plusieurs ressources non définies dans la configuration JMS.

Solution

Pour remédier à cette erreur :

1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Fabriques d'émetteur d'événements > fabrique_émetteur > Paramètres de transmission JMS**. Veillez à consulter la transmission JMS pour la fabrique d'émetteurs utilisée par votre application source d'événement.
2. Vérifiez les valeurs spécifiées pour les propriétés **Nom JNDI de la file d'attente** et **Nom JNDI de la fabrique de connexions de file d'attente**. Assurez-vous que le nom JNDI spécifié existe dans l'espace de nom JNDI et correspond à des objets JMS valides. Au besoin, modifiez ces propriétés ou créez les ressources JMS requises.

Événement non valide (message CEIEM0027E)

Ma source d'événements tente d'envoyer un événement mais l'émetteur ne le soumet pas au service Événements et envoie le message CEIEM0027E au fichier journal ("The emitter did not send the event to the event server because the Common Base Event is not valid").

Cause

Ce message indique qu'une ou plusieurs des propriétés de l'événement contiennent des données qui ne sont pas conformes à la spécification Common Base Event. Il existe de nombreuses raisons pour lesquelles une donnée d'événement peut ne pas être valide, notamment :

- La longueur de l'ID d'instance global doit être d'au moins 32 caractères mais ne doit pas dépasser 64 caractères.
- La gravité doit se situer entre 0 et 70.

Solution

Pour remédier à cette erreur :

1. Consultez le détail du message d'exception dans le fichier journal afin de déterminer la propriété d'événement qui n'est pas valide. Par exemple, le message suivant indique que la longueur de l'ID d'instance global (ABC) n'est pas valide :

```
Exception : org.eclipse.hyades.logging.events.cbe.ValidationException
: IWAT0206E The length of the identifier in the specified Common
Base Event property is outside the valid range of 32 to 64
characters.
Property: CommonBaseEvent.globalInstanceId
Value: ABC
```

2. Corrigez le contenu de l'événement à la source, de manière à ce qu'il soit conforme à la spécification Common Base Event.
3. Soumettez à nouveau l'événement.

Mode de synchronisation non pris en charge (message CEIEM0015E)

Ma source d'événements tente d'envoyer un événement mais l'émetteur ne le soumet pas au service Événements et envoie le message CEIEM0015E au fichier journal ("The emitter does not support the specified synchronization mode").

Cause

Ce problème indique que les paramètres transmis par la source lors de l'envoi de l'événement spécifient un mode de synchronisation non pris en charge par l'émetteur. Ce phénomène peut être dû à une ou plusieurs des conditions suivantes :

- La source d'événements spécifie un mode de synchronisation non valide. Cela est signalé par l'élément `IllegalArgumentException` avec le message "Synchronization mode *mode* is not valid".
- La source d'événements spécifie un mode de synchronisation pour lequel l'émetteur n'est pas configuré. Cela est signalé par l'élément `SynchronizationModeNotSupportedException` avec le message "The emitter does not support the specified synchronization mode: *mode*".

Solution

Si le message d'exception indique que votre source d'événements spécifie un mode de synchronisation non valide (`IllegalArgumentException`), vérifiez l'appel de méthode qui tente d'envoyer l'événement. Assurez-vous que les paramètres de méthode spécifient l'un des modes de synchronisation valides :

- `SynchronizationMode.ASYNCHRONOUS`
- `SynchronizationMode.SYNCHRONOUS`
- `SynchronizationMode.DEFAULT`

Ces constantes sont définies par l'interface `com.ibm.events.emitter.SynchronizationMode`.

Si le message d'exception indique que le mode de synchronisation spécifié n'est pas pris en charge par l'émetteur (`SynchronizationModeNotSupportedException`), vérifiez la configuration de la fabrique d'émetteurs :

1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Fabrique d'émetteurs d'événement > fabrique_émetteurs**. Veillez à consulter la fabrique d'émetteurs utilisée par l'application de source d'événements.
2. Examinez les paramètres de la fabrique d'émetteurs pour connaître les modes de synchronisation pris en charge :
 - Si la propriété relative à la **prise en charge de la transmission de services d'événements** est sélectionnée, le mode synchrone est pris en charge.
 - Si la propriété relative à la **prise en charge de la transmission JMS** est sélectionnée, le mode asynchrone est pris en charge.

Interrogation des modes de transaction : Une source d'événements peut interroger à l'aide d'un programme les modes de transaction pour un émetteur spécifique par le biais de la méthode `isSynchronizationModeSupported()`. Reportez-vous à la documentation Javadoc API pour plus d'informations.

3. Si l'émetteur ne prend pas en charge le mode de synchronisation que vous tentez d'utiliser, vous devez soit changer la configuration de la fabrique d'émetteurs, soit modifier votre source d'événements pour qu'elle utilise un mode de synchronisation pris en charge.

Mode de transaction non pris en charge (message CEIEM0016E)

Ma source d'événements tente d'envoyer un événement mais l'émetteur ne le soumet pas au service Événements et envoie le message CEIEM0016E au fichier journal ("The emitter does not support the specified transaction mode").

Cause

Ce problème indique que les paramètres transmis par la source lors de l'envoi de l'événement spécifient un mode de transaction non pris en charge par l'émetteur. Ce phénomène peut être dû à une ou plusieurs des conditions suivantes :

- La source d'événements spécifie un mode de transaction non valide.
- La source d'événements spécifie un mode de synchronisation non pris en charge par l'environnement de l'émetteur. Les transactions ne sont prises en charge que dans un conteneur J2EE.

Solution

Pour résoudre ce problème, vérifiez l'appel de méthode qui tente d'envoyer l'événement et assurez-vous que les paramètres de méthode spécifient le mode de transaction approprié :

- Si l'émetteur s'exécute dans un conteneur J2EE, veillez à ce que les paramètres de méthode spécifient l'un des modes de transaction valides :
 - TransactionMode.NEW
 - TransactionMode.SAME
 - TransactionMode.DEFAULT

Ces constantes sont définies par l'interface `com.ibm.events.emitter.TransactionMode`.

- Si l'émetteur ne s'exécute pas dans un conteneur J2EE, veillez à ce que les paramètres de méthode spécifient le mode `TransactionMode.DEFAULT`.

Problèmes relatifs à la réception ou à la recherche d'événements

Erreur lors de la recherche d'événements (message CEIDS0060E)

Mon consommateur d'événements rencontre une erreur lors de la tentative de recherche d'événements dans le service Événements, et le message CEIDS0060E s'affiche dans le fichier journal WebSphere.

Cause

Le service Événements utilise des métadonnées stockées dans la base de données d'événements afin de mapper des éléments et des attributs Common Base Event aux colonnes et aux tables de base de données. Ces informations sont lues dans la base de données lorsque l'application tente pour la première fois d'utiliser le service Événements suite au démarrage.

Les tables de métadonnées sont renseignées lors de la création de la base de données d'événements. Cette erreur survient si les tables ne contiennent pas les métadonnées requises lors de l'exécution.

Solution

Pour corriger ce problème, vous devez recréer les métadonnées requises. Lors de la création de la base de données d'événements, la commande d'administration de configuration de base de données génère également un script de base de données pouvant être utilisé pour renseigner à nouveau les métadonnées ultérieurement. Le nom de ce script dépend du type de base de données :

Type de base de données	Nom du script
Derby	ins_metadata.derby
DB2	ins_metadata.db2
Informix	ins_metadata.sql
Oracle	ins_metadata.ora
SQL Server	ins_metadata.mssql
DB2 UDB for iSeries	ins_metadata.db2

Par défaut, le script est créé dans le répertoire *racine_profil/dbscripts/nomnoeud_base de données_CEI*. Vous pouvez exécuter ce script à tout moment.

Pour recréer les métadonnées, utilisez le processeur SQL qui convient pour exécuter le script :

- Windows Linux UNIX DB2 : db2
- Windows Linux UNIX Oracle : SQL*Plus
- Windows Linux UNIX Informix : dbaccess
- Windows SQL Server : osql
- i5/OS DB2 UDB for iSeries : exécutez le script dans QShell

Une fois les métadonnées à nouveau renseignées, redémarrez le serveur.

Événements non stockés dans le magasin de données persistantes

Mon application de source d'événements soumet correctement les événements à l'émetteur mais lorsqu'une source d'événements interroge les événements, ceux-ci ne se trouvent pas dans le magasin de données persistantes.

Cause

Ce problème indique que l'émetteur n'envoie pas les événements au service Événements, ou que le service Événements ne stocke pas les événements dans le magasin de données persistantes. Ce phénomène peut être dû à une ou plusieurs des conditions suivantes :

- Le magasin de données persistantes n'est pas activé pour le service Événements.
- Les événements n'appartiennent pas à un groupe configuré pour la persistance des événements.
- Les événements sont filtrés par l'émetteur.

Solution

Pour vérifier que le magasin de données persistantes est activé pour le service Événements :

1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Service Événements > Services Événements > service_Evénements**.
2. Vérifiez que la case permettant d'**activer le magasin de données d'événements** est cochée.
3. Cliquez sur **OK** pour enregistrer les modifications.

Pour vérifier que le groupe d'événements est configuré pour la persistance des événements :

1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Service Événements > Services Événements > service_Evénements > Groupes d'événements > groupe_événements**.
2. Vérifiez que la case permettant de **conserver les événements dans le magasin de données d'événements** est cochée.
3. Cliquez sur **OK** pour enregistrer les modifications.

Groupes d'événements multiples : Il se peut qu'un événement appartienne à plusieurs groupes d'événements. Si l'un des groupes d'événements applicables est configuré pour la persistance des événements et que le magasin de données est activé, alors l'événement est stocké dans le magasin de données.

Pour vérifier les paramètres de filtrage :

1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Fabriques d'émetteur d'événements > fabrique_émetteur > Filtres d'événements**. (assurez-vous que les paramètres affichés sont bien ceux de la fabrique d'émetteurs que votre application de source d'événements utilise).
2. Vérifiez si la chaîne de configuration du filtre n'exclut pas les événements que vous tentez d'envoyer aux consommateurs. Si c'est le cas, vous pouvez soit modifier la chaîne de configuration du filtre, soit modifier les données d'événements de telle sorte que les événements ne soient pas filtrés.
3. Cliquez sur **OK** pour enregistrer les modifications.

Événements non reçus par les consommateurs (aucun message d'erreur)

Mon application source d'événement soumet correctement les événements à l'émetteur, mais ceux-ci ne sont pas reçus par les consommateurs à l'aide de l'interface JMS.

Cause

Ce phénomène peut être dû à une ou plusieurs des conditions suivantes :

- La distribution d'événement n'est pas activée pour le service Événements.
- Les événements sont filtrés par l'émetteur.
- Les événements sont filtrés par l'assistant de notifications.
- Le consommateur d'événements ne spécifie pas le groupe d'événements correct.
- La connexion JMS n'est pas lancée.

Solution

La solution à ce problème dépend de la cause sous-jacente.

