

Netcool/OMNIbus
Version 8.1

*Event Integration Facility - Guide de
référence*



Netcool/OMNIbus
Version 8.1

*Event Integration Facility - Guide de
référence*



Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant à la section «Remarques», à la page 89.

Première édition - Juillet 2014

Réf. US : SC27-6267-00

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

*Compagnie IBM France
Direction Qualité
17, avenue de l'Europe
92275 Bois-Colombes Cedex*

Cette édition s'applique à la version 8, édition 1 d'IBM Tivoli Netcool/OMNibus (numéro de produit 5724-S44) ainsi qu'à toutes les éditions et modifications suivantes, sauf indication contraire dans les nouvelles éditions.

© Copyright IBM Corporation 2003, 2014.

Table des matières

Avis aux lecteurs canadiens v

A propos de cette publication vii

Public visé	vii
Publications	vii
Accessibilité	ix
Formation technique Tivoli	x
Informations de support	x
Conventions utilisées dans cette publication	x

Chapitre 1. Présentation de Tivoli Event

Integration Facility 1

Événements.	1
Adaptateurs	2
Classes d'événement.	3
Fichiers de configuration	3
Serveur d'événements	4
Filtrage d'événements	6
Règles	7

Chapitre 2. Installation de l'Event

Integration Facility 9

Structure de répertoire.	10
Migration des adaptateurs	12
Configuration de l'utilitaire posteifmsg sous z/OS à l'aide d'USS	12

Chapitre 3. Transport des événements 15

Méthodes de distribution d'événement	15
Options de connexion	15
Options de transfert	16
Test et scriptage	21
Réception d'événements pour les applications	26
Envoi d'événements via des pare-feux	27
Distribution d'événements lors d'échecs système	27
Activation du cache	28
Configuration des serveurs de sauvegarde pour la distribution d'événement	29
Utilisation des mots clés de l'associateur de port	30
Configuration d'une application réceptrice générée avec l'interface de programme d'application (API) C	31

Chapitre 4. Génération d'un adaptateur 33

Fichiers d'adaptateur	33
Identification des événements à surveiller	34
Définition de la source.	34
Définitions de classes d'événement	34
Sélection des méthodes de distribution d'événements.	35
Configuration d'une application réceptrice EIF pour SSL	36
Configuration d'une application client EIF pour SSL	38

Programmation de l'adaptateur.	40
Mise à niveau d'adaptateurs existants	42
Interface de programme d'application (API) des fichiers de configuration	42
Interface de programme d'application (API) de communication	42
Remarques spécifiques pour Microsoft Windows	43
Compilation de l'adaptateur généré avec l'API C	43
Liaison de l'adaptateur généré avec l'interface de programme d'application (API) C	44
Installation, configuration et test de l'adaptateur	44
Exécution d'adaptateurs générés avec l'interface de programme d'application (API) Java d'Event Integration Facility	45
Configuration d'adaptateurs pour des environnements internes	46

Chapitre 5. Filtrage des événements à la source. 47

Filtrage à l'aide des fichiers de configuration	47
Filtrage des événements lors d'échecs des systèmes	48
Expressions régulières dans les filtres.	49

Chapitre 6. Résolution des problèmes 51

Journaux des messages	51
Journaux de trace	51
Performances et disponibilité	53
Paramètres de connexion pour le processus de réception	54
Incidents et scénarios usuels.	54
Génération et exécution d'adaptateurs	54
Établissement de connexions au serveur d'événements.	55
Envoi d'événements	55

Annexe A. Interfaces de programme d'application (API). 59

Interface de programme d'application (API) du langage C	59
tec_agent_getenv	59
tec_agent_init.	60
tec_create{EIF_handle	60
tec_create_handle	61
tec_create_handle_c.	62
tec_create_handle_r.	63
tec_destroy_handle	64
tec_erno	64
tec_get_event.	64
tec_put_event.	65
tec_register_callback	66

Annexe B. Utilitaires de l'interface de programme d'application (API) C . . . 67

ed_scan_get_n 67
ed_scan_n 68
ed_sleep 68

Annexe C. Interface de programme d'application (API) du langage Java . . . 71

disconnect 71
disconnect(time) 71
getConfigVal 71
onMessage 72

receiveEvent 72
registerListener 73
sendEvent 73
TECAgent 74
TECEvent 74

Annexe D. Mots clés de configuration 77

Remarques 89
Marques 92

Index 93

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.

A propos de cette publication

Le présent Guide de référence de Tivoli Event Integration Facility contient des éléments de référence pour le kit d'outils de la fonction d'intégration d'événements (Event Integration Facility, EIF).

Public visé

Ce guide explique les concepts visant à développer de manière efficace de nouveaux adaptateurs ou à en modifier des existants. Ce livre est destiné aux développeurs qui possèdent des connaissances en programmation et souhaitent créer des adaptateurs personnalisés et utiliser Tivoli Event Integration Facility dans leurs applications. Ce livre est également utile pour les administrateurs Tivoli Netcool/OMNIbus qui modifient les fichiers de configuration.

Les lecteurs doivent connaître les logiciels suivants :

- Java™ ou langages de programmation C
- UNIX, Microsoft Windows, ou autre système d'exploitation cible

Publications

Cette section dresse la liste des publications de la bibliothèque Tivoli Netcool/OMNIbus ainsi que des documents associés. Elle indique également comment accéder aux publications Tivoli en ligne et comment organiser les publications Tivoli.

Votre bibliothèque Tivoli Netcool/OMNIbus

Les documents suivants sont disponibles dans la bibliothèque Tivoli Netcool/OMNIbus :

- *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*, SC11-7236
Comprend des procédures d'installation et de mise à niveau de Tivoli Netcool/OMNIbus et décrit la manière de configurer la sécurité et les communications des composants. La publication contient également des exemples d'architectures Tivoli Netcool/OMNIbus et explique comment les mettre en oeuvre.
- *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*, SC11-7237
Explique comment effectuer les tâches d'administration à l'aide de l'interface graphique d'administration de Tivoli Netcool/OMNIbus, des outils de ligne de commande et des commandes de processus. La publication contient également des descriptions et des exemples de syntaxe SQL du serveur ObjectServer et des automatisations.
- *Guide d'administration et d'utilisation de l'interface graphique Web d'IBM Tivoli Netcool/OMNIbus*, SC11-7238
Décrit comment exécuter les tâches d'administration et de visualisation d'événements à l'aide de l'interface graphique Web de Tivoli Netcool/OMNIbus.
- *IBM Tivoli Netcool/OMNIbus User's Guide*, SC11-7243
Présente les outils du bureau et décrit les tâches de l'utilisateur liées à la gestion d'événements à l'aide de ces outils.
- *Guide des sondes et des passerelles d'IBM Tivoli Netcool/OMNIbus*, SC11-7239

Contient des informations de présentation et de référence sur les sondes et les passerelles, notamment la syntaxe des fichiers de règles de sonde et des commandes de passerelle.

- *IBM Tivoli Monitoring for Tivoli Netcool/OMNIbus Agent User's Guide*, SC11-7303

Explique comment installer l'agent de moniteur d'état de Tivoli Netcool/OMNIbus et contient des informations de référence sur l'agent.

- *Guide de référence d'IBM Tivoli Netcool/OMNIbus Event Integration Facility*, SC11-7241

Décrit comment développer des adaptateurs d'événements adaptés à votre environnement réseau et aux besoins spécifiques de votre entreprise. Cette publication décrit également comment filtrer des événements à la source.

- *Guide des messages d'erreur d'IBM Tivoli Netcool/OMNIbus*, SC11-7242

Décrit les messages système dans Tivoli Netcool/OMNIbus et les réponses à apporter à ces messages.

- *Guide d'utilisation de l'API d'administration de l'interface graphique Web (WAAPI) d'IBM Tivoli Netcool/OMNIbus*, SC11-7244

Montre comment administrer l'interface graphique Web Tivoli Netcool/OMNIbus à l'aide de l'interface de programmation d'application XML nommée WAAPI

- *Guide de référence de l'interface HTTP ObjectServer d'IBM Tivoli Netcool/OMNIbus*, SC11-7345
Décrit les URI et les comportements communs de l'interface de programme d'application (API) appelée interface HTTP ObjectServer. Décrit comment activer cette API et fournit des exemples de charges de message JSON et de demandes et réponses HTTP.