- **Pour vérifier que la distribution d'événement est activée pour le service Événements :**
 1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Service Événements > Services Événements > *service_Evénements***.
 2. Si la propriété **Activer la distribution d'événement** n'est pas sélectionnée, cochez la case correspondante.
 3. Cliquez sur **OK** pour enregistrer les modifications.
- **Pour vérifier les paramètres des filtres d'événements pour l'émetteur :**
 1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Fabriques d'émetteur d'événements > *fabrique_émetteur* > Filtres d'événements**. (assurez-vous que les paramètres affichés sont bien ceux de la fabrique d'émetteurs que votre application de source d'événements utilise).
 2. Vérifiez si la chaîne de configuration du filtre n'exclut pas les événements que vous tentez d'envoyer aux consommateurs. Si c'est le cas, vous pouvez soit modifier la chaîne de configuration du filtre, soit modifier les données d'événements de telle sorte que les événements ne soient pas filtrés.
 3. Cliquez sur **OK** pour enregistrer les modifications.
- **Pour vérifier les paramètres des filtres d'événements pour l'assistant de notifications :**
 1. Vérifiez l'application du consommateur d'événements afin de voir si un sélecteur d'événements est spécifié pour l'assistant de notifications à l'aide de la méthode `NotificationHelper.setEventSelector`.
 2. Si un sélecteur d'événements est spécifié, assurez-vous qu'il n'exclut pas l'événement que vous tentez de recevoir. (Un sélecteur d'événements nul transmet tous les événements.)
- **Pour vérifier le groupe d'événements spécifié par le consommateur d'événements :**
 1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Service Événements > Services Événements > *service_Evénements* > Groupes d'événements**. Le tableau présente la liste de l'ensemble des groupes d'événements définis pour le service Événements.
 2. Sélectionnez le groupe d'événements auquel le consommateur d'événements s'abonne.
 3. Recherchez la propriété **Chaîne de sélecteur d'événements**.
 4. Assurez-vous que le sélecteur d'événements spécifié correspond au contenu de l'événement que vous tentez de recevoir. Si celui-ci ne correspond pas, vous pouvez apporter l'une des modifications suivantes :
 - Modifiez le sélecteur d'événements de sorte que l'événement soit inclus au groupe d'événements.
 - Modifiez les données d'événement de sorte que l'événement corresponde au groupe d'événements.
 - Modifiez votre consommateur d'événements afin de s'abonner à un groupe d'événements différent incluant l'événement.

- **Pour lancer la connexion JMS :**

Dans votre consommateur d'événements, utilisez la méthode `QueueConnection.start()` ou `TopicConnection.start()` avant de tenter de recevoir des événements.

Événements non reçus par les consommateurs (NameNotFoundException)

Mon application source d'événement soumet correctement les événements à l'émetteur, mais ceux-ci ne sont pas publiés aux consommateurs à l'aide de l'interface JMS, et le fichier journal signale une exception `NameNotFoundException`.

Cause

Ce problème indique que la configuration du groupe d'événements spécifie une ou plusieurs ressources JMS n'existant pas.

Solution

Pour remédier à cette erreur :

1. Dans la console d'administration, cliquez sur **Intégration de services > Common Event Infrastructure > Service Événements > Services Événements > service_Événements > Groupes d'événements > groupe_événements**.

Groupes d'événements multiples : Il se peut qu'un événement appartienne à plusieurs groupes d'événements.

2. Vérifiez les valeurs des propriétés **Nom JNDI de la rubrique** et **Nom JNDI de la fabrique de connexions de rubriques**. Vérifiez que les ressources JMS spécifiées existent. Au besoin, utilisez l'interface de configuration de votre fournisseur JMS pour créer les ressources nécessaires.

Un groupe d'événements avec des éléments de données étendus ne comprend aucun événement

J'ai défini un groupe d'événements spécifiant des prédicats d'éléments de données étendus, mais les requêtes adressées au groupe d'événements ne renvoient pas les événements attendus.

Cause

Il se peut que les données d'événement soient au format XML valide mais ne soient pas conformes à la spécification Common Base Event. Cela peut entraîner des résultats inattendus et sans message d'erreur.

Prenons par exemple un événement au contenu suivant :

```
<?xml version="1.0" encoding="ASCII"?>
<!-- Event that will match the XPath expression
CommonBaseEvent[@globalInstanceId] -->
<CommonBaseEvent
  xmlns:xsi="http://www.w3.org/TR/xmlschema-1/"
  xmlns="http://www.ibm.com/AC/commonbaseevent1_0_1"
  version="1.0.1"
  creationTime="2005-10-17T12:00:01Z"
  severity="10"
  priority="60"
  >
  <situation categoryName="RequestSituation">
    <situationType xsi:type="RequestSituation"
      reasoningScope="INTERNAL"
      successDisposition="Succeeded"
      situationQualifier="TEST"
    />
  </situation>
  <sourceComponentId
    component="component"
    subComponent="subcomponent"
    componentIdType="componentIdType"
    location="localhost"
    locationType="Hostname"
    componentType="sourceComponentType"
  />
  <extendedDataElement name="color" type="string">
    <values>red</values>
  </extendedDataElement>
</CommonBaseEvent>
```

Cet élément contient un seul élément de données étendu doté d'un seul élément enfant.

Prenons à présent comme exemple une définition de groupe d'événements configurée avec la chaîne du sélecteur d'événements XPath suivante :

```
CommonBaseEvent[extendedDataElements[@name='color'
and @type='string' and @values='red']]
```

Ce sélecteur d'événements ne correspond pas à l'événement car la définition XML de l'événement contient une faute d'orthographe. Dans les données d'événement, l'élément `extendedDataElements` est mal orthographié et se présente comme suit : `extendedDataElement`. Puisqu'il se trouve au format XML avec une syntaxe correcte, aucune erreur n'est renvoyée ; cet élément est considéré comme n'importe quel élément, et il n'est pas possible de le rechercher.

Solution

Assurez-vous que les données XML relatives aux événements soumis sont conformes à la spécification Common Base Event.

Erreur lors de la recherche d'un groupe d'événements (message CEIES0048E)

Mon application destinataire d'événements rencontre une erreur lors de la tentative de recherche d'événements dans un groupe d'événements. Le fichier journal indique une exception `EventGroupNotDefinedException` et affiche le message CEIES0048E ("Le groupe d'événements n'est pas défini dans la liste de groupes d'événements utilisée par l'instance du serveur d'événements.")

Cause

Ce problème indique que l'application destinataire d'événements a effectué une recherche à l'aide du bean `EventAccess`, mais le destinataire a spécifié un nom de groupe d'événements ne correspondant à aucun groupe d'événements existant.

Solution

Pour remédier à cette erreur :

1. Dans la console d'administration, cliquez sur **Intégration de services** > **Common Event Infrastructure** > **Service Événements** > **Services Événements** > **service_Evénements** > **Groupes d'événements**. Le tableau présente la liste de l'ensemble des groupes d'événements définis pour le service Événements.
2. Assurez-vous que la source d'événements spécifie un nom de groupe d'événements défini dans les paramètres de l'appel de méthode de requête.

Problèmes divers

La demande de modèle de catalogue d'événements échoue sur un système Windows

Je tente d'effectuer une demande de modèle pour les définitions d'événements sur un système Windows à l'aide de la commande `eventcatalog`. Par exemple : `eventcatalog -listdefinitions -name EVENT% -pattern`. Je n'obtiens pas les résultats escomptés.

Cause

Le signe Pourcentage (%) correspond à un caractère réservé dans l'interface de ligne de commande Windows et n'est pas correctement transmis à la commande `eventcatalog`.

Solution

Sur les systèmes Windows, vous devez remédier au problème relatif au signe de pourcentage en saisissant %% :

```
eventcatalog -listdefinitions -name EVENT%% -pattern
```

Chapitre 6. Commandes de l'infrastructure d'événement commune (CEI)

Commandes utilisées par Common Event Infrastructure pour la création et la suppression des bases de données et des services d'événements

Commandes pour la création de sources de données d'événements sur des bases de données spécifiques

- Multiplateforme DB2 — «configEventServiceDB2DB», à la page 96
- DB2 iSeries — «configEventServiceDB2iSeriesDB», à la page 98
- DB2 zOS —
- Derby — «configEventServiceDerbyDB», à la page 104
- Informix — «configEventServiceInformixDB», à la page 106
- Oracle — «configEventServiceOracleDB», à la page 108
- SQL Server — «configEventServiceSQLServerDB», à la page 111

Commandes Common Event Infrastructure génériques utilisées par le serveur pour gérer le service Événements :

- Déployer l'application du service Événements — «deployEventService», à la page 113
- Déployer le bean géré par message de l'application du service Événements — «deployEventServiceMdb», à la page 115
- Créer l'alias d'authentification utilisé par Common Event Infrastructure pour la messagerie — «setEventServiceJmsAuthAlias», à la page 117
- Activation du service Événements au redémarrage du serveur — «enableEventService», à la page 118
- Désactivation du service Événements au redémarrage du serveur — «disableEventService», à la page 120
- Renvoyer l'état du service Événements — «showEventServiceStatus», à la page 121
- Supprimer l'application du service Événements — «removeEventService», à la page 122
- Supprimer le bean géré par message de l'application du service Événements — «removeEventServiceMdb», à la page 123

Commandes de suppression des sources de données d'événements sur des bases de données spécifiques :

- Multiplateforme DB2 — «removeEventServiceDB2DB», à la page 124
- DB2 iSeries — «removeEventServiceDB2iSeriesDB», à la page 126
- DB2 zOS —
- Derby — «removeEventServiceDerbyDB», à la page 129
- Informix — «removeEventServiceInformixDB», à la page 130
- Oracle — «removeEventServiceOracleDB», à la page 132
- SQL Server — «removeEventServiceSQLServerDB», à la page 134

configEventServiceDB2DB

Fonction

La commande `configEventServiceDB2DB` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Utilisez cette commande pour créer la base de données et les sources de données du service Événements pour DB2 sur un serveur ou un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

Paramètres

- `createDB`

Cette commande génère les scripts de la base de données DDL et crée la base de données lorsque ce paramètre est défini sur `true`. Elle génère uniquement les scripts de la base de données DDL lorsque ce paramètre est défini sur `false`. Pour créer la base de données, le serveur actuel doit avoir été configuré en vue d'exécuter des commandes de base de données. La valeur par défaut est `false` s'il n'est pas défini.

- `overrideDataSource`

Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le paramètre spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans *profile_root/bin*. En l'absence de spécification de ce paramètre, le répertoire de sortie du script de base de données par défaut est *profile_root/databases/event/node/server/dbscripts/dbtype*.

- `nodeName`

Nom du noeud contenant le serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `serverName`

Nom du serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est spécifié sans le paramètre `nodeName`, la commande utilisera le nom du noeud du profil WebSphere. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `clusterName`

Nom du cluster sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas être spécifiés. Ne spécifiez pas ce paramètre si les paramètres `serverName` et `nodeName` sont définis.

- `jdbcClassPath`

Chemin d'accès du pilote JDBC. Indiquez uniquement le chemin d'accès au fichier du pilote de périphérique ; n'incluez pas le nom du fichier dans le chemin. Ce paramètre est obligatoire.

- `dbNodeName`

Nom du noeud DB2 (il doit être composé de 8 caractères maximum). Ce noeud doit déjà être catalogué et configuré pour communiquer avec le serveur DB2. Ce paramètre doit être défini si le serveur actuel est configuré en tant que client DB2 et que le paramètre `createDB` est défini sur `true`.

- **dbHostName**
Nom d'hôte du serveur sur lequel le serveur de base de données est installé. Ce paramètre est obligatoire.
- **dbPort**
Port d'instance DB2. La valeur par défaut est 50 000 s'il n'est pas défini.
- **dbName**
Nom de la base de données à créer. La valeur par défaut est event s'il n'est pas défini.
- **dbUser**
ID utilisateur DB2 ayant les privilèges adéquats pour créer et supprimer les bases de données. La valeur par défaut est db2inst1 s'il n'est pas défini.
- **dbPassword**
Mot de passe DB2. Ce paramètre est obligatoire.
- **outputScriptDir**
Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le paramètre spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans *profile_root*/bin. En l'absence de spécification de ce paramètre, le répertoire de sortie de scripts de base de données par défaut est *profile_root*/databases/event/node/server/dbscripts/db2.

Exemple

Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :


```
$AdminTask configEventServiceDB2DB {-createdB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servename
  -jdbcClassPath c:\sqllib\java
  -dbUser db2inst1
  -dbPassword dbpassword
  -dbHostName host_name
  -dbPort 50000 }
```
- A l'aide d'une chaîne Jython :


```
AdminTask.configEventServiceDB2DB(['-createdB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servename
  -jdbcClassPath c:\sqllib\java
  -dbUser db2inst1
  -dbPassword dbpassword
  -dbHostName host_name
  -dbPort 50000 '])
```
- A l'aide d'une liste Jython :


```
AdminTask.configEventServiceDB2DB(['-createdB', 'true',
  '-overrideDataSource', 'true', '-nodeName', 'nodename',
  '-serverName', 'servename',
  '-jdbcClassPath', 'c:\sqllib\java',
  '-dbUser', 'db2inst1',
  '-dbPassword', 'dbpassword ',
  '-dbHostName', 'host_name',
  '-dbPort', '50000 '])
```

Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :
`$AdminTask configEventServiceDB2DB -interactive`
- A l'aide d'une chaîne Jython :
`AdminTask.configEventServiceDB2DB(['-interactive'])`
- A l'aide d'une liste Jython :
`AdminTask.configEventServiceDB2DB(['-interactive'])`

configEventServiceDB2iSeriesDB

Fonction

La commande `configEventServiceDB2iSeriesDB` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. La commande génère les scripts de base de données DDL afin de l'utiliser sur les serveurs de bases de données éloignées, de créer la base de données du service Événements DB2 iSeries sur un serveur local et de créer des sources de données sur un serveur ou un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

Paramètres

- `createDB`

La commande génère les scripts de la base de données DDL et crée la base de données lorsque ce paramètre est défini sur `true`. Elle génère uniquement les scripts de la base de données DDL lorsque ce paramètre est défini sur `false`. Pour créer la base de données, le serveur actuel doit avoir été configuré en vue d'exécuter des commandes de base de données. La valeur par défaut est `false` si aucune valeur n'est spécifiée.

- `overrideDataSource`

Lorsque ce paramètre a la valeur `true`, la commande supprime la source de données du service d'événements existante à la portée indiquée avant d'en créer une nouvelle. Quand ce paramètre a la valeur `false`, cette commande ne crée pas de source de données de service d'événements à la portée spécifiée si une autre source de ce type est trouvée au niveau de la même portée. La valeur par défaut est `false` si aucune valeur n'est spécifiée.

- `nodeName`

Nom du noeud contenant le serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `serverName`

Nom du serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est spécifié sans le paramètre `nodeName`, la commande utilisera le nom du noeud du profil WebSphere. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `clusterName`

Nom du cluster sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas être spécifiés. Ne spécifiez pas ce paramètre si les paramètres `serverName` et `nodeName` sont définis.

- **toolboxJdbcClassPath**

Chemin d'accès au pilote JDBC DB2 d'IBM Toolbox for Java . Indiquez uniquement le chemin d'accès au fichier de pilote ; ne spécifiez pas le nom du fichier. Vous devez indiquer ce paramètre ou le paramètre **jdbcClassPath**.

Remarque : Vous devez spécifier le chemin **toolboxJdbcClassPath** si vous créez la base de données sur un serveur iSeries.

- **nativeJdbcClassPath**

Chemin d'accès au pilote JDBC natif DB2 for iSeries. Indiquez uniquement le chemin d'accès du fichier du pilote de périphérique ; n'incluez pas le nom du fichier dans le chemin. Vous devez indiquer ce paramètre ou le paramètre **toolboxJdbcClassPath**.

Remarque : Vous devez spécifier le chemin d'accès **nativeJdbcClassPath** si vous utilisez un serveur iSeries pour créer la base de données sur un serveur qui n'est pas de type i-Series. Spécifiez le pilote DB2 Universal si vous créez la base de données sur DB2 Universal Database pour les multiplateformes ou DB2 Universal Database pour z/OS. Spécifiez un pilote de type 4 si vous créez la base de données sur Informix, Oracle ou Microsoft SQL Server.

- **dbHostName**

Nom d'hôte du serveur sur lequel le serveur de base de données DB2 for iSeries est installé. Ce paramètre est obligatoire si vous utilisez le pilote JDBC IBM Toolbox for Java DB2.

- **dbName**

Nom de la base de données DB2 for iSeries. La valeur par défaut est *LOCAL s'il n'est pas spécifié.

- **collection**

Collection SQL de la bibliothèque DB2 for iSeries. La longueur maximale du nom de la collection est de 10 caractères. La valeur par défaut est une chaîne vide s'il n'est pas spécifié.

- **dbUser**

ID utilisateur DB2 ayant les privilèges adéquats pour créer et supprimer les bases de données. Ce paramètre est obligatoire.

- **dbPassword**

Mot de passe de l'ID utilisateur de la base de données. Ce paramètre est obligatoire.

- **outputScriptDir**

Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le répertoire spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans *profile_root/bin*. En l'absence de spécification de ce paramètre, le répertoire de sortie de script de la base de données par défaut est *profile_root/databases/event/node/server/dbscripts/db2iseries*.

Exemple

Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask configEventServiceDB2iSeriesDB {createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -dbUser db2user
  -dbPassword dbpassword
  -nativeJdbcClassPath /myDB2ClassPath
  -collection event}
```

- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceDB2iSeriesDB(['-createDB true
-overrideDataSource true -nodeName nodename -serverName servername
-nativeJdbcClassPath /myDB2ClassPath -collection event'])
```

- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceDB2iSeriesDB(['-createDB', 'true',
'-overrideDataSource', 'true', '-nodeName', 'nodename',
'-serverName', 'servername', '-nativeJdbcClassPath',
'/myDB2ClassPath', '-collection', 'event'])
```

Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask configEventServiceDB2iSeriesDB -interactive
```

- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceDB2iSeriesDB(['-interactive'])
```

- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceDB2iSeriesDB(['-interactive'])
```

configEventServiceDB2ZOSDB

Fonction

La commande `configEventServiceDB2ZOSDB` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Utilisez cette commande pour créer la base de données et les sources de données du service Événements pour DB2 z/OS sur un serveur ou un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

Paramètres

- `createDB`

La commande génère les scripts de la base de données DDL et crée la base de données lorsque ce paramètre est défini sur `true`. Elle génère uniquement les scripts de la base de données DDL lorsque ce paramètre est défini sur `false`. Pour créer la base de données, le serveur actuel doit avoir été configuré en vue d'exécuter des commandes de base de données. La valeur par défaut est `false` en l'absence de spécification.

- `overrideDataSource`

Lorsque ce paramètre a la valeur `true`, la commande supprime la source de données du service d'événements existante à la portée indiquée avant d'en créer une nouvelle. Quand ce paramètre a la valeur `false`, cette commande ne crée pas de source de données de service d'événements à la portée spécifiée si une autre source de ce type est trouvée au niveau de la même portée. La valeur par défaut est `false` en l'absence de spécification.

- `nodeName`

Nom du noeud contenant le serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `serverName`

Nom du serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est spécifié sans le paramètre `nodeName`, la commande utilisera le nom du noeud du profil WebSphere actuel. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `clusterName`

Nom du cluster sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas être spécifiés. Ne spécifiez pas ce paramètre si les paramètres `serverName` et `nodeName` sont définis.

- `jdbcClassPath`

Chemin d'accès du pilote JDBC. Indiquez uniquement le chemin d'accès au fichier du pilote de périphérique ; n'incluez pas le nom du fichier dans le chemin. Ce paramètre est obligatoire.

- `dbHostName`

Nom d'hôte du serveur sur lequel la base de données est installée. Ce paramètre est obligatoire.

- `dbPort`

Port d'instance DB2 pour z/OS. La valeur par défaut est 5027 s'il n'est pas spécifié.

- **dbName**
Nom de base de données DB2. Sur la machine cliente DB2, il s'agit du nom de la base de données cataloguée. Sur le serveur z/OS natif, il s'agit du nom du sous-système de base de données. La valeur par défaut est event, si aucune valeur n'est spécifiée.
- **dbDiskSizeInMB**
Indiquez la taille du disque en Mo pour la base de données de service d'événements. Cette valeur doit au moins être égale à 10 Mo. La valeur par défaut est 100 Mo s'il n'est pas défini.
- **dbUser**
ID utilisateur DB2 ayant les privilèges adéquats pour créer et supprimer les bases de données. Ce paramètre est obligatoire.
- **dbPassword**
Mot de passe de l'ID utilisateur de la base de données. Ce paramètre est obligatoire.
- **storageGroup**
Groupe de stockage de la base de données des événements et de la base de données du catalogue d'événements. Le groupe de stockage doit avoir été créé et être actif.
- **bufferPool4K**
Nom du pool de mémoire tampon de 4 Ko. Ce pool de mémoire tampon doit être actif pour que les scripts DDL de la base de données puissent être exécutés.
- **bufferPool8K**
Nom du pool de mémoire tampon de 8 Ko. Ce pool de mémoire tampon doit être actif pour que les scripts DDL de la base de données puissent être exécutés.
- **bufferPool16K**
Nom du pool de mémoire tampon de 16 ko. Ce pool de mémoire tampon doit être actif pour que les scripts DDL de la base de données puissent être exécutés.
- **outputScriptDir**
Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le paramètre spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans *profile_root*/bin. En l'absence de spécification de ce paramètre, le répertoire de sortie de script de base de données par défaut est *profile_root*/databases/event/*node/server*/dbscripts/db2zos.

Exemple

Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask configEventServiceDB2ZOSDB {-createDB true
-overrideDataSource true -nodeName nodename
-serverName servername -jdbcClassPath c:\sqllib\java
-dbUser db2user -dbPassword dbpassword
-dbHostName nom_hôte -dbPort 5027
-storageGroup sysdeflt -bufferPool4K BP9
-bufferPool18K BP8K9 -bufferPool16K BP16K9}
```
- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceDB2ZOSDB(['-createDB true -overrideDataSource
true -nodeName nodename -serverName servername -jdbcClassPath c:\sqllib\java
-dbUser db2user -dbPassword dbpassword -dbHostName nom_hôte -dbPort 5027
-storageGroup sysdeflt -bufferPool4K BP9 -bufferPool18K BP8K9
-bufferPool16K BP16K9'])
```
- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceDB2ZOSDB(['-createDB', 'true',
'-overrideDataSource', 'true', '-nodeName', 'nodename',
'-serverName', 'servername', '-jdbcClassPath', 'c:\sqllib\java',
'-dbUser', 'db2user', '-dbPassword', 'dbpassword', '-dbHostName',
'nom_hôte', '-dbPort', '5027', '-storageGroup', 'sysdeflt',
'-bufferPool4K', 'BP9', '-bufferPool18K', 'BP8K9',
'-bufferPool16K', 'BP16K9'])
```

Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask configEventServiceDB2ZOSDB -interactive
```
- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceDB2ZOSDB(['-interactive'])
```
- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceDB2ZOSDB ['-interactive'])
```

configEventServiceDerbyDB

Fonction

La commande `configEventServiceDerbyDB` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Utilisez cette commande pour créer la base de données et les sources de données du service Événements pour Derby sur un serveur ou un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

Paramètres

- `createDB`

Cette commande génère les scripts de la base de données DDL et crée la base de données lorsque ce paramètre est défini sur `true`. Elle génère uniquement les scripts de la base de données DDL lorsque ce paramètre est défini sur `false`. Pour créer la base de données, le serveur actuel doit avoir été configuré en vue d'exécuter des commandes de base de données. La valeur par défaut est `false` en l'absence de spécification.

- `overrideDataSource`

Lorsque ce paramètre a la valeur `true`, la commande supprime la source de données du service d'événements existante à la portée indiquée avant d'en créer une nouvelle. Quand ce paramètre a la valeur `false`, cette commande ne crée pas de source de données de service d'événements à la portée spécifiée si une autre source de ce type est trouvée au niveau de la même portée. La valeur par défaut est `false` en l'absence de spécification.

- `nodeName`

Nom du noeud contenant le serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `serverName`

Nom du serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est spécifié sans le paramètre `nodeName`, la commande utilisera le nom du noeud du profil WebSphere actuel. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `clusterName`

Nom du cluster sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas être spécifiés. Ne spécifiez pas ce paramètre si les paramètres `serverName` et `nodeName` sont définis.