- *Guide de référence de l'interface OSLC ObjectServer d'IBM Tivoli Netcool/OMNIbus*, SC11-7345
Décrit les services, les ressources et les comportements communs de l'interface de programme d'application (API) OSLC (Open Services for Lifecycle Collaboration), appelée interface OSLC ObjectServer. Décrit comment activer cette API et fournit des exemples de définitions de fournisseur de services, de charges de message RDF/XML et des demandes et réponses HTTP.

Si vous utilisez d'autres produits IBM pour étendre les fonctionnalités de Tivoli Netcool/OMNIbus, tels que DB2, IBM Tivoli Monitoring ou Tivoli Common Reporting, consultez le centre de documentation du produit concerné, afin d'obtenir les publications appropriées.

Publications connexes

Les documents suivants doivent être consultés lors de l'utilisation de IBM® Tivoli Enterprise Console en tant que serveur d'événements. Ils sont disponibles dans la bibliothèque IBM Tivoli Enterprise Console :

- *IBM Tivoli Enterprise Console Adapters Guide*, SC32-1242

Fournit des informations sur les adaptateurs pris en charge, y compris sur l'installation et la configuration de ces adaptateurs.

Accès à la terminologie en ligne

Le site Web de terminologie IBM regroupe la terminologie des bibliothèques de logiciels IBM en un seul emplacement, pour des raisons pratiques. Vous pouvez accéder au site Web de terminologie à l'adresse Web suivante :

<http://www.ibm.com/software/globalization/terminology>

Accès aux publications en ligne

IBM publie la documentation liée à ce produit-ci, ainsi qu'à tous les autres produits Tivoli, dès qu'elle devient disponible ainsi qu'après toute mise à jour, sur le site Web du centre de documentation Tivoli à l'adresse suivante :

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp>

Remarque : Si vous imprimez des documents PDF dans un autre format que le format A4, définissez dans la fenêtre **Fichier > Imprimer** l'option qui permet à Adobe Reader d'imprimer des pages A4 sur votre format de papier.

Commande de publications

Vous pouvez commander de nombreuses publications Tivoli en ligne sur le site Web suivant :

<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>

Vous pouvez également passer votre commande par téléphone en composant l'un des numéros suivants :

- Aux Etats-Unis : 800-879-2755
- Au Canada : 800-426-4968

Dans les autres pays, veuillez prendre contact avec votre représentant commercial pour commander des publications Tivoli. Pour connaître le numéro de téléphone de votre représentant local, procédez comme suit :

1. Allez sur le site Web suivant :
<http://www.ibm.com/e-business/linkweb/publications/servlet/pbi.wss>
2. Sélectionnez votre pays dans la liste et cliquez sur **Go**. La page Welcome to the IBM Publications Center est affichée pour votre pays.
3. Sur le côté gauche de la page, cliquez sur **About this site** pour accéder à une page d'informations qui contient le numéro de téléphone de votre représentant local.

Accessibilité

Les fonctions d'accessibilité aident les utilisateurs atteints d'un handicap physique, tel qu'une mobilité réduite ou une déficience visuelle, à utiliser correctement les applications logicielles.

Avec ce produit, vous pouvez utiliser des technologies d'assistance pour faciliter la navigation dans l'interface par des moyens audio-visuels. Vous pouvez également utiliser le clavier à la place de la souris pour exploiter certaines fonctions de l'interface graphique.

Formation technique Tivoli

Pour obtenir des informations sur les formations techniques Tivoli, reportez-vous au site Web de formation IBM Tivoli à l'adresse suivante :

<http://www.ibm.com/software/tivoli/education>

Informations de support

Si vous rencontrez un problème avec le logiciel IBM, vous pouvez le résoudre rapidement. IBM vous propose les solutions suivantes pour obtenir le support dont vous avez besoin :

En ligne

Accédez au site service de support logiciel IBM à l'adresse <http://www.ibm.com/software/support/probsub.html>, puis suivez les instructions.

IBM Support Assistant

IBM Support Assistant (ISA) est un plan de travail de maintenabilité logicielle gratuit qui vous aide à résoudre les problèmes liés aux applications logicielles IBM. ISA permet d'accéder rapidement aux informations de support et aux outils de maintenabilité pour identifier les problèmes. Pour installer le logiciel ISA, rendez-vous à l'adresse <http://www.ibm.com/software/support/isa>.

Documentation

Si vous avez une suggestion pour améliorer le contenu ou l'organisation de ce guide, envoyez-la à l'équipe de Tivoli Netcool/OMNIBus Information Development, à l'adresse :

<mailto://L3MMDOCS@uk.ibm.com>

Conventions utilisées dans cette publication

Cette publication utilise plusieurs conventions pour les actions et les termes spéciaux, ainsi que pour les commandes et les chemins d'accès liés au système d'exploitation.

Conventions typographiques

Cette publication utilise les conventions typographiques suivantes :

Gras

- Commandes en minuscules et commandes à casse mixte pour mieux les distinguer dans le texte
- Contrôles d'interface (cases à cocher, boutons de commande, boutons radio, flèches d'incrément, zones, dossiers, icônes, zones de liste, éléments de zone de liste, listes à plusieurs colonnes, conteneurs, options de menu, noms de menu, onglets, feuilles de propriétés), libellés (tels que **Conseil :** et **Remarques relatives au système d'exploitation :**)
- Mots clés et paramètres dans le texte

Italique

- Citations (exemples : titres de publications, disquettes et CD)
- Mots définis dans le texte (exemple : une ligne spécialisée est appelée ligne *point-à-point*)

- Mise en évidence de mots et de lettres (exemples : "Utilisez le mot *que* pour introduire une clause restrictive."; "L'adresse LUN doit commencer par la lettre L.")
- Nouveaux termes rencontrés dans un texte (sauf dans une liste de définitions) : une *vue* est un cadre situé dans un espace de travail qui contient des données
- Variables et valeurs que vous devez fournir : ... où *nom* représente....

Espacement fixe

- Exemples et extraits de code
- Noms de fichiers, mots clés de programmation et autres éléments qu'il est difficile de distinguer dans du texte
- Texte de message et invites adressées à l'utilisateur
- Texte que l'utilisateur doit entrer
- Valeurs d'arguments ou d'options de commande

Variables et chemins d'accès liés au système d'exploitation

Cette publication utilise la convention UNIX pour la définition des variables d'environnement et la notation des répertoires.

Lorsque vous utilisez la ligne de commande Windows, remplacez *\$variable* par *%variable%* pour les variables d'environnement. De la même façon, remplacez chaque barre oblique (/) par une barre oblique inversée (\) dans les chemins de répertoire. Par exemple, sur les systèmes UNIX, la variable d'environnement \$NCHOME désigne le chemin du répertoire de base de Netcool. Sur les systèmes Windows, la variable d'environnement %NCHOME% désigne le chemin d'accès au répertoire de base de Netcool. Les noms de variables d'environnement ne sont pas toujours identiques dans les environnements Windows et UNIX. Par exemple, dans les environnements Windows, %TEMP% est l'équivalent de \$TMPDIR dans les environnements UNIX.

Si vous utilisez l'interpréteur de commandes shell sur un système Windows, vous pouvez utiliser les conventions UNIX.

Chapitre 1. Présentation de Tivoli Event Integration Facility

Event Integration Facility est un kit d'outils qui développe les types d'événements et les informations système que vous pouvez surveiller. Vous pouvez l'utiliser pour développer vos propres adaptateurs, qui sont personnalisés pour votre environnement. Event Integration Facility est un module d'installation distinct du produit Tivoli Netcool/OMNIBus principal.

Event Integration Facility contient une interface de programme d'application (API) d'événement à utiliser avec les langages de programmation Java et C. Ce produit contient également une fonction de débogage qui vérifie la syntaxe des événements et envoie les événements à un fichier.

Vous pouvez utiliser Event Integration Facility pour réaliser les tâches suivantes :

- Indiquez les données sur l'événement à envoyer à l'ObjectServer à des fins de traitement.
- Créer un adaptateur pour filtrer, convertir et réacheminer les données sur l'événement vers le serveur d'événements.
- Filtrer des événements à proximité de la source.
- Créer une application qui peut recevoir des événements.

Événements

L'unité centrale d'informations est l'événement. Un événement est une modification de l'état d'une ressource système ou d'une application. Les événements peuvent être générés pour des problèmes ou l'achèvement de tâches avec succès.