- `dbHostName`

Nom d'hôte de la base de données Derby réseau. Pour créer la source de données réseau Derby, indiquez ce paramètre et le paramètre `dbPort`. Pour créer la source de données locale Derby, n'indiquez pas ce paramètre ni le paramètre `dbPort`.

- `dbPort`

Numéro de port de la base de données Derby réseau. Pour créer la source de données réseau Derby, indiquez ce paramètre et le paramètre `dbHostName`. Pour créer la source de données locale Derby, n'indiquez pas ce paramètre ni le paramètre `dbHostName`.

- **dbName**
Nom de la base de données à créer. La valeur par défaut est `event` s'il n'est pas défini.
- **dbUser**
ID utilisateur auquel fait appel la source de données pour l'authentification de la base de données Derby. Ce paramètre est facultatif lorsque la sécurité de domaine WebSphere est désactivée. Si vous spécifiez ce paramètre, vous devez aussi indiquer le paramètre **dbPassword**. Ce paramètre est obligatoire lorsque la sécurité de domaine WebSphere est activée.
- **dbPassword**
Mot de passe auquel fait appel la source de données pour l'authentification de la base de données Derby. Ce paramètre est facultatif lorsque la sécurité de domaine WebSphere est désactivée. Si vous spécifiez ce paramètre, vous devez aussi indiquer le paramètre **dbUser**. Ce paramètre est obligatoire lorsque la sécurité de domaine WebSphere est activée.
- **outputScriptDir**
Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le répertoire spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans `profile_root/bin`. En l'absence de spécification de ce paramètre, le répertoire de sortie de script de la base de données par défaut est `profile_root/databases/event/node/server/dbscripts/derby`

Exemple

Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask configEventServiceDerbyDB {-createDB true -overrideDataSource true -nodeName nodename -serverName servername}
```
- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceDerbyDB( ['-createDB true -overrideDataSource true -nodeName nodename -serverName servername'] )
```
- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceDerbyDB(['-createDB', 'true', '-overrideDataSource', 'true', '-nodeName', 'nodename', '-serverName', 'servername'])
```

Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask configEventServiceDerbyDB -interactive
```
- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceDerbyDB(['-interactive'] )
```
- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceDerbyDB(['-interactive'])
```

configEventServiceInformixDB

Fonction

La commande `configEventServiceInformixDB` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Utilisez cette commande pour supprimer la base de données et les sources de données du service Événements pour Informix sur un serveur ou un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

Paramètres

- `createDB`

Cette commande génère les scripts de la base de données DDL et crée la base de données lorsque ce paramètre est défini sur `true`. Elle génère uniquement les scripts de la base de données DDL lorsque ce paramètre est défini sur `false`. Pour créer la base de données, le serveur actuel doit avoir été configuré en vue d'exécuter des commandes de base de données. La valeur par défaut est `false` en l'absence de spécification.

- `overrideDataSource`

Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le paramètre spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans *profile_root/bin*. En l'absence de spécification de ce paramètre, le répertoire de sortie du script de base de données par défaut est *profile_root/databases/event/node/server/dbscripts/dbtype*.

- `nodeName`

Nom du noeud contenant le serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `serverName`

Nom du serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est spécifié sans le paramètre `nodeName`, la commande utilisera le nom du noeud du profil WebSphere. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `clusterName`

Nom du cluster sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas être spécifiés. Ne spécifiez pas ce paramètre si les paramètres `serverName` et `nodeName` sont définis.

- `jdbcClassPath`

Chemin d'accès du pilote JDBC. Indiquez uniquement le chemin d'accès au fichier du pilote de périphérique ; n'incluez pas le nom du fichier dans le chemin. Ce paramètre est obligatoire.

- `dbInformixDir`

Répertoire dans lequel est installée la base de données Informix. Ce paramètre doit être spécifié quand le paramètre `createDB` est défini sur `true`. Ce paramètre est obligatoire.

- **dbHostName**
Nom d'hôte du serveur sur lequel la base de données est installée. Ce paramètre est obligatoire.
- **dbServerName**
Nom du serveur Informix (par exemple, `nom_serveur_01`). Ce paramètre est obligatoire.
- **dbPort**
Port de l'instance Informix. La valeur par défaut est 1526 s'il n'est pas défini.
- **dbName**
Nom de la base de données à créer. La valeur par défaut est `event` s'il n'est pas défini.
- **dbUser**
ID utilisateur du schéma de base de données Informix qui détiendra les tables de base de données d'événements. La source de données WebSphere utilise cet ID utilisateur pour authentifier la connexion à la base de données Informix. Ce paramètre est obligatoire.
- **dbPassword**
Mot de passe de l'ID utilisateur de schéma détenant les tables Informix du service Événements. La source de données WebSphere utilise ce mot de passe pour authentifier la connexion à la base de données Informix. Ce paramètre est obligatoire.
- **ceiInstancePrefix**
Cette commande utilise le nom de l'instance du service Événements pour regrouper les fichiers de la base de données sous des noms uniques au sein d'un même groupe. La valeur par défaut est `cei inst1` s'il n'est pas défini.
- **outputScriptDir**
Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le paramètre spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans `profile_root/bin`. En l'absence de spécification de ce paramètre, le répertoire de sortie de base de données par défaut est `profile_root/databases/event/node/server/dbscripts/informix`.

Exemple

Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :


```
$AdminTask configEventServiceInformixDB {-createdb true -overrideDataSource true
-nodeName nodename -serverName servername -jdbcClassPath
"c:\program files\ibm\informix\jdbc\lib" -dbInformixDir
"c:\program files\ibm\informix" -dbUser informix -dbPassword dbpassword
-dbHostName nom_hôte -dbPort 1526 -dbServerName serveur_01 }
```
- A l'aide d'une chaîne Jython :


```
AdminTask.configEventServiceInformixDB(['-createdb true
-overrideDataSource true
-nodeName nodename
-serverName servername
-jdbcClassPath "c:\program files\ibm\informix\jdbc\lib"
-dbInformixDir "c:\program files\ibm\informix"
-dbUser informix
```

```
-dbPassword dbpassword
-dbHostName host_name
-dbPort 1526
-dbServerName ol_server]')
```

- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceInformixDB(['-createDB', 'true',
'-overrideDataSource', 'true', '-nodeName', 'nodename',
'-serverName', 'servername', '-jdbcClassPath',
'c:\program files\ibm\informix\jdbc\lib', '-dbInformixDir',
'c:\program files\ibm\informix', '-dbUser', 'informix ',
'-dbPassword', 'dbpassword', '-dbHostName', 'host_name',
'-dbPort', '1526', '-dbServerName', 'ol_server'])
```

Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask configEventServiceInformixDB -interactive
```

- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceInformixDB(['-interactive'])
```

- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceInformixDB(['-interactive'])
```

configEventServiceOracleDB

Fonction

La commande `configEventServiceOracleDB` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Utilisez cette commande pour créer les tables et les sources de données du service Événements pour Oracle sur un serveur ou un cluster. Cette commande ne crée pas la base de données ; le SID Oracle doit déjà exister. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

Paramètres

- createDB

Cette commande génère les scripts de la base de données DDL et crée la base de données lorsque ce paramètre est défini sur `true`. Elle génère uniquement les scripts de la base de données DDL lorsque ce paramètre est défini sur `false`. Pour créer la base de données, le serveur actuel doit avoir été configuré en vue d'exécuter des commandes de base de données. La valeur par défaut est `false` s'il n'est pas défini.

- overrideDataSource

Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le répertoire spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans `profile_root/bin`. En l'absence de spécification de ce paramètre, le répertoire de sortie du script de base de données par défaut est `profile_root/databases/event/node/server/dbscripts/dbtype`.

- nodeName

Nom du noeud contenant le serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- **serverName**
Nom du serveur sur lequel la source de données du service Evénements doit être créée. Si ce paramètre est spécifié sans le paramètre **nodeName**, la commande utilisera le nom du noeud du profil WebSphere. N'indiquez pas ce paramètre si le paramètre **clusterName** est spécifié.
- **clusterName**
Nom du cluster sur lequel la source de données du service Evénements doit être créée. Si ce paramètre est défini, les paramètres **serverName** et **nodeName** ne doivent pas être spécifiés. Ne spécifiez pas ce paramètre si les paramètres **serverName** et **nodeName** sont définis.
- **jdbcClassPath**
Chemin d'accès du pilote JDBC. Indiquez uniquement le chemin d'accès au fichier du pilote de périphérique ; n'incluez pas le nom du fichier dans le chemin. Ce paramètre est obligatoire.
- **oracleHome**
Répertoire ORACLE_HOME. Ce paramètre doit être défini quand le paramètre **createDB** est défini sur true.
- **dbHostName**
Nom d'hôte du serveur sur lequel la base de données Oracle est installée. La valeur par défaut est localhost s'il n'est pas spécifié.
- **dbPort**
Port de l'instance Oracle. La valeur par défaut est 1521 s'il n'est pas défini.
- **dbName**
Identificateur système (SID) Oracle. Le SID doit déjà exister et doit être disponible pour la commande du service d'événements afin de créer les tables et d'y placer les données. La valeur par défaut est orcl s'il n'est pas défini.
- **dbUser**
ID utilisateur du schéma Oracle qui détiendra les tables Oracle de service d'événements. L'ID utilisateur sera créé durant la création de la base de données ; la source de données WebSphere utilise cet ID utilisateur pour authentifier la connexion à la base de données Oracle. La valeur par défaut est ceiuser s'il n'est pas spécifié.
- **dbPassword**
Mot de passe de l'ID utilisateur du schéma. Ce mot de passe sera créé durant la création de la base de données ; la source de données WebSphere utilise ce mot de passe pour authentifier la connexion à la base de données Oracle. Ce paramètre est obligatoire.
- **sysUser**
ID utilisateur système Oracle. Il doit s'agir d'un utilisateur ayant les privilèges SYSDBA. La valeur par défaut est sys s'il n'est pas défini.
- **sysPassword**
Mot de passe pour l'utilisateur spécifié par le paramètre **sysUser**. La valeur par défaut est une chaîne vide s'il n'est pas spécifié.
- **ceiInstancePrefix**
Cette commande utilise le nom de l'instance du service Evénements pour regrouper les fichiers de la base de données sous des noms uniques au sein d'un même répertoire. La valeur par défaut est ceiinst1 s'il n'est pas défini.
- **outputScriptDir**
Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du

service d'événements dans le répertoire indiqué. Si le répertoire spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans *profile_root/bin*. En l'absence de spécification de ce paramètre, le répertoire de sortie de base de données par défaut est *profile_root/databases/event/node/server/dbscripts/oracle*.

Exemple

Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask configEventServiceOracleDB {-createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -jdbcClassPath c:\oracle\ora92\jdbc\lib
  -oracleHome c:\oracle\ora92
  -dbUser ceiuser
  -dbPassword ceipassword
  -dbHostName host_name
  -dbPort 1521
  -sysUser sys
  -sysPassword syspassword}
```

- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceOracleDB( ['-createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -jdbcClassPath c:\oracle\ora92\jdbc\lib
  -oracleHome c:\oracle\ora92
  -dbUser ceiuser
  -dbPassword ceipassword
  -dbHostName host_name
  -dbPort 1521
  -sysUser sys
  -sysPassword syspassword] )
```

- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceOracleDB(['-createDB', 'true',
  '-overrideDataSource', 'true', '-nodeName', 'nodename',
  '-serverName', 'servername', '-jdbcClassPath',
  'c:\oracle\ora92\jdbc\lib', '-oracleHome',
  'c:\oracle\ora92', '-dbUser', 'ceiuser',
  '-dbPassword', 'ceipassword', '-dbHostName',
  'host_name', '-dbPort', '1521', '-sysUser', 'sys',
  '-sysPassword', 'syspassword'])
```

Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask configEventServiceOracleDB -interactive
```

- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceOracleDB(['-interactive'] )
```

- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceOracleDB(['-interactive'])
```

configEventServiceSQLServerDB

Fonction

La commande `configEventServiceSQLServerDB` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Utilisez cette commande pour créer la base de données et les sources de données du service Événements pour SQL Server sur un serveur ou un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

Paramètres

- `createDB`

Cette commande génère les scripts de la base de données DDL et crée la base de données lorsque ce paramètre est défini sur `true`. Elle génère uniquement les scripts de la base de données DDL lorsque ce paramètre est défini sur `false`. Pour créer la base de données, le serveur actuel doit avoir été configuré en vue d'exécuter des commandes de base de données. La valeur par défaut est `false` s'il n'est pas défini.