Les événements fournissent divers types d'informations, par exemple lorsqu'un hôte est hors tension, lorsque quelqu'un tente sans succès de se connecter à un hôte en tant qu'administrateur ou lorsqu'un disque dur est presque saturé. Les événements peuvent également être générés pour effacer d'autres événements.

Un événement commence comme un message d'erreur, une interruption ou une information similaire qui s'affichent ou sont écrits dans un fichier. Les informations fournies dans un événement dépendent de la source. Certaines sources fournissent des informations détaillées sur un événement, alors que d'autres sources sont brèves dans leurs descriptions. Le rôle de l'adaptateur vise à convertir tous les événements en un format qui fournit une certaine cohérence des informations, notamment la source, l'emplacement, la date et l'heure d'un événement.

Certains événements, notamment les interruptions, contiennent des données que vous ne pouvez pas lire. Dans ces cas, l'adaptateur doit convertir les données en un format que vous pouvez lire afin que le traitement des informations puisse se poursuivre. Certaines sources, notamment un fichier journal système, fournissent des données que vous pouvez lire sans traduction automatique.

Les adaptateurs convertissent les données sur l'événement en un jeu d'attributs. Chaque attribut est prédéfini par l'adaptateur et contient le nom et la valeur d'attribut. L'adaptateur place les informations appropriées dans chaque attribut, puis envoie l'événement au serveur d'événements.

La figure suivante illustre l'évolution d'un événement :

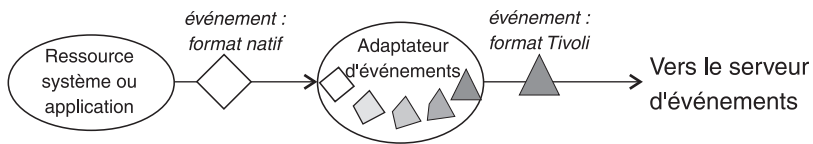


Figure 1. Evolution d'un événement

L'exemple suivant explique comment les informations du fichier journal sont converties en événements. Dans cet exemple, une tentative d'exécution échouée de la commande **su root** sur l'hôte oak est écrite dans le fichier journal système. Vous pouvez lire le format en résultant :

```
Nov 7 08:51:42 oak su: 'su root' failed for don on /dev/tty0
```

L'adaptateur convertit ensuite les informations du fichier journal en un événement de la manière suivante :

```
Su_Failure:
source=LOGFILE;
origin=oak;
date="Nov 7 08:51:42 ";
host=oak;
sub_source=login;
from_user=don;
tty=/dev/tty0
to_user=root;
END
```

Concepts associés:

«Serveur d'événements», à la page 4

Un serveur d'événements est un serveur central qui gère tous les événements collectés par les adaptateurs répartis. Le serveur d'événements crée une entrée pour chaque événement entrant et évalue cet événement par rapport à un ensemble de règles pour déterminer s'il peut répondre à l'événement ou pour modifier l'événement automatiquement.

Adaptateurs

Les adaptateurs sont des processus qui surveillent des sources gérées. Une source peut être une application, comme une base de données, ou une ressource système, comme de l'espace disque. Lorsqu'un adaptateur reçoit des informations de sa source, il formate les informations et les réachemine vers le serveur d'événements.

Pour surveiller une source, notamment une application tierce ou personnalisée, vous devez utiliser Event Integration Facility pour créer un adaptateur et des classes d'événement pour chaque nouvelle source.

Les adaptateurs surveillent les sources de la manière suivante :

- Un adaptateur peut recevoir des messages d'une source qui produit des messages de manière active. Par exemple, les adaptateurs peuvent recevoir des messages qui sont envoyés par les applications logicielles Tivoli.
- Un adaptateur peut vérifier un fichier à des intervalles configurables si la source met à jour un fichier.
- Un adaptateur peut interroger une ressource système ou une condition système à des intervalles configurables, puis interpréter et réacheminer les informations en résultant directement vers le serveur d'événements.

Classes d'événement

Les classes d'événement sont des classifications spécifiques aux événements. Après avoir séparé les informations en classes d'événement, les adaptateurs formatent les informations en messages qui représentent des instances spécifiques de classes d'événement. Ils envoient ensuite les informations au serveur d'événements, qui les traite, ainsi que les informations reçues des autres adaptateurs.

Les classes d'événement peuvent être subdivisées en sous-classes pour faciliter la ventilation des informations afin que des règles plus détaillées puissent être appliquées aux données sur l'événement. Essentiellement, les classes d'événement sont un accord entre l'adaptateur et le serveur d'événements sur le type d'informations envoyées par l'adaptateur au serveur d'événements.

Remarque : Les classes d'événement ne sont pas identiques aux objets Tivoli.

Concepts associés:

«Définitions de classes d'événement», à la page 34

Une tâche importante lors de la création d'un adaptateur est de déterminer les classes d'événement pour les informations à surveiller. Pour vous aider à écrire des règles pour gérer les événements, vous devez rendre les définitions d'événement aussi spécifiques que possible.

Fichiers de configuration

Utilisez le fichier de configuration pour contrôler le comportement des adaptateurs. Vous ne devez pas modifier plusieurs instances d'un adaptateur à exécuter dans différents environnements. Modifiez uniquement les fichiers de configuration.

Le fichier de configuration contrôle les paramètres suivants :

- Les informations doivent être envoyées au serveur d'événements.
- L'interface de connexion utilisée par l'adaptateur.
- L'hôte sur lequel le serveur d'événements se trouve.
- Le filtrage des événements que l'adaptateur doit exécuter.
- Les informations spécifiques à un adaptateur particulier.

Une fois les informations subdivisées en classes d'événement, l'adaptateur envoie les informations au serveur d'événements à des fins de traitement. Pour réduire la charge sur le réseau, utilisez le fichier de configuration pour contrôler les informations envoyées par l'adaptateur au serveur d'événements.

Les paramètres de configuration sont contrôlés par des mots clés. Un modèle de fichier de configuration contenant tous les mots clés disponibles est installé avec Event Integration Facility, dans le répertoire `mi.sc`.

Référence associée:

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

«Structure de répertoire», à la page 10

Structure de répertoire d'Event Integration Facility après l'extraction des fichiers compressés.

Serveur d'événements

Un serveur d'événements est un serveur central qui gère tous les événements collectés par les adaptateurs répartis. Le serveur d'événements crée une entrée pour chaque événement entrant et évalue cet événement par rapport à un ensemble de règles pour déterminer s'il peut répondre à l'événement ou pour modifier l'événement automatiquement.

Le serveur ObjectServer fait office de serveur d'événements pour Tivoli Event Integration Facility.

La sonde EIF permet de transmettre les événements EIF au serveur ObjectServer.

Un adaptateur ne fournit généralement pas de valeur pour tous les attributs d'un événement particulier. Certaines valeurs d'attribut sont fournies par la sonde EIF. Probe for Tivoli EIF peut traiter tous les événements EIF bien formés. Vous avez la possibilité de définir les attributs suivants :

\$ClassName

Nom de classe de l'événement.

\$EventSeqNo

Numéro de séquence de l'événement.

\$EventString

Intégralité de l'événement dans une seule chaîne.

\$date Date de l'événement au format mm/jj/aa.

\$EventClass

Classe de l'événement comme attribuée par la source d'événement.

Remarque : Probe for Tivoli EIF utilise la classe d'événement pour identifier le format du message de chaque type d'événement.

\$hostname

Nom d'hôte du système.

\$msg Contenu du message.

\$peerhost

Deuxième hôte.

\$severity

Gravité de l'alarme.

\$source

Type d'application qui a créé l'événement.

\$sub_origin

Sous-catégorisation optimale de l'origine de l'événement.

\$sub_source

Description détaillée de la source.

Les règles décrivent les actions exécutées lorsque le serveur d'événements reçoit un événement système particulier. Dans certains cas, vous pouvez personnaliser d'autres applications pour recevoir des événements du serveur d'événements.

Le serveur d'événements valide les événements entrants. Si un événement est valide, le serveur d'événements lui attribue un identificateur unique et un

horodatage, puis le stocke dans une base de données d'événement. Le serveur d'événement traite chaque transaction entrante avant de traiter la prochaine transaction.

L'exemple d'événement suivant a été traité par Probe for Tivoli EIF et par le serveur ObjectServer de Tivoli Netcool/OMNIbus :

```
Node Alias: 9.42.19.243
TECEventHandle:
Process Required: 0
Prec. Entity ID: 0
First Occurrence: 07/28/2009 02:17:23 PM
Rem. Sec Obj.:
Port: 0
Suppr./Esc1.: Normal
Rem. Root Obj.:
Extended Attributes:
Slot: 0
Local Node Alias:
Managed Status: Managed
Alert Group: EVENT
Class: TME10tecad
Flash: No
URL:
Summary: hello_eif_probe
Cause Type: Unknown
Count: 1
Poll: 0
Agent: TEC
State Change: 07/28/2009 02:17:23 PM
Alert Key: TEC
Local Sec. Obj.:
TaskList: Not in Task List
Local Root Obj.:
Group: Public
Card:
Serial: 1950
Manager: tivoli_eif_probe on austin.tivlab.raleigh.ibm.com
Event Type: Not Defined
Rem. Pri. Obj.:
Customer:
Expire Time: Not Set
Identifier: :TEC:EVENT
Prec. Obj. Inst.: 0
TECRepeatCount: 0
Correlated Notif.:
TECFQHostname:
Specific Problem:
Owner: Nobody
Location:
Server Name: NCOMS
Service:
Internal Timestamp: 07/28/2009 02:17:23 PM
Node: 9.42.19.243
TECStatus:
TECDate:
Probable Cause: Not Defined
Rem. Node Alias:
Severity: Indeterminate
Last Occurrence: 07/28/2009 02:17:23 PM
TECHostname:
Grade: 1
Server Serial: 1950
TECDateReception:
Local Pri. Obj.:
TECServerHandle:
Precision Domain:
```

Type: Problem
ÉvénementEvent ID:
Prec. Serial:
Ack: No

L'exemple suivant d'événement a été traité par le serveur Tivoli Enterprise Console :

```
Su_Failure:
server_handle=1;
date_reception=784408852;
event_handle=1;
source=LOG;
sub_source=login;
origin=oak;
sub_origin='';
hostname='';
last_modified_time='Nov 07, 1994 08:51:42';
adapter_host='';
status=OPEN;
administrator='';
acl=[admin];
severity=WARNING;
date='Nov 07, 1994 08:51:42';
duration=0;
msg='';
msg_catalog='';
msg_index=0;
num_actions=1;
credibility=0;
repeat_count=0;
cause_date_reception=0;
cause_event_handle=0;
from_user=don;
to_user=root;
END
```

Concepts associés:

«Événements», à la page 1

L'unité centrale d'informations est l'événement. Un événement est une modification de l'état d'une ressource système ou d'une application. Les événements peuvent être générés pour des problèmes ou l'achèvement de tâches avec succès.

Filtrage d'événements

Le filtrage d'événements réduit la complexité pour les opérateurs de la console et améliore les temps de réponse pour des erreurs système complexes.

Vous pouvez filtrer les événements avec Tivoli Event Integration Facility en définissant les instructions de filtrage dans le fichier de configuration.

Concepts associés:

Chapitre 5, «Filtrage des événements à la source», à la page 47

L'un des problèmes liés à la gestion des événements est l'utilisation du volume élevé d'événements que les unités peuvent générer. Vous pouvez résoudre ce problème en filtrant des événements à la source.

Règles

Le serveur d'événements utilise des règles pour indiquer et contrôler les réponses automatiques à des événements dans une entreprise. Si vous développez un nouvel adaptateur, vous pouvez écrire de nouvelles règles pour la sonde EIF et créer des automatisations.

Les automatisations du serveur ObjectServer lui permettent de traiter des événements sans l'intervention d'un opérateur. Pour plus d'informations sur les automatisations de Tivoli Netcool/OMNIbus, consultez *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Pour de plus amples informations sur les règles de Probe for Tivoli EIF, voir *IBM Tivoli Netcool/OMNIbus Probe for Tivoli EIF Reference Guide* dans le centre de documentation Network Availability Management à l'adresse <http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp>. Voir également *Guide des sondes et des passerelles d'IBM Tivoli Netcool/OMNIbus*.

Chapitre 2. Installation de l'Event Integration Facility

Tivoli Event Integration Facility n'est pas installé par le programme d'installation de l'Tivoli Netcool/OMNIBus et doit être installé séparément à partir du produit principal. Le kit de développement de logiciels EIF inclut les fichiers JAR et les bibliothèques C.

Avant de commencer

- Vérifiez que le module Microsoft Visual C++ 2005 SP1 Redistributable Package est installé sur l'hôte. Ce package est requis par les utilitaires EIF tels que **postei fmsg**. Ce package est requis car la versions de Tivoli Netcool/OMNIBus d'EIF est compilée à l'aide de Microsoft Visual C++ 2005 SP1. L'utilisateur ou l'application qui installe EIF doit installer le package redistribuable car EIF ne possède pas de programme d'installation et n'est pas installé par Tivoli Netcool/OMNIBus. Vous pouvez télécharger le package redistribuable à partir des emplacements suivants :
 - Pour les systèmes d'exploitation x86 : <http://www.microsoft.com/downloads/details.aspx?familyid=200b2fd9-ae1a-4a14-984d-389c36f85647>
 - Pour les systèmes d'exploitation x64 : <http://www.microsoft.com/downloads/details.aspx?familyid=EB4EBE2D-33C0-4A47-9DD4-B9A6D7BD44DA>
- Si vous téléchargez EIF à partir du site Web en ligne d'IBM Passport Advantage, suivez les instructions figurant dans le document de téléchargement de votre système d'exploitation.

Tableau 1. Emplacement des documents de téléchargement de tous les systèmes d'exploitation pris en charge

Système d'exploitation	Emplacement du document de téléchargement
AIX	http://www-01.ibm.com/support/docview.wss?rs=3120&uid=swg24033234
HP-UX Integrity	http://www-01.ibm.com/support/docview.wss?rs=3120&uid=swg24033294
Linux	http://www-01.ibm.com/support/docview.wss?rs=3120&uid=swg24033295
Linux for System z	http://www-01.ibm.com/support/docview.wss?rs=3120&uid=swg24033296
Solaris	http://www-01.ibm.com/support/docview.wss?rs=3120&uid=swg24033297
Windows	http://www-01.ibm.com/support/docview.wss?rs=3120&uid=swg24033298

Procédure

1. Téléchargez le groupement EIF inclus dans le module d'installation de Tivoli Netcool/OMNIBus dans un répertoire temporaire.
2. Extrayez les fichiers d'EIF compressés dans un répertoire distinct de votre répertoire d'installation d'IBM Tivoli.

Structure de répertoire

Structure de répertoire d'Event Integration Facility après l'extraction des fichiers compressés.

La table suivante décrit la structure de fichiers et de répertoire d'Event Integration Facility. Dans cette structure de répertoire, *interp* peut être l'un des noms de répertoire suivants qui sont mappés sur les systèmes d'exploitation pris en charge.

- aix4-r1
- hpux10
- hpuxia
- linux-ix86
- linux-ppc
- linux-s390
- linux-ia64
- os390
- solaris2
- solaris2-ix86
- w32-ix86

Tableau 2. Structure de répertoire Event Integration Facility

Nom de répertoire ou de fichier	En-tête
<i>bin/interp</i>	Contient, pour chaque système d'exploitation pris en charge représenté par <i>interp</i> , la version 32 bits de la commande postei fmsg utilisée pour envoyer des événements EIF.
<i>bin64/interp</i>	Contient, pour chaque système d'exploitation pris en charge représenté par <i>interp</i> , la version 64 bits de la commande postei fmsg utilisée pour envoyer des événements EIF.
<i>codeset</i>	Contient les pages de code.
<i>contrib/interp</i>	Contient les modèles de programmes C 32 bits compilés pour chaque système d'exploitation pris en charge représenté par <i>interp</i> . Les modèles de récepteurs sont <i>eifrcv1</i> , <i>eifrcv2</i> , <i>eifrcv3</i> et <i>eifrcv4</i> . Les modèles d'émetteurs sont <i>eifsend1</i> et <i>eifsend2</i> .
<i>contrib64/interp</i>	Contient les modèles de programmes C 64 bits compilés pour chaque système d'exploitation pris en charge représenté par <i>interp</i> . Les modèles de récepteurs sont <i>eifrcv1</i> , <i>eifrcv2</i> , <i>eifrcv3</i> et <i>eifrcv4</i> . Les modèles d'émetteurs sont <i>eifsend1</i> et <i>eifsend2</i> .
<i>default_sm</i>	Contient le jeu de règles vide par défaut pour le moteur de corrélation d'état Tivoli Enterprise Console State Correlation Engine. Vous pouvez ajouter des règles, lesquelles sont également appelées automates, comme requis par votre environnement.