- `overrideDataSource`

Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le répertoire spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans *profile_root/bin*. En l'absence de spécification de ce paramètre, le répertoire de sortie du script de base de données par défaut est *profile_root/databases/event/node/server/dbscripts/dbtype*.

- `nodeName`

Nom du noeud contenant le serveur sur lequel la source de données du service d'événements doit être créée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `serverName`

Nom du serveur sur lequel la source de données du service Événements doit être créée. Si ce paramètre est spécifié sans le paramètre `nodeName`, la commande utilisera le nom du noeud du profil WebSphere. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

- `clusterName`

Nom du cluster sur lequel la source de données du service Événements doit être créée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas être définis. Ne spécifiez pas ce paramètre si les paramètres `serverName` et `nodeName` sont définis.

- `dbServerName`

Nom du serveur de la base de données SQL Server. Ce paramètre doit être défini quand le paramètre `createDB` est défini sur `true`.

- `dbHostName`

Nom d'hôte du serveur sur lequel la base de données SQL Server s'exécute.

- `dbPort`

Port SQL Server. La valeur par défaut est 1433 s'il n'est pas défini.

- **dbName**
Nom de la base de données à créer. La valeur par défaut est event s'il n'est pas défini.
- **dbUser**
ID utilisateur SQL Server qui détiendra les tables du service d'événements. La valeur par défaut est ceiuser s'il n'est pas spécifié.
- **dbPassword**
Mot de passe associé à l'ID utilisateur SQL Server spécifié par le paramètre **dbUser**. Ce paramètre est obligatoire.
- **saUser**
ID utilisateur ayant les privilèges adéquats pour créer et supprimer les bases de données et les utilisateurs. Ce paramètre doit être spécifié quand le paramètre **createDB** est défini sur true. La valeur par défaut est sa s'il n'est pas défini.
- **saPassword**
Mot de passe sa. Vous ne devez pas spécifier ce paramètre si l'ID utilisateur sa n'a pas de mot de passe.
- **ceiInstancePrefix**
Cette commande utilise le nom de l'instance du service Evénements pour regrouper les fichiers de la base de données sous des noms uniques au sein d'un même répertoire. La valeur par défaut est cei inst1 s'il n'est pas défini.
- **outputScriptDir**
Répertoire facultatif de sortie des scripts de base de données. Lorsque ce paramètre est défini, la commande génère les scripts de base de données du service d'événements dans le répertoire indiqué. Si le répertoire spécifié ne contient pas de chemin complet, la commande crée le répertoire spécifié dans *profile_root/bin*. En l'absence de spécification de ce paramètre, le répertoire de sortie de base de données par défaut est *profile_root/databases/event/node/server/dbscripts/sqlserver*.

Exemple

Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :


```
$AdminTask configEventServiceSQLServerDB {-createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -dbUser ceiuser
  -dbPassword ceipassword
  -dbServerName sqlservername
  -dbHostName host_name
  -dbPort 1433
  -saUser sa
  -saPassword sapassword}
```
- A l'aide d'une chaîne Jython :


```
AdminTask.configEventServiceSQLServerDB(['-createDB true
  -overrideDataSource true
  -nodeName nodename
  -serverName servername
  -dbUser ceiuser
  -dbPassword ceipassword
  -dbServerName sqlservername
```


- ```
-dbHostName host_name
-dbPort 1433
-saUser sa
-saPassword sapassword]')
```
- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceSQLServerDB(['-createDB', 'true',
'-overrideDataSource', 'true', '-nodeName', 'nodename',
'-serverName', 'servername', '-dbUser', 'ceiuser',
'-dbPassword', 'ceipassword', '-dbServerName',
'sqlservername', '-dbHostName', 'host_name',
'-dbPort', '1433', '-saUser', 'sa',
'-saPassword', 'sapassword'])
```

#### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask configEventServiceSQLServerDB -interactive
```
- A l'aide d'une chaîne Jython :

```
AdminTask.configEventServiceSQLServerDB(['-interactive'])
```
- A l'aide d'une liste Jython :

```
AdminTask.configEventServiceSQLServerDB(['-interactive'])
```

---

## deployEventService

### Fonction

La commande `deployEventService` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Utilisez cette commande pour déployer et configurer le service Événements sur un serveur ou un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à `WebSphere Application Server Network Deployment`, version 6.1.

### Paramètres

#### - nodeName

Nom du noeud sur lequel déployer le service Événements. Si ce paramètre est indiqué, le paramètre `serverName` doit l'être aussi. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - serverName

Nom du serveur sur lequel déployer le service Événements. Vous devez indiquer ce paramètre si le paramètre **nodeName** est spécifié. N'indiquez pas ce paramètre si le paramètre **clusterName** est spécifié.

#### - clusterName

Nom du cluster sur lequel déployer le service Événements. Vous ne devez pas spécifier ce paramètre si le paramètre **nodeName** ou **serverName** est spécifié.

#### - enable

Définissez ce paramètre sur `true` pour que le service Événements démarre au prochain redémarrage du serveur. La valeur par défaut est `true`.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask deployEventService {-nodeName nodename -serverName servername}
$AdminTask deployEventService {-clusterName clustername -enable false}
```

- A l'aide d'une chaîne Jython :  

```
AdminTask.deployEventService('[-nodeName nodename
-serverName servename]')
```

```
AdminTask.deployEventService('[-clusterName clustername -enable false]')
```
- A l'aide d'une liste Jython :  

```
AdminTask.deployEventService(['-nodeName', 'nodename',
'-serverName', '-servename'])
```

```
AdminTask.deployEventService(['-clusterName', 'clustername', '-enable', 'false'])
```

**Exemple d'utilisation du mode interactif :**

- A l'aide de Jacl :  

```
$AdminTask deployEventService {-interactive}
```
- A l'aide d'une chaîne Jython :  

```
AdminTask.deployEventService('[-interactive]')
```
- A l'aide d'une liste Jython :  

```
AdminTask.deployEventService(['-interactive'])
```

---

## deployEventServiceMdb

### Fonction

La commande `deployEventServiceMdb` est une commande d'administration Common Event Infrastructure qui est applicable à l'objet `AdminTask`. Utilisez cette commande pour déployer le bean géré par message du service Événements sur un serveur ou un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

### Paramètres

- **nodeName**

Nom du noeud sur lequel déployer le bean géré par message du service Événements. Si ce paramètre est défini, le paramètre **serverName** doit l'être également. N'indiquez pas ce paramètre si le paramètre **clusterName** est spécifié.

- **serverName**

Nom du serveur sur lequel déployer le bean géré par message du service Événements. Vous devez indiquer ce paramètre si le paramètre **nodeName** est spécifié. N'indiquez pas ce paramètre si le paramètre **clusterName** est spécifié.

- **clusterName**

Nom du cluster sur lequel déployer le bean géré par message du service Événements. N'indiquez pas ce paramètre si les paramètres **nodeName** et **serverName** sont spécifiés.

- **applicationName**

Nom de l'application du bean géré par message du service d'événements à déployer sur un serveur ou cluster.

- **listenerPort**

Nom du port d'écoute sur lequel le bean géré par message du service Événements doit publier ces derniers. Le port d'écoute doit déjà avoir été créé. N'indiquez pas ce paramètre si le paramètre **activationSpec** est spécifié.

- **activationSpec**

Nom JNDI de la spécification d'activation sur laquelle le bean géré par message du service Événements doit publier ces derniers. La spécification d'activation doit déjà avoir été créée. N'indiquez pas ce paramètre si le paramètre **listenerPort** est spécifié.

- **qcfJndiName**

Nom JNDI de l'objet de fabrication de connexion de file d'attente JMS à utiliser par le bean géré par message du service Événements. Indiquez ce paramètre si le paramètre **activationSpec** est spécifié.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask deployEventServiceMdb {-applicationName appname -nodeName nodename
-serverName servername -listenerPort lpname}$AdminTask deployEventServiceMdb
{-applicationName appname -clusterName clustername -activationSpec asjndiname
-qcfJndiName qcfjndiname}
```

- A l'aide d'une chaîne Jython :

```
AdminTask.deployEventServiceMdb(['-applicationName appname
-nodeName nodename -serverName servername -listenerPort
lpname'])AdminTask.deployEventServiceMdb(['-applicationName
appname -clusterName clustername -activationSpec asjndiname
-qcfJndiName qcfjndiname'])
```

- A l'aide d'une liste Jython :

```
AdminTask.deployEventServiceMdb(['-applicationName', 'appname',
'-nodeName', 'nodename', '-serverName', '-servername',
'-listenerPort', 'lpname'])
AdminTask.deployEventServiceMdb(['-applicationName', 'appname',
'-clusterName', 'clustername', '-activationSpec', 'asjndiname',
'-qcfJndiName', 'qcfjndiname'])
```

#### **Exemple d'utilisation du mode interactif :**

- A l'aide de Jacl :

```
$AdminTask deployEventServiceMdb {-interactive}
```

- A l'aide d'une chaîne Jython :

```
AdminTask.deployEventServiceMdb(['-interactive'])
```

- A l'aide d'une liste Jython :

```
AdminTask.deployEventServiceMdb(['-interactive'])
```

---

## setEventServiceJmsAuthAlias

### Fonction

La commande `setEventServiceJmsAuthAlias` est une commande administrative Common Event Infrastructure qui est disponible pour l'objet `AdminTask`. Cette commande permet de mettre à jour l'alias d'authentification employé par les objets JMS du service d'événements sur un serveur ou un cluster. Si l'alias d'authentification JMS n'existe pas, il est créé. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation `WebSphere Application Server Network Deployment`, version 6.1 .

### Paramètres

#### - `nodeName`

Nom du noeud sur lequel mettre à jour l'alias d'authentification JMS du service d'événements. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `serverName`

Nom du serveur sur lequel mettre à jour l'alias d'authentification JMS du service d'événements. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `clusterName`

Nom du cluster sur lequel mettre à jour l'alias d'authentification JMS du service d'événements. N'indiquez pas ce paramètre si les paramètres `nodeName` et `serverName` sont spécifiés.

#### - `userName`

Nom de l'utilisateur à utiliser dans la mise à jour de l'alias d'authentification JMS du service d'événements sur un serveur ou cluster.

**Important :** Vous devez indiquer un ID utilisateur valable; ce champ ne peut pas rester vide.

#### - `password`

Mot de passe de l'utilisateur à utiliser dans la mise à jour de l'alias d'authentification JMS du service d'événements sur un serveur ou un cluster.