Tableau 2. Structure de répertoire Event Integration Facility (suite)

Nom de répertoire ou de fichier	En-tête
include	Contient des fichiers d'en-tête pour générer des adaptateurs. Ces fichiers ont une extension .h. Ces fichiers sont communs dans tous les systèmes d'exploitation.
jars	Contient tous les fichiers JAR qui sont nécessaires pour utiliser l'API EIF basée sur Java. Ces fichiers sont evd.jar et log.jar.
javadoc	Contient le Javadoc pour l'API EIF basée sur Java.
lib/interp	Contient la bibliothèque statique 32 bits pour chaque système d'exploitation pris en charge représenté par <i>interp</i> . Le nom de bibliothèque est libeif.a.
lib64/interp	Contient la bibliothèque statique 64 bits pour chaque système d'exploitation pris en charge représenté par <i>interp</i> . Le nom de bibliothèque est libeif.a.
misc	Contient un exemple de fichier eif.conf que vous pouvez utiliser pour définir vos fichiers de configuration et différents fichiers readme qui expliquent comment utiliser l'EIF ainsi que la fonction et l'utilisation des modèles de fichiers qui sont fournis. En particulier, le fichier readme.tips contient des informations utiles sur l'utilisation de l'EIF.
samples	<p>Contient le code source du modèle d'adaptateur suivant :</p> <ul style="list-style-type: none"> • Modèles de récepteurs en langage C, qui sont les suivants : <ul style="list-style-type: none"> - eifrcv1.c - eifrcv2.c - eifrcv3.c - eifrcv4.c • Modèles d'émetteurs en langage C, qui sont les suivants : <ul style="list-style-type: none"> - eifsend1.c - eifsend2.c • Un modèle d'adaptateur à exécution longue en langage C, appelé sampleAdapter.c. • Un modèle d'adaptateur à exécution longue en langage Java, appelé SampleAdapter.java. <p>Conseil : Pour plus d'informations sur la fonction et l'utilisation de chaque modèle de fichier, voir le fichier misc/readme.samples.</p>
EIF.version	Fichier montrant la version actuelle d'Event Integration Facility.

Concepts associés:

«Fichiers de configuration», à la page 3

Utilisez le fichier de configuration pour contrôler le comportement des adaptateurs. Vous ne devez pas modifier plusieurs instances d'un adaptateur à exécuter dans différents environnements. Modifiez uniquement les fichiers de configuration.

Migration des adaptateurs

Si vous mettez à niveau à partir d'une version précédente d'Event Integration Facility, faites migrer tous les adaptateurs que vous générez avec la version précédente. Pour migrer, assurez-vous que les fichiers qui sont requis par la dernière version d'Event Integration Facility sont inclus dans l'adaptateur.

Les adaptateurs personnalisés créés avec une version précédente d'Event Integration Facility peuvent uniquement être migrés lorsque le type de transfert SOCKET est utilisé.

Procédure

- **C** Dans les applications que vous souhaitez migrer vers la dernière version d'Event Integration Facility, éditez le lien des fichiers binaires qui dépendent d'Event Integration Facility. Définissez un lien statique vers `libeif.a` (anciennement `libteceef.a`).
- **Java** Assurez-vous que les nouveaux fichiers JAR suivants sont inclus dans les applications que vous souhaitez migrer vers la dernière version d'Event Integration Facility :
 - `evd.jar`
 - `log.jar`

Concepts associés:

Chapitre 4, «Génération d'un adaptateur», à la page 33

Avant de générer un adaptateur, vous devez identifier les événements à surveiller. Définissez ensuite la source d'événement et les classes d'événement, puis sélectionnez la méthode de distribution d'événement. Programmez l'adaptateur d'événements, installez-le, configurez-le puis testez-le. Vous êtes ensuite prêt à exécuter l'adaptateur.

Configuration de l'utilitaire `posteifmsg` sous z/OS à l'aide d'USS

Pour configurer l'utilitaire `posteifmsg` sous z/OS à l'aide d'USS, procédez comme suit :

Procédure

1. A partir du répertoire Tivoli Netcool/OMNIBus EIFSDK, copiez le fichier `bin/os390/posteifsg` dans un emplacement dans lequel la commande est exécutée.
2. Configurez le fichier de configuration `posteifmsg` approprié. Par exemple :

```
TransportList=t1
t1_Channels=c1
c1_ServerLocation=<IP_Address_Goes_Here>
c1_Port=<EventServerListeningPortGoesHere>
t1_Type=SOCKET
```

```
BufEvtPath=/tmp/posteifmsg.cache
```

3. Dans votre environnement USS, créez le répertoire `/etc/Tivoli/codeset`. Ce répertoire contient les fichiers de jeu de codes EBCDIC extraits du répertoire Tivoli Netcool/OMNibus EIFSDK.

Conseil : Si vous avez téléchargé les fichiers dans un autre système, utilisez FTP pour transférer le fichier de jeu de codes EBCDIC ou les fichiers en mode binaire vers votre système USS et copiez-les dans le répertoire `/etc/Tivoli/codeset`.

Les noms de fichier correspondent au numéro de page de codes associé au jeu de codes. Les pages de codes sont fournies pour EBCDIC :

37, 273, 274, 277, 278, 280, 282, 284, 297, 424, 500,

870, 875, 933, 935, 937, 939, 1025, 1026, 1047, 1112, 1122, 1388

Les pages de codes correspondent aux jeux de codes EBCDIC utilisés pour les langues de l'ouest, d'Europe de l'est, du Moyen-Orient et d'Asie. La langue par défaut est l'anglais (jeu de codes 1047). L'utilitaire **postEIFMSG** utilise uniquement les fichiers dont il a besoin. C'est pourquoi vous pouvez transférer tous les fichiers de jeux de codes vers votre environnement USS.

4. Exécutez la commande suivante pour paramétrer la variable d'environnement **TISDIR** sur la valeur correcte :

```
export TISDIR=/etc/Tivoli
```

L'utilitaire **postEIFMSG** peut désormais accéder à Tivoli Management Framework (TMF), localiser le fichier de jeu de codes associé à votre environnement système, puis le convertir en UTF-8.

5. Exécutez la commande **postEIFMSG** pour envoyer des événements. Par exemple :

```
postEIFMSG -f nameOfConfigFile -m testMessage EVENT TESTING
```

Référence associée:

«**postEIFMSG**», à la page 22

L'utilitaire **postEIFMSG** vous permet de publier des événements sur l'ObjectServer via la sonde Probe for Tivoli EIF. Après avoir installé Event Integration Facility, vous pouvez copier le fichier binaire **postEIFMSG** sur d'autres hôtes.

Chapitre 3. Transport des événements

Tivoli Event Integration Facility propose plusieurs méthodes pour envoyer des informations au serveur d'événements. Le système sur lequel l'adaptateur s'exécute doit fournir une fonction de communication interprocessus basée sur le protocole TCP/IP.

Méthodes de distribution d'événement

Les événements sont distribués par un mécanisme de communication interprocessus. Pour indiquer les méthodes de distribution pour les adaptateurs et les applications à l'aide de Tivoli Event Integration Facility, modifiez le fichier de configuration et le lien vers la bibliothèque applicable. Pour distribuer des événements test directement depuis une invite de commande, utilisez l'interface de ligne de commande.

Concepts associés:

«Sélection des méthodes de distribution d'événements», à la page 35

Lors de la génération d'un adaptateur, vous devez également décider quelle méthode de distribution d'événement l'adaptateur utilise pour communiquer avec le serveur d'événements.

Référence associée:

«Test et scriptage», à la page 21

Vous pouvez utiliser l'utilitaire de ligne de commande **postei fmsg** et une classe Java pour envoyer des événements manuellement. **postei fmsg** et la classe Java sont utiles pour utiliser des scripts de shell afin de développer de nouveaux adaptateurs et tester des adaptateurs après que vous avez créé des groupes et des affectations, édité des règles ou modifié le mode de traitement des événements par un serveur. Vous pouvez également les utiliser pour identifier et résoudre les problèmes de distribution d'événement après avoir développé un nouvel adaptateur.

Options de connexion

Les options de connexion sont soit "connection-oriented" (orienté connexion), soit "connectionless" (sans connexion). Dans des situations où vous souhaitez envoyer plusieurs événements, utilisez l'option orienté connexion. Dans des situations où vous souhaitez envoyer peu d'événements au cours d'une certaine période, utilisez l'option sans connexion.

Vous pouvez écrire un seul adaptateur et l'utiliser avec une méthode de distribution orientée connexion ou sans connexion. Vous pouvez indiquer une méthode de distribution en modifiant le fichier de configuration.

- Le paramètre par défaut pour l'option de connexion est sans connexion. Cette méthode établit et annule une nouvelle connexion pour chaque événement ou groupe d'événements.
- Pour une distribution d'événement orientée connexion, indiquez `ConnectionMode=C0` dans le fichier de configuration. Cette méthode maintient le canal vers le serveur d'événements ouvert, ce qui améliore les performances lorsque vous envoyez beaucoup d'événements.

Concepts associés:

«Sélection des méthodes de distribution d'événements», à la page 35
Lors de la génération d'un adaptateur, vous devez également décider quelle méthode de distribution d'événement l'adaptateur utilise pour communiquer avec le serveur d'événements.

Référence associée:

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

Options de transfert

Une application peut utiliser l'interface de programme d'application d'Event Integration Facility pour agir en tant qu'expéditeur ou récepteur d'événements. Les options de transfert pour les connexions sont SOCKET ou Secure Sockets Layer (SSL). Elles sont définies dans le fichier de configuration d'EIF.

SOCKET

Compatible avec IPv4 et IPv6.

Effectue une liaison à la bibliothèque `libeif.a`.

Utilise le protocole TCP/IP standard pour établir des connexions.

SSL

Compatible avec IPv4 et IPv6.

Effectue une liaison à la bibliothèque `libeif.a`.

Etablit la liaison SSL standard en interne.

Peut s'exécuter en mode FIPS 140-2.

Exige des fichiers de clés et un fichier de clés certifiées SSL de l'application. Les fichiers de clés et le fichier de clés certifiées contiennent les certificats numériques et les clés nécessaires pour établir une connexion SSL. Utilisez l'utilitaire de ligne de commande `nc_gskcmd` ou l'utilitaire graphique iKeyman qui sont fournis avec GSKit pour créer les fichiers de clés et le fichier de clés certifiées. GSKit est accessible à partir de `$NCHOME/bin` sur les systèmes d'exploitation UNIX ou à partir de `%NCHOME%\bin` sur les systèmes d'exploitation Windows.

Concepts associés:

«Sélection des méthodes de distribution d'événements», à la page 35

Lors de la génération d'un adaptateur, vous devez également décider quelle méthode de distribution d'événement l'adaptateur utilise pour communiquer avec le serveur d'événements.

Référence associée:

«Liaison de l'adaptateur généré avec l'interface de programme d'application (API C)», à la page 44

Le tableau suivant répertorie les bibliothèques requises pour lier des adaptateurs développés avec l'API C.

«Support SSL et FIPS 140-2», à la page 17

Tivoli Event Integration Facility prend en charge l'utilisation du protocole de chiffrement et d'authentification SSL (Secure Sockets Layer) pour envoyer et recevoir des événements. En outre, les connexions SSL d'EIF peuvent fonctionner en mode FIPS 140-2, ce qui implique l'utilisation de fournisseurs cryptographiques approuvés par la norme FIPS 140-2.

Support SSL et FIPS 140-2

Tivoli Event Integration Facility prend en charge l'utilisation du protocole de chiffrement et d'authentification SSL (Secure Sockets Layer) pour envoyer et recevoir des événements. En outre, les connexions SSL d'EIF peuvent fonctionner en mode FIPS 140-2, ce qui implique l'utilisation de fournisseurs cryptographiques approuvés par la norme FIPS 140-2.

SSL utilise des certificats numériques pour l'échange et l'authentification de clés. Lorsqu'un client initie une connexion SSL, le serveur présente au client un certificat signé par une autorité de certification. Une autorité de certification est une partie certifiée qui garantit l'identité du certificat et de son créateur. Le certificat serveur contient l'identité du serveur, la clé publique et la signature numérique de l'émetteur du certificat.

En mode FIPS 140-2, toutes les fonctions de chiffrement et de génération de clés nécessaires pour les connexions SSL sécurisées sont fournies par les fournisseurs cryptographiques approuvés par la norme FIPS 140-2. Ces fournisseurs sont identiques à ceux utilisés pour le mode FIPS 140-2 pour le produit Tivoli Netcool/OMNIBus principal. Pour plus d'informations relatives au mode FIPS 140-2, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIBus*. Le support FIPS 140-2 est fourni comme suit pour chaque version de l'API EIF :

- **C** Le mode FIPS 140-2 est facilité par l'utilisation d'IBM Global Security Kit (GSKit) version 7. Vérifiez que le chemin d'accès aux bibliothèques GSKit s'affiche dans les variables d'environnement suivantes :

Système d'exploitation	Variables d'environnement
UNIX	LIBPATH, SHLIB_PATH ou LD_LIBRARY_PATH
Windows	PATH

GSKit est accessible à partir de \$NCHOME/bin sur les systèmes d'exploitation UNIX ou à partir de %NCHOME%\bin sur les systèmes d'exploitation Windows.

Restriction : La prise en charge de SSL est indisponible dans l'API C d'EIF pour les plateformes non prises en charge par GSKit.

- **Java** Le mode FIPS 140-2 est facilité par l'utilisation de l'environnement d'exécution IBM Java (JRE) version 1.4.2 ou supérieure. L'environnement JRE doit être configuré pour le mode FIPS 140-2.

Activation du mode FIPS 140-2

Pour activer le mode FIPS 140-2 :

1. Editez le fichier de configuration d'EIF et définissez la propriété **channel_nameSSLFIPSMODE** sur ON.
2. Pour toutes les applications basées sur Java, configurez l'environnement d'exécution Java pour le mode FIPS 140-2.
3. Si une application réceptrice EIF est activée pour le mode FIPS 140-2, activez le mode FIPS 140-2 pour tous les programmes clients EIF qui se connectent à ce récepteur.

Génération et gestion des certificats de clé et numériques

Si vous configurez SSL dans le fichier de configuration EIF, utilisez l'utilitaire de ligne de commande **nc_gskcmd** ou l'utilitaire graphique iKeyman pour générer et gérer des clés et des certificats numériques qui sont requis pour la communication SSL. Le tableau suivant décrit les clés et les certificats à gérer et les détaille dans le contexte de l'EIF.

Tableau 3. Configuration et utilisation du fichier de base de données de clés pour les opérations SSL et FIPS 140-2 de l'EIF





Fichier de base de données	Description	Instructions
Fichiers de clés des applications réceptrices EIF	<p>Le fichier de clés du récepteur est une base de données de clés. Le certificat personnel par défaut du fichier de clés est présenté par le récepteur EIF aux clients EIF lors d'une connexion SSL. Ce certificat peut être un certificat autosigné que vous créez ou un certificat obtenu de et signé par une autorité de certification. Le certificat doit être défini comme le certificat personnel par défaut dans la base de données de clés du récepteur. Les formats de base de données suivants sont utilisés :</p> <ul style="list-style-type: none"> •  CMS •  JKS 	<p>Créez des fichiers de clés pour les applications réceptrices EIF .</p> <p>Utilisez le mot clé de configuration channel_nameSSLKeystore pour configurer le récepteur EIF afin qu'il utilise une base de données de clés en tant que fichier de clés. Si une application réceptrice EIF utilise le mot clé de configuration, à savoir si le mot clé channel_nameSSLRequire ClientAuthentication est défini sur YES, les clients doivent également présenter un certificat pendant une connexion SSL. Le fichier de clés du récepteur doit contenir le certificat personnel du récepteur et les certificats personnels par défaut des clients sécurisés. Importez ces certificats sécurisés dans le fichier de clés du récepteur.</p>
Fichiers de clés des applications client EIF	<p>Le fichier de clés du client est une base de données de clés contenant les certificats personnels par défaut des applications réceptrices EIF sécurisées par le client EIF. Importez ces certificats sécurisés dans le fichier de clés du client afin que ce dernier puisse se connecter au récepteur EIF utilisant SSL. Les formats de base de données suivants sont utilisés :</p> <ul style="list-style-type: none"> •  CMS •  JKS 	<p>Créez des fichiers de clés pour les applications client EIF.</p> <p>Utilisez le mot clé de configuration channel_nameSSLKeystore pour configurer le récepteur EIF afin qu'il utilise une base de données de clés en tant que fichier de clés. Si une application réceptrice EIF utilise le mot clé de configuration, à savoir si le mot clé channel_nameSSLRequire ClientAuthentication est défini sur YES, les clients doivent également présenter un certificat pendant une connexion SSL. Le fichier de clés du récepteur doit contenir le certificat personnel du récepteur et les certificats personnels par défaut des clients sécurisés. Importez ces certificats sécurisés dans le fichier de clés du récepteur.</p>