**Important :** Vous devez indiquer un mot de passe valable; ce champ ne peut pas rester vide.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de `Jacl` :

```
$AdminTask setEventServiceJmsAuthAlias{-nodeName nodename
-serverName servername username -password pwd}
$AdminTask setEventServiceJmsAuthAlias{-clusterName clustername
-userName username -password pwd}
```

- A l'aide d'une chaîne `Jython` :

```
AdminTask.setEventServiceJmsAuthAlias('[-nodeName nodename
-serverName servername -userName username -password pwd]')
AdminTask.setEventServiceJmsAuthAlias('[-clusterName clustername
-userName username -password pwd]')
```

- A l'aide d'une liste Jython :
 

```
AdminTask.setEventServiceJmsAuthAlias(['-nodeName', 'nodename',
'-serverName', '-servername', '-userName', 'username',
'-password', 'pwd'])
AdminTask.setEventServiceJmsAuthAlias(['-clusterName', 'clustername',
'-userName', 'username', '-password', 'pwd'])
```

#### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :
 

```
$AdminTask setEventServiceJmsAuthAlias {-interactive}
```
- A l'aide d'une chaîne Jython :
 

```
AdminTask.setEventServiceJmsAuthAlias(['-interactive'])
```
- A l'aide d'une liste Jython :
 

```
AdminTask.setEventServiceJmsAuthAlias(['-interactive'])
```

---

## enableEventService

### Fonction

La commande `enableEventService` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Cette commande permet d'activer le démarrage du service Événements à l'issue du prochain redémarrage du ou des serveurs désignés par les paramètres `nodeName`, `serverName` ou `clusterName`. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

### DITA

### Paramètres

- **nodeName**  
Nom du noeud sur lequel activer le service Événements. Si ce paramètre est défini, le paramètre **serverName** doit l'être également. N'indiquez pas ce paramètre si le paramètre **clusterName** est spécifié.
- **serverName**  
Nom du serveur sur lequel activer le service Événements. Vous devez indiquer ce paramètre si le paramètre **nodeName** est spécifié. N'indiquez pas ce paramètre si le paramètre **clusterName** est spécifié.
- **clusterName**  
Nom du cluster sur lequel activer le service Événements. N'indiquez pas ce paramètre si les paramètres **nodeName** et **serverName** sont spécifiés.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :
 

```
$AdminTask enableEventService {-nodeName nodename -serverName servername}

$AdminTask enableEventService {-clusterName clustername}
```
- A l'aide d'une chaîne Jython :
 

```
AdminTask.enableEventService(['-nodeName nodename -serverName servername'])

AdminTask.enableEventService(['-clusterName clustername'])
```

- A l'aide d'une liste Jython :  
`AdminTask.enableEventService(['-nodeName', 'nodename',  
'-serverName', '-servername'])`  
`AdminTask.enableEventService(['-clusterName', 'clustername'])`

**Exemple d'utilisation du mode interactif :**

- A l'aide de Jacl :  
`$AdminTask enableEventService {-interactive}`
- A l'aide d'une chaîne Jython :  
`AdminTask.enableEventService(['-interactive'])`
- A l'aide d'une liste Jython :  
`AdminTask.enableEventService(['-interactive'])`

---

## disableEventService

### Fonction

La commande `disableEventService` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Cette commande permet de désactiver le démarrage du service Événements à l'issue du prochain redémarrage du ou des serveurs désignés par les paramètres `nodeName`, `serverName` ou `clusterName`. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

### Paramètres

#### - nodeName

Nom du noeud sur lequel désactiver le service Événements. Si ce paramètre est défini, le paramètre **serverName** doit l'être également. N'indiquez pas ce paramètre si le paramètre **clusterName** est spécifié.

#### - serverName

Nom du serveur sur lequel désactiver le service Événements. Vous devez indiquer ce paramètre si le paramètre **nodeName** est spécifié. N'indiquez pas ce paramètre si le paramètre **clusterName** est spécifié.

#### - clusterName

Nom du cluster sur lequel désactiver le service Événements. N'indiquez pas ce paramètre si les paramètres **nodeName** et **serverName** sont spécifiés.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask disableEventService {-nodeName nodename -serverName servername}

$AdminTask disableEventService {-clusterName clustername}
```
- A l'aide d'une chaîne Jython :

```
AdminTask.disableEventService(['-nodeName nodename -serverName servername'])

AdminTask.disableEventService(['-clusterName clustername'])
```
- A l'aide d'une liste Jython :

```
AdminTask.disableEventService(['-nodeName', 'nodename',
'-serverName', '-servername'])

AdminTask.disableEventService(['-clusterName', 'clustername'])
```

#### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask disableEventService {-interactive}
```
- A l'aide d'une chaîne Jython :

```
AdminTask.disableEventService(['-interactive'])
```
- A l'aide d'une liste Jython :

```
AdminTask.disableEventService(['-interactive'])
```



---

## showEventServiceStatus

### Fonction

La commande `showEventServiceStatus` est une commande administrative Common Event Infrastructure qui est disponible pour l'objet `AdminTask`. Cette commande renvoie l'état du service d'événements dans un serveur ou un cluster. Si la tâche est exécutée sans paramètre, l'état de tous les services d'événements est affiché. Pour filtrer la liste des services d'événements à afficher, spécifiez les paramètres `nodeName`, `serverName`, ou `clusterName`. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation `WebSphere Application Server Network Deployment`, version 6.1 .

### Paramètres

#### - nodeName

Ce paramètre permet d'afficher uniquement l'état des services d'événements appartenant au noeud spécifié. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - serverName

Ce paramètre permet d'afficher uniquement l'état des services d'événements appartenant au serveur spécifié. Vous pouvez utiliser ce paramètre avec le paramètre `nodeName` pour afficher l'état du service d'événements appartenant au noeud et au serveur spécifiés. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - clusterName

Ce paramètre permet d'afficher uniquement l'état des services d'événements appartenant au cluster spécifié. Vous ne devez pas spécifier ce paramètre si le paramètre `nodeName` ou `serverName` est spécifié.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask showEventServiceStatus {-nodeName nodename -serverName servername}

$AdminTask showEventServiceStatus {-clusterName clustername}
```
- A l'aide d'une chaîne Jython :

```
AdminTask.showEventServiceStatus(['-nodeName nodename -serverName servername'])

AdminTask.showEventServiceStatus(['-clusterName clustername'])
```
- A l'aide d'une liste Jython :

```
AdminTask.showEventServiceStatus(['-nodeName', 'nodename',
'-serverName', '-servername'])

AdminTask.showEventServiceStatus(['-clusterName', 'clustername'])
```

#### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask showEventServiceStatus {-interactive}
```
- A l'aide d'une chaîne Jython :

```
AdminTask.showEventServiceStatus(['-interactive'])
```
- A l'aide d'une liste Jython :

```
AdminTask.showEventServiceStatus(['-interactive'])
```

---

## removeEventService

### Fonction

La commande `removeEventService` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Utilisez cette commande pour supprimer le service Événements du serveur ou du cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

### Paramètres

#### - `nodeName`

Nom du noeud duquel supprimer le service d'événements. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `serverName`

Nom du serveur duquel supprimer le service d'événements. Vous devez indiquer ce paramètre si le paramètre `nodeName` est spécifié. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `clusterName`

Nom du cluster duquel supprimer le service d'événements. N'indiquez pas ce paramètre si les paramètres `nodeName` et `serverName` sont spécifiés.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask removeEventService {-nodeName nodename -serverName servername}

$AdminTask removeEventService {-clusterName clustername}
```
- A l'aide d'une chaîne Jython :

```
AdminTask.removeEventService(['-nodeName nodename -serverName servername'])

AdminTask.removeEventService(['-clusterName clustername'])
```
- A l'aide d'une liste Jython :

```
AdminTask.removeEventService(['-nodeName', 'nodename',
'-serverName', '-servername'])

AdminTask.removeEventService(['-clusterName', 'clustername'])
```

#### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask removeEventService {-interactive}
```
- A l'aide d'une chaîne Jython :

```
AdminTask.removeEventService(['-interactive'])
```
- A l'aide d'une liste Jython :

```
AdminTask.removeEventService(['-interactive'])
```

---

## removeEventServiceMdb

### Fonction

La commande `removeEventServiceMdb` est une commande administrative Common Event Infrastructure qui est disponible pour l'objet `AdminTask`. Cette commande permet de supprimer le bean géré par message du service d'événements d'un serveur ou d'un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation `WebSphere Application Server Network Deployment`, version 6.1 .

### Paramètres

#### - nodeName

Nom du noeud duquel supprimer le bean géré par message du service d'événements. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - serverName

Nom du serveur duquel supprimer le bean géré par message du service d'événements. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - clusterName

Nom du cluster duquel supprimer le bean géré par message du service d'événements. N'indiquez pas ce paramètre si les paramètres `nodeName` et `serverName` sont spécifiés.

#### - applicationName

Nom de l'application du bean géré par message du service d'événements à supprimer du serveur ou cluster.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de `Jacl` :

```
$AdminTask removeEventServiceMdb {-applicationName appname -nodeName nodename
-serverName servername}
$AdminTask removeEventServiceMdb {-applicationName appname
-clusterName clustername}
```
- A l'aide d'une chaîne `Jython` :

```
AdminTask.removeEventServiceMdb(['-applicationName appname
-nodeName nodename -serverName servername'])
AdminTask.removeEventServiceMdb(['-applicationName appname
-clusterName clustername'])
```
- A l'aide d'une liste `Jython` :

```
AdminTask.removeEventServiceMdb (['-applicationName', 'appname',
'-nodeName', 'nodename', '-serverName', 'servername'])
AdminTask.removeEventServiceMdb (['-applicationName', 'appname',
'-clusterName', 'clustername'])
```

### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :  
`$AdminTask removeEventServiceMdb {-interactive}`
- A l'aide d'une chaîne Jython :  
`AdminTask.removeEventServiceMdb('[-interactive]')`
- A l'aide d'une liste Jython :  
`AdminTask.removeEventServiceMdb(['-interactive'])`

---

## removeEventServiceDB2DB

### Fonction

La commande `removeEventServiceDB2DB` est une commande d'administration Common Event Infrastructure qui s'applique à l'objet `AdminTask`. Utilisez cette commande pour supprimer la base de données et les sources de données du service Événements pour DB2 à partir d'un serveur ou d'un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation relative à WebSphere Application Server Network Deployment, version 6.1.

### Paramètres

#### - `removeDB`

La commande supprime la base de données lorsque ce paramètre est défini sur `true` et ne la supprime pas s'il est défini sur `false`. Pour supprimer la base de données, le serveur actuel doit être déjà configuré en vue de l'exécution des commandes de base de données.

#### - `nodeName`

Nom du noeud qui contient le serveur duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `serverName`

Nom du serveur sur lequel la source de données du service d'événements doit être supprimée. Si ce paramètre est spécifié sans le paramètre `nodeName`, la commande utilisera le nom du noeud du profil WebSphere. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `clusterName`

Nom du cluster duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est spécifié, alors les paramètres `serverName` et `nodeName` ne doivent pas être définis. Ne spécifiez pas ce paramètre si les paramètres `serverName` et `nodeName` sont spécifiés.

#### - `dbUser`

ID Utilisateur DB2 ayant les privilèges adéquats pour créer et supprimer les bases de données. Ce paramètre doit être défini quand le paramètre `removeDB` est défini sur `true`. La valeur par défaut est `db2inst1` s'il n'est pas défini.

#### - `dbPassword`

Mot de passe DB2. Ce paramètre doit être défini quand le paramètre `removeDB` est défini sur `true`.

## - dbScriptDir

Répertoire contenant les scripts de base de données générés par la commande de configuration de la base de données du service Événements. En cas de spécification, cette commande va exécuter les scripts dans ce répertoire pour supprimer la base de données de service d'événements. Le répertoire de sortie du script de base de données par défaut est *profile\_root/databases/event/node/server/dbscripts/db2*.