Tableau 3. Configuration et utilisation du fichier de base de données de clés pour les opérations SSL et FIPS 140-2 de l'EIF (suite)

Fichier de base de données	Description	Instructions
Fichiers de clés certifiées	Les fichiers de clés certifiées sont des bases de données de clés contenant des certificats sécurisés par une application EIF. Une application EIF, client ou réceptrice, peut utiliser une seule base de données de clés pour conserver les certificats sécurisés et le certificat personnel par défaut de l'application.	<p>Le mot clé channel_nameSSLKeystore vous permet de configurer des fichiers de clés certifiées.</p> <p>Java Vous avez également la possibilité d'utiliser une base de données de clés pour conserver les certificats sécurisés et une seconde base de données de clés pour conserver le certificat personnel par défaut de l'application. Utilisez le mot clé channel_nameSSLTruststore pour indiquer la base de données de clés contenant les certificats sécurisés. Utilisez le mot clé channel_nameSSLKeystore pour spécifier la base de données de clés contenant le certificat personnel par défaut de l'application.</p>
Fichier de dissimulation		<p>C Pour une connexion automatique permettant d'accéder aux certificats numériques, sauvegardez le mot de passe pour une base de données de clés ou sécurisée dans un format chiffré dans un fichier de dissimulation. Lors de l'accès au fichier de clés ou au fichier de clés certifiées, le système vérifie si un fichier de dissimulation existe. S'il en détecte un, le contenu du fichier est déchiffré et utilisé comme entrée pour le mot de passe.</p> <p>Java Pour créer et stocker des mots de passe codés pour les applications, utilisez le script <code>com.tivoli.tec.event_delivery.common.Encryption</code>.</p>

Tableau 3. Configuration et utilisation du fichier de base de données de clés pour les opérations SSL et FIPS 140-2 de l'EIF (suite)

Fichier de base de données	Description	Instructions
Chiffrements	Les extrémités d'envoi et de réception de la connexion EIF doivent posséder au moins un chiffrement en commun afin que la connexion aboutisse. Le mode FIPS 140-2 limite les chiffrements autorisés pour les versions C et Java d'EIF	<p>Utilisez le mot clé SSLCipherList pour spécifier les chiffrements autorisés pendant l'authentification SSL. Si aucun chiffrement n'est indiqué ou limité, tous les chiffrements disponibles sont autorisés.</p> <p>C Les chiffrements suivants sont disponibles :</p> <ul style="list-style-type: none"> • SSL_RC2_CBC_128_CBC_WITH_MD5 • SSL_RC2_CBC_128_CBC_EXPORT40_WITH_MD5 • SSL_DES_64_CBC_WITH_MD5 • SSL_DES_192_EDE3_CBC_WITH_MD5 • SSL_NULL_WITH_NULL_NULL • SSL_RSA_WITH_NULL_MD5 • SSL_RSA_WITH_NULL_SHA • SSL_RSA_EXPORT_WITH_RC4_40_MD5 • SSL_RSA_WITH_RC4_128_MD5 • SSL_RSA_WITH_RC4_128_SHA • SSL_RSA_EXPORT_WITH_RC2_CBC_40_MD5 • SSL_RSA_WITH_DES_CBC_SHA <p>Java Tout chiffrement pris en charge par l'environnement d'exécution Java IBM est valide. Pour obtenir la liste complète des chiffrements pris en charge, voir le manuel <i>Java Secure Socket Extension (JSSE) IBMJSSE2 Provider Reference Guide for the Java 2 SDK, Standard Edition, Version 5</i> à l'adresse http://www.ibm.com/developerworks/java/jdk/security/50/secguides/jsse2Docs/JSSE2RefGuide.html.</p>

Concepts associés:

«Sélection des méthodes de distribution d'événements», à la page 35
 Lors de la génération d'un adaptateur, vous devez également décider quelle méthode de distribution d'événement l'adaptateur utilise pour communiquer avec le serveur d'événements.

Tâches associées:

«Configuration d'une application client EIF pour SSL», à la page 38
 Afin d'utiliser la communication SSL entre les applications réceptrices et client EIF, vous devez configurer l'application client EIF.

«Configuration d'une application réceptrice EIF pour SSL», à la page 36
 Afin d'utiliser la communication SSL entre les applications réceptrices et client EIF, vous devez configurer l'application réceptrice. Les récepteurs requièrent un certificat personnel autosigné ou signé par une autorité de certification.

Référence associée:

«Test et scriptage»

Vous pouvez utiliser l'utilitaire de ligne de commande **postei fmsg** et une classe Java pour envoyer des événements manuellement. **postei fmsg** et la classe Java sont utiles pour utiliser des scripts de shell afin de développer de nouveaux adaptateurs et tester des adaptateurs après que vous avez créé des groupes et des affectations, édité des règles ou modifié le mode de traitement des événements par un serveur. Vous pouvez également les utiliser pour identifier et résoudre les problèmes de distribution d'événement après avoir développé un nouvel adaptateur.

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

Test et scriptage

Vous pouvez utiliser l'utilitaire de ligne de commande **postei fmsg** et une classe Java pour envoyer des événements manuellement. **postei fmsg** et la classe Java sont utiles pour utiliser des scripts de shell afin de développer de nouveaux adaptateurs et tester des adaptateurs après que vous avez créé des groupes et des affectations, édité des règles ou modifié le mode de traitement des événements par un serveur. Vous pouvez également les utiliser pour identifier et résoudre les problèmes de distribution d'événement après avoir développé un nouvel adaptateur.

Si vous utilisez un ID utilisateur autre qu'Administrateur ou root, vérifiez que vous disposez des droits corrects pour créer le fichier. Ces droits sont spécifiés par le mot clé **BufEvtPath**.

Concepts associés:

«Méthodes de distribution d'événement», à la page 15

Les événements sont distribués par un mécanisme de communication interprocessus. Pour indiquer les méthodes de distribution pour les adaptateurs et les applications à l'aide de Tivoli Event Integration Facility, modifiez le fichier de configuration et le lien vers la bibliothèque applicable. Pour distribuer des événements test directement depuis une invite de commande, utilisez l'interface de ligne de commande.

Tâches associées:

«Programmation de l'adaptateur», à la page 40

Pour programmer un adaptateur, implémentez les interfaces et les paramètres préférés pour le fichier de configuration. Décidez de définir les valeurs d'attribut dans le fichier de configuration ou dans le code d'adaptateur. Compilez ensuite et établissez une liaison avec l'adaptateur, le cas échéant.

Référence associée:

«Support SSL et FIPS 140-2», à la page 17

Tivoli Event Integration Facility prend en charge l'utilisation du protocole de chiffrement et d'authentification SSL (Secure Sockets Layer) pour envoyer et recevoir des événements. En outre, les connexions SSL d'EIF peuvent fonctionner en mode FIPS 140-2, ce qui implique l'utilisation de fournisseurs cryptographiques approuvés par la norme FIPS 140-2.

«Installation, configuration et test de l'adaptateur», à la page 44

Après avoir assemblé tous les fichiers de l'adaptateur, vous devez installer, configurer et tester l'adaptateur avant de l'exécuter.

postei fmsg

L'utilitaire **postei fmsg** vous permet de publier des événements sur l'ObjectServer via la sonde Probe for Tivoli EIF. Après avoir installé Event Integration Facility, vous pouvez copier le fichier binaire **postei fmsg** sur d'autres hôtes.

Veillez à copier la version correcte du fichier binaire pour votre système d'exploitation et le nombre de bits.

L'utilitaire **postei fmsg** est un remplacement des utilitaires Tivoli Enterprise Console **postemsg** et **postzmsg**. L'utilitaire **postei fmsg** exécute la même fonction d'envoi d'événement pour Tivoli Netcool/OMNIbus que les utilitaires **postemsg** et **postzmsg** pour Tivoli Enterprise Console.