## Exemple

### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask removeEventServiceDB2DB {-removeDB true -nodeName nodename
-serverName servername -dbUser db2inst1 -dbPassword dbpassword }
```

- A l'aide d'une chaîne Jython :

```
AdminTask.removeEventServiceDB2DB(['-removeDB true -nodeName nodename
-serverName servername -dbUser db2inst1 -dbPassword dbpassword'])
```

- A l'aide d'une liste Jython :

```
AdminTask.removeEventServiceDB2DB(['-removeDB', 'true', '-nodeName', 'nodename',
'-serverName', 'servername', '-dbUser', 'db2inst1', '-dbPassword', 'dbpassword'])
```

### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask removeEventServiceDB2DB -interactive
```

- A l'aide d'une chaîne Jython :

```
AdminTask.removeEventServiceDB2DB(['-interactive'])
```

- A l'aide d'une liste Jython :

```
AdminTask.removeEventServiceDB2DB(['-interactive'])
```

---

## removeEventServiceDB2iSeriesDB

### Fonction

La commande `removeEventServiceDB2iSeriesDB` est une commande administrative Common Event Infrastructure et est disponible pour l'objet `AdminTask`. Cette commande supprime les sources de données DB2 for iSeries d'un serveur ou d'un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation WebSphere Application Server Network Deployment, version 6.1 .

### Paramètres

#### - `nodeName`

Nom du noeud qui contient le serveur duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `serverName`

Nom du serveur sur lequel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini sans le paramètre `nodeName`, la commande utilisera le nom de noeud du profil WebSphere actif. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `clusterName`

Nom du cluster duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas l'être. N'indiquez pas ce paramètre si les paramètres `serverName` et `nodeName` sont spécifiés.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
$AdminTask removeEventServiceDB2iSeriesDB {-nodeName nodename
-serverName servername }
```
- A l'aide d'une chaîne Jython :

```
AdminTask.removeEventServiceDB2iSeriesDB(['-nodeName nodename
-serverName servername'])
```
- A l'aide d'une liste Jython :

```
AdminTask.removeEventServiceDB2iSeriesDB(['-nodeName', 'nodename',
'-serverName', 'servername'])
```

#### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask removeEventServiceDB2iSeriesDB -interactive
```
- A l'aide d'une chaîne Jython :

```
AdminTask.removeEventServiceDB2iSeriesDB(['-interactive'])
```
- A l'aide d'une liste Jython :

```
AdminTask.removeEventServiceDB2iSeriesDB(['-interactive'])
```

---

## removeEventServiceDB2ZOSDB

### Fonction

La commande `removeEventServiceDB2ZOSDB` est une commande administrative Common Event Infrastructure et est disponible pour l'objet `AdminTask`. Cette commande permet de supprimer la base de données et les sources de données du service d'événements pour DB2 z/OS d'un serveur ou d'un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation WebSphere Application Server Network Deployment, version 6.1.

### Paramètres

#### - `removeDB`

Cette commande supprime la base de données quand ce paramètre est défini sur `true` et ne le fait pas s'il est défini sur `false`. Pour supprimer la base de données, le serveur actuel doit avoir été configuré en vue de l'exécution des commandes de base de données.

#### - `nodeName`

Nom du noeud qui contient le serveur duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `serverName`

Nom du serveur sur lequel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini sans le paramètre `nodeName`, la commande utilisera le nom de noeud du profil WebSphere actif. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `clusterName`

Nom du cluster duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas l'être. N'indiquez pas ce paramètre si les paramètres `serverName` et `nodeName` sont spécifiés.

#### - `dbName`

Nom de la base de données DB2. Sur la machine cliente DB2, il s'agit du nom de la base de données cataloguée. Sur le serveur z/OS natif, il s'agit du nom du sous-système de la base de données. Ce paramètre doit être défini quand le paramètre `removeDB` est défini sur `true`. La valeur par défaut est `event` s'il n'est pas défini.

#### - `dbUser`

ID utilisateur DB2 ayant les privilèges adéquats pour créer et supprimer les bases de données. Ce paramètre doit être défini quand le paramètre `removeDB` est défini sur `true`.

#### - `dbPassword`

Mot de passe DB2. Ce paramètre doit être défini quand le paramètre `removeDB` est défini sur `true`.

#### - `dbScriptDir`

Répertoire contenant les scripts de base de données générés par la commande de configuration de la base de données de service d'événements. En cas de spécification, cette commande va exécuter les scripts dans ce répertoire pour supprimer la base de données de service d'événements. Le répertoire de sortie par défaut du script de la base de données est `racine_profil/databases/event/noeud/serveur/dbscripts/db2zos`.

## Exemple

### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :  

```
$AdminTask removeEventServiceDB2ZOSDB {-removeDB true -nodeName nodename
-serverName servername -dbUser db2user -dbPassword dbpassword}
```
- A l'aide d'une chaîne Jython :  

```
AdminTask.removeEventServiceDB2ZOSDB(['-removeDB true -nodeName nodename
-serverName servername -dbUser db2user -dbPassword dbpassword'])
```
- A l'aide d'une liste Jython :  

```
AdminTask.removeEventServiceDB2ZOSDB(['-removeDB', 'true',
'-nodeName', 'nodename', '-serverName', 'servername',
'-dbUser', 'db2user', '-dbPassword', 'dbpassword'])
```

### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :  

```
$AdminTask removeEventServiceDB2ZOSDB -interactive
```
- A l'aide d'une chaîne Jython :  

```
AdminTask.removeEventServiceDB2ZOSDB(['-interactive'])
```
- A l'aide d'une liste Jython :  

```
AdminTask.removeEventServiceDB2ZOSDB(['-interactive'])
```



---

## removeEventServiceDerbyDB

### Fonction

La commande `removeEventServiceDerbyDB` est une commande administrative Common Event Infrastructure qui est disponible pour l'objet `AdminTask`. Cette commande permet de supprimer la base de données et la source de données du service d'événements pour Derby d'un serveur ou d'un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation WebSphere Application Server Network Deployment, version 6.1 .

### Paramètres

#### - `removeDB`

La commande supprime la base de données quand ce paramètre est défini sur `true` et ne le fait pas s'il est défini sur `false`. Pour supprimer la base de données, le serveur actuel doit avoir été configuré en vue de l'exécution des commandes de base de données.

#### - `nodeName`

Nom du noeud qui contient le serveur duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `serverName`

Nom du serveur sur lequel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini sans le paramètre `nodeName`, la commande utilisera le nom de noeud du profil WebSphere actif. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `clusterName`

Nom du cluster duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas l'être. N'indiquez pas ce paramètre si les paramètres `serverName` et `nodeName` sont spécifiés.

#### - `dbScriptDir`

Répertoire contenant les scripts de base de données générés par la commande de configuration de la base de données de service d'événements. En cas de spécification, cette commande va exécuter les scripts dans ce répertoire pour supprimer la base de données de service d'événements. Le répertoire de sortie par défaut du script de la base de données est `racine_profil/databases/event/noeud/serveur/dbscripts/derby`.

### Exemple

#### Exemple d'utilisation du mode par lots :

- A l'aide de `Jacl` :

```
$AdminTask removeEventServiceDerbyDB {-removeDB true -nodeName nodename -serverName servername}
```
- A l'aide d'une chaîne `Jython` :

```
AdminTask.removeEventServiceDerbyDB(['-removeDB true -nodeName nodename -serverName servername'])
```
- A l'aide d'une liste `Jython` :

```
AdminTask.removeEventServiceDerbyDB(['-removeDB', 'true', '-nodeName', 'nodename', '-serverName', 'servername'])
```

### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :  
`$AdminTask removeEventServiceDerbyDB -interactive`
- A l'aide d'une chaîne Jython :  
`AdminTask.removeEventServiceDerbyDB( '[-interactive]')`
- A l'aide d'une liste Jython :  
`AdminTask.removeEventServiceDerbyDB(['-interactive'])`

---

## removeEventServiceInformixDB

### Fonction

La commande `removeEventServiceInformixDB` est une commande administrative Common Event Infrastructure qui est disponible pour l'objet `AdminTask`. Utilisez cette commande pour créer la base de données et les sources de données du service d'événements pour Informix sur un serveur ou un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation WebSphere Application Server Network Deployment, version 6.1.

### Paramètres

#### - `removeDB`

Cette commande supprime la base de données quand ce paramètre est défini sur `true` et ne le fait pas s'il est défini sur `false`. Pour supprimer la base de données, le serveur actuel doit avoir été configuré en vue de l'exécution des commandes de base de données.

#### - `nodeName`

Nom du noeud qui contient le serveur duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `serverName`

Nom du serveur sur lequel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini sans le paramètre `nodeName`, la commande utilisera le nom de noeud du profil WebSphere actif. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `clusterName`

Nom du cluster duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas l'être. N'indiquez pas ce paramètre si les paramètres `serverName` et `nodeName` sont spécifiés.

#### - `dbScriptDir`

Répertoire contenant les scripts de base de données générés par la commande de configuration de la base de données du service d'événements. En cas de spécification, cette commande va exécuter les scripts dans ce répertoire pour supprimer la base de données de service d'événements. Le répertoire de sortie par défaut du script de la base de données est `racine_profil/databases/event/noeud/serveur/dbscripts/informix`.

## Exemple

### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :  

```
$AdminTask removeEventServiceInformixDB {-removedB true -nodeName nodename
-serverName servername}
```
- A l'aide d'une chaîne Jython :  

```
AdminTask.removeEventServiceInformixDB(['-removedB true -nodeName nodename
-serverName servername'])
```
- A l'aide d'une liste Jython :  

```
AdminTask.removeEventServiceInformixDB(['-removedB', 'true',
'-nodeName', 'nodename', '-serverName', 'servername'])
```

### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :  

```
$AdminTask removeEventServiceInformixDB -interactive
```
- A l'aide d'une chaîne Jython :  

```
AdminTask.removeEventServiceInformixDB(['-interactive'])
```
- A l'aide d'une liste Jython :  

```
AdminTask.removeEventServiceInformixDB(['-interactive'])
```

---

## removeEventServiceOracleDB

### Fonction

La commande `removeEventServiceOracleDB` est une commande administrative Common Event Infrastructure qui est disponible pour l'objet `AdminTask`. Cette commande permet de supprimer les tables et sources de données du service d'événements d'Oracle pour un serveur ou un cluster. Cette commande ne supprime pas la base de données. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation WebSphere Application Server Network Deployment, version 6.1 .

### Paramètres

#### - `removeDB`

Cette commande supprime les tables du service d'événements quand ce paramètre est défini sur `true` et ne le fait pas s'il est défini sur `false`.

#### - `nodeName`

Nom du noeud qui contient le serveur duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `serverName`

Nom du serveur sur lequel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini sans le paramètre `nodeName`, la commande utilisera le nom de noeud du profil WebSphere actif. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `clusterName`

Nom du cluster duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas l'être. N'indiquez pas ce paramètre si les paramètres `serverName` et `nodeName` sont spécifiés.

#### - `sysUser`

ID utilisateur système de la base de données Oracle. La valeur par défaut est `sys` s'il n'est pas défini.

#### - `sysPassword`

Le mot de passe pour l'utilisateur spécifié par le paramètre `sysUser`.

#### - `dbScriptDir`

Répertoire contenant les scripts de base de données générés par la commande de configuration de la base de données du service d'événements. En cas de spécification, cette commande va exécuter les scripts dans ce répertoire pour supprimer la base de données de service d'événements. Le répertoire de sortie par défaut du script de la base de données est `racine_profil/databases/event/noeud/serveur/dbscripts/oracle`.

## Exemple

### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :  

```
$AdminTask removeEventServiceOracleDB {-removeDB true -nodeName nodename
-serverName servername -sysUser sys -sysPassword syspassword}
```
- A l'aide d'une chaîne Jython :  

```
AdminTask.removeEventServiceOracleDB(['-removeDB true -nodeName nodename
-serverName servername -sysUser sys -sysPassword syspassword'])
```
- A l'aide d'une liste Jython :  

```
AdminTask.removeEventServiceOracleDB(['-removeDB', 'true',
'-nodeName', 'nodename', '-serverName', 'servername',
'-sysUser', 'sys', '-sysPassword', 'syspassword'])
```

### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :  

```
$AdminTask removeEventServiceOracleDB -interactive
```
- A l'aide d'une chaîne Jython :  

```
AdminTask.removeEventServiceOracleDB(['-interactive'])
```
- A l'aide d'une liste Jython :  

```
AdminTask.removeEventServiceOracleDB(['-interactive'])
```

---

## removeEventServiceSQLServerDB

### Fonction

La commande `removeEventServiceSQLServerDB` est une commande administrative Common Event Infrastructure qui est disponible pour l'objet `AdminTask`. Cette commande permet de supprimer la base de données et les sources de données du service d'événements pour SQL Server d'un serveur ou d'un cluster. Pour plus d'informations sur l'objet `AdminTask`, voir la documentation WebSphere Application Server Network Deployment, version 6.1 .

### Paramètres

#### - `removeDB`

Cette commande supprime la base de données quand ce paramètre est défini sur `true` et ne le fait pas s'il est défini sur `false`. Pour supprimer la base de données, le serveur actuel doit avoir été configuré en vue de l'exécution des commandes de base de données.

#### - `nodeName`

Nom du noeud qui contient le serveur duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, le paramètre `serverName` doit l'être également. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `serverName`

Nom du serveur sur lequel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini sans le paramètre `nodeName`, la commande utilisera le nom de noeud du profil WebSphere actif. N'indiquez pas ce paramètre si le paramètre `clusterName` est spécifié.

#### - `clusterName`

Nom du cluster duquel la source de données du service d'événements doit être supprimée. Si ce paramètre est défini, les paramètres `serverName` et `nodeName` ne doivent pas l'être. Ne spécifiez pas ce paramètre si les paramètres `serverName` et `nodeName` sont spécifiés.

#### - `dbServerName`

Nom du serveur de la base de données SQL Server. Ce paramètre doit être défini quand le paramètre `removeDB` est défini sur `true`.

#### - `dbUser`

ID utilisateur SQL Server qui détiendra les tables de service d'événements. La valeur par défaut est `ceiuser` s'il n'est pas spécifié.

#### - `saUser`

ID utilisateur ayant les privilèges adéquats pour supprimer les bases de données et les utilisateurs. La valeur par défaut est `sa` s'il n'est pas défini.

#### - `saPassword`

Le mot de passe spécifié par le paramètre `saUser`. Ce paramètre est obligatoire quand le paramètre `removeDB` est défini sur `true`.

#### - `dbScriptDir`

Répertoire contenant les scripts de base de données générés par la commande de configuration de la base de données du service d'événements. En cas de spécification, cette commande va exécuter les scripts dans ce répertoire pour supprimer la base de données du service d'événements. Le répertoire de sortie par défaut du script de la base de données est `racine_profil/databases/event/noeud/serveur/dbscripts/sqlserver`.

## Exemple

### Exemple d'utilisation du mode par lots :

- A l'aide de Jacl :

```
*
$AdminTask removeEventServiceSQLServerDB {-removeDB true -nodeName nodename
-serverName servername -dbUser ceiuser -saUser sa -saPassword sapassword
-dbServerName sqlservername}
```
- A l'aide d'une chaîne Jython :

```
AdminTask.removeEventServiceSQLServerDB(['-removeDB true -nodeName nodename
-serverName servername -dbUser ceiuser -saUser sa -saPassword sapassword
-dbServerName sqlservername'])
```
- A l'aide d'une liste Jython :

```
AdminTask.removeEventServiceSQLServerDB(['-removeDB', 'true',
'-nodeName', 'nodename', '-serverName', 'servername',
'-dbUser', 'ceiuser', '-saUser', 'sa', '-saPassword', 'sapassword',
'-dbServerName', 'sqlservername'])
```

### Exemple d'utilisation du mode interactif :

- A l'aide de Jacl :

```
$AdminTask removeEventServiceSQLServerDB -interactive
```
- A l'aide d'une chaîne Jython :

```
AdminTask.removeEventServiceSQLServerDB(['-interactive'])
```
- A l'aide d'une liste Jython :

```
AdminTask.removeEventServiceSQLServerDB(['-interactive'])
```

---

## eventbucket

La commande **eventbucket** permet d'afficher ou de modifier la configuration des compartiments de la base de données d'événements.

### Fonction

Affichage ou modification de la configuration des compartiments de la base de données d'événements.

**eventbucket** [-status] [-change]

### Description

La commande eventbucket permet d'afficher ou de modifier la configuration des compartiments de la base de données d'événements. Les compartiments sont utilisés par l'utilitaire de purge rapide pour purger les anciennes données d'événement de la base de données d'événements. L'exécution de cette commande vous permet de déterminer la configuration des compartiments en cours et de permuter les compartiments actif et inactif.

**Sécurité :** Si la sécurité WebSphere est activée, vous pouvez afficher ou modifier la configuration des compartiments de la base de données d'événements uniquement si votre ID utilisateur est mappé au rôle eventAdministrator.

### Paramètres

#### -status

Affichage d'informations sur la configuration des compartiments en cours, notamment les paramètres du compartiment actif et l'intervalle de vérification des compartiments (fréquence de la vérification effectuée par le plug-in du magasin de données pour identifier le compartiment actif).

#### -change

Permutation des compartiments afin que le compartiment actif devienne inactif et le compartiment inactif devienne actif. Le compartiment inactif doit être vide avant de pouvoir utiliser cette option.

### Exemples

Cet exemple permet d'afficher la configuration des compartiments en cours :

```
eventbucket -status
```

Cet exemple permet de permuter les compartiments actif et inactif :

```
eventbucket -change
```



---

## eventpurge

La commande **eventpurge** permet de supprimer des événements de la base de données d'événements.

### Fonction

Suppression d'événements dans la base de données d'événements.

```
eventpurge [-seconds secondes | -end heure_fin] [-group groupe_événements]
[-severity gravité] [-extensionname nom_extension] [-start heure_début] [-size taille]
```

### Description

La commande **eventpurge** permet de supprimer des événements de la base de données d'événements. Vous pouvez supprimer tous les événements de la base de données d'événements ou limiter la suppression à certains événements correspondant à certains critères.

**Sécurité :** Si la sécurité WebSphere est activée, vous pouvez supprimer des événements uniquement si votre ID utilisateur est mappé au rôle `eventAdministrator`.

### Paramètres

**-seconds** *secondes*

Ancienneté minimale des événements à supprimer. La valeur *secondes* doit être un nombre entier. Seuls les événements antérieurs au nombre de *secondes* indiqué sont supprimés. Ce paramètre est obligatoire si vous ne définissez pas le paramètre **-end**.

**-end** *heure\_fin*

Heure de fin du groupe d'événements que vous souhaitez supprimer. Seuls les événements générés avant l'heure indiquée sont supprimés. La valeur *heure\_fin* doit être au format XML `dateTime` (CCYY-MM-DDThh:mm:ss ). Par exemple, midi le 1 janvier 2006 en heure EST (Eastern Standard Time) correspond à `2006-01-01T12:00:00-05:00`. Pour plus d'informations sur le type de données `dateTime`, voir la page relative au Schéma XML sur le site [www.w3.org](http://www.w3.org).

Ce paramètre est obligatoire si vous ne définissez pas le paramètre **-seconds**.

**-group** *groupe\_événements*

Groupe d'événements dans lequel vous souhaitez supprimer des événements. La valeur *groupe\_événements* doit être le nom d'un groupe d'événements défini dans la configuration de l'infrastructure d'événement commune (CEI). Ce paramètre est facultatif.

**-severity** *gravité*

Gravité des événements que vous souhaitez supprimer. La valeur *gravité* doit être un nombre entier. Seuls sont supprimés les événements dont la gravité est égale à la valeur que vous indiquez. Ce paramètre est facultatif.

**-extensionname** *nom\_extension*

Nom de l'extension des événements que vous souhaitez inclure à la suppression. Ce paramètre permet de limiter la suppression aux événements d'un certain type. Seuls sont supprimés les événements dont la propriété `extensionName` est égale à *nom\_extension*. Ce paramètre est facultatif.

**-start** *heure\_début*

Heure de début du groupe des événements à supprimer. Seuls les événements générés après l'heure indiquée sont supprimés. La valeur *heure\_début* doit être au format XML dateTime (CCYY-MM-DDThh:mm:ss ). Ce paramètre est facultatif.

**-size** *taille*

Nombre d'événements à supprimer dans une transaction. La valeur *taille* doit être un nombre entier. Après la suppression de ce nombre d'événements, la commande valide la transaction avant de passer à une nouvelle transaction. Ce paramètre est facultatif.

## Exemple

L'exemple suivant permet de supprimer de la base de données tous les événements dont la gravité est égale à 20 (sans effet) et qui ont été générés depuis plus de 10 minutes.

```
eventpurge -group "All events" -severity 20 -seconds 600
```

---

## Remarques

Ces informations concernent initialement des produits et services fournis aux Etats-Unis.

Il se peut qu'IBM ne propose pas les produits, services ou fonctions décrits dans ce document dans d'autres 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 produit, programme ou service fonctionnellement équivalent peut être utilisé s'il n'enfreint aucun droit de propriété intellectuelle 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. Vous pouvez envoyer des demandes de licence, en écrivant à :

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

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

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

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

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

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier 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 matériels de ces sites Web ne font pas partie des matériels utilisés dans ce produit IBM et l'utilisation de ces sites Web s'effectue à vos risques et périls.

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  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
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.

Toutes données de performance contenues dans ce document ont été déterminées dans un environnement contrôlé. De ce fait, les résultats obtenus dans d'autres environnements d'exploitation peuvent varier de manière significative. Certaines mesures peuvent avoir été effectuées sur des systèmes au niveau du développement et il n'existe aucune garantie que ces mesures seront identiques sur des systèmes disponibles de façon générale. En outre, elles peuvent résulter d'extrapolations. Les résultats obtenus peuvent varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations relatives aux produits non IBM ont été obtenues via les fournisseurs de ces produits, leurs annonces publiées ou d'autres sources publiquement disponibles. IBM n'a pas testé ces produits et ne peut pas confirmer avec exactitude les performances, la compatibilité ou toutes autres déclarations relatives aux produits non fournis par IBM. Toute question relative aux fonctions des produits non fournis par 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 contient 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.

#### LICENCE DE COPYRIGHT :

Les présentes informations contiennent des exemples de programmes d'application en langage source illustrant les techniques de programmation sur diverses 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 n'ont pas été intégralement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Toute copie totale ou partielle de ces programmes exemples et des oeuvres qui en sont dérivées doit comprendre une notice de copyright, libellée comme suit : (c) (votre société) (année). Des segments de code sont dérivés des Programmes exemples d'IBM Corp. (c) Copyright IBM Corp. \_entrez l'année ou les années\_. Tous droits réservés.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

### **Documentation sur l'interface de programmation**

Si elle est fournie, la documentation sur l'interface de programmation aide les utilisateurs à créer des applications en utilisant le produit.

Les interfaces de programmation génériques permettent aux utilisateurs d'écrire des applications, qui bénéficient des services proposés par les outils du produit.

Cependant, cette documentation peut également comporter des informations de diagnostic, de modification et de personnalisation. Ces informations de diagnostic, de modification et d'optimisation sont fournies pour faciliter le débogage du logiciel d'application.

**Avertissement** : N'utilisez pas les informations de diagnostic, de modification et d'optimisation en guise d'interface de programmation car elles peuvent être modifiées sans préavis.

## Marques, noms de produits et logos

IBM, le logo IBM, ibm.com, DB2, i5/OS, Informix, iSeries, WebSphere, z/OS, Cloudscape, DB2 Connect et DB2 Universal Database sont des marques d'International Business Machines Corporation aux Etats-Unis et/ou dans certains autres pays.

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

Microsoft et Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque enregistrée de The Open Group 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.

Ce produit inclut un logiciel développé par Eclipse Project(<http://www.eclipse.org>).



IBM WebSphere Process Server for Multiplatforms version 6.1.0



**IBM**