Syntaxe

La syntaxe de l'utilitaire **postei fmsg** est la suivante.

```
postei fmsg -S serveur [-p port] | -f fichier de configuration [-m "message"]  
[-t délai] [-l fichier_verrouillage] [-r gravité] [attribut=valeur...] classe source
```

Où :

- *serveur* est le nom d'hôte ou l'adresse IP du serveur cible. L'option de ligne de commande -S suppose que le serveur cible est un système Tivoli Enterprise Console hébergé sur un système d'exploitation UNIX et que le service d'associateur de port est utilisé. Consultez «Configuration de la sonde Probe for Tivoli EIF», à la page 23 si vous envoyez des événements à la sonde Probe for Tivoli EIF et que Tivoli Enterprise Console est sur le même hôte que la sonde.
- *fichier de configuration* est le fichier de configuration.
- *message* est le texte du message, placé entre guillemets (" "). Le nombre maximal de caractères ou d'octets pouvant être transmis avec l'option de ligne de commande -m est 4148 caractères ou octets. Si vous dépassez ce nombre maximal, un incident de segmentation et un fichier core sont générés.
- *timeout* représente le délai maximal, en secondes, pendant lequel un événement est verrouillé (envoi empêché) lorsque le fichier de verrouillage (spécifié par l'option -l) est verrouillé. Cette option permet à plusieurs instances de **postei fmsg** d'émettre des commandes simultanément, à condition que chaque instance de **postei fmsg** utilise le même fichier de verrouillage. L'option de délai d'expiration (-t) ne peut être utilisée qu'avec l'option de fichier de verrouillage (-l). Le délai par défaut est de 1000 secondes.

Une fois que le délai d'expiration de verrouillage s'est écoulé, l'événement est supprimé et le code d'erreur 1 est renvoyé. Pour éviter des événements supprimés, configurez le délai d'expiration afin de traiter le débit d'événements dans votre environnement.

- *fichier_verrouillage* représente le nom du fichier de verrouillage permettant d'activer les accès concurrents. Si le fichier de verrouillage que vous spécifiez n'existe pas déjà, il est automatiquement créé. Le fichier de verrouillage n'inclut pas de contenu et n'en exige aucun. Le fichier n'est pas supprimé ; vous devez donc le supprimer manuellement lorsqu'il n'est plus requis.
- *gravité* est une gravité valide définie pour la classe d'événement.
- *attribut=valeur* affecte une valeur à un attribut valide. La valeur doit être définie pour la classe d'événement. Séparez les expressions multiples *attribut=valeur* par un espace.
- *classe* spécifie la classe de l'événement. La classe que vous indiquez doit être une classe configurée dans le serveur. Si le nom de classe contient des espaces, placez-le entre guillemets (" ").

- *source* spécifie la source de l'événement. Si le nom de source contient des espaces, placez-le entre guillemets (" ").

Important : Vérifiez que la paire *classe source* est le dernier argument dans la chaîne de ligne de commande. Faute de quoi, la commande échoue. Pour le test, la base de toutes les paires *classe source* est EVENT EVENT.

Remarques sur la configuration

postei fmsg nécessite un fichier de configuration pour s'exécuter. Le fichier de configuration utilise des mots clés. La configuration minimale pour l'utilitaire est de définir uniquement les mots clés **ServerLocation** et **ServerPort**. Ces mots clés spécifient l'hôte et le numéro de port sur lesquels la sonde Probe for Tivoli EIF s'exécute, comme illustré dans l'exemple suivant.

```
ServerLocation=10.10.10.10
ServerPort=9998
```

Vérifiez que le mot clé **ServerPort** est défini sur le même numéro de port que le mot clé **PortNumber** dans le fichier de propriétés de la sonde Probe for Tivoli EIF, `tivoli_eif.props`.

Les autres configurations possibles sont les suivantes :

- Vérifiez que la valeur du mot clé **ServerPort** correspond à la valeur de la propriété **PortNumber** de la sonde Probe for Tivoli EIF.
- Vérifiez que le mot clé **BufferEvents** est défini sur YES, qui est la valeur par défaut.
- Laissez le mot clé **ConnectionMode** en commentaire. L'utilitaire **postei fmsg** utilise toujours la méthode `connection_less` pour la connexion et la déconnexion. Ce comportement s'applique y compris si vous définissez le mot clé **ConnectionMode** sur C0.
- Facultatif : définissez le mot clé **BufEvtPath**. La valeur par défaut est `$OMNIBUSHOME/var/postei fmsg.cache`.

Voir aussi «Exemple 1», à la page 24.

L'utilitaire **postei fmsg** n'autorise pas les unités d'exécution multiples et ne sécurise pas les caches. Il est préférable d'envoyer des nombres faibles d'événements, à des fréquences faibles. Pour envoyer de grands volumes d'événements à l'ObjectServer sur de courtes durées, vous pouvez utiliser l'utilitaire **nco_postmsg** qui se trouve dans la fonction **Probe Support** de Tivoli Netcool/OMNIBus. A la différence de Event Integration Facility, la fonction **Probe Support** de Tivoli Netcool/OMNIBus nécessite une instance du moteur Deployment Engine pour s'exécuter. Pour plus d'informations sur l'utilitaire **nco_postmsg**, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*. Pour plus d'informations sur le moteur Deployment Engine et les fonctions installables de Tivoli Netcool/OMNIBus, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIBus*.

Configuration de la sonde Probe for Tivoli EIF

Les événements provenant de **postei fmsg** doivent être envoyés à la sonde Probe for Tivoli EIF laquelle envoie les événements à l'ObjectServer. Pour faire en sorte que l'utilitaire **postei fmsg** envoie des événements à la sonde Probe for Tivoli EIF, tenez compte des informations suivantes.

- Pour exécuter l'utilitaire **postei fmsg** avec l'option de ligne de commande `-S`, vérifiez que la propriété **PortMapper** dans la sonde est définie sur TRUE.

- Si la sonde est installée sur le même hôte que Tivoli Enterprise Console, un seul de ces produits peut utiliser l'associateur de port. Par exemple, si Tivoli Enterprise Console est installé sur l'hôte et est configuré pour utiliser un associateur de port, vous ne pouvez pas exécuter la sonde sur le même hôte et paramétrer la propriété **PortMapper** sur TRUE. Le contraire s'applique également.

L'exemple suivant illustre la configuration requise dans le fichier de propriétés de la sonde, `tivoli_eif.props`, pour que l'associateur de port fonctionne.

```
Server      : 'COL_P_1'
ServerBackup : 'COL_B_1'
PortNumber  : 9998
PortMapper  : "true"
PortMapperNumber : 100033057
NetworkTimeout : 30
PollServer  : 60
AutoSAF     : 1
```

Pour plus d'informations sur la sonde Probe for Tivoli EIF, voir *IBM Tivoli Netcool/OMNIBus Probe for Tivoli EIF Reference Guide*.

Configurations des pare-feu

Vous pouvez utiliser le serveur de pont de pare-feu se trouvant dans le produit Tivoli Netcool/OMNIBus principal pour envoyer des événements via l'utilitaire **postei fmsg**. Par exemple, vous pouvez configurer des utilitaires **postei fmsg** se trouvant hors d'un pare-feu pour vous connecter à une sonde Probe for Tivoli EIF hors du pare-feu. Cette instance de la sonde Probe for Tivoli EIF se connecte à un pont de pare-feu d'accès client et non à l'ObjectServer. Le pont d'accès client hors du pare-feu se connecte à un pont de pare-feu d'accès serveur protégé par le pare-feu. Le pont de pare-feu limite l'utilisation des ports à un port unique entre le pont de pare-feu d'accès client et le pont de pare-feu d'accès serveur. Le pont de pare-feu d'accès serveur se connecte à l'ObjectServer. Pour plus d'informations sur le serveur du pont de pare-feu, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Exemple 1

L'exemple suivant illustre l'utilitaire **postei fmsg** utilisé avec un fichier de configuration appelé `postei fmsg.conf`.

```
postei fmsg -f postei fmsg.conf source=MYTEST severity=HARMLESS hostname=testhost
physical_node=testhost.us.ibm.com sub_source=Heartbeat poll_time=3600
host_type=WINDOWS msg="TEST HEARTBEST MESSAGE" origin=10.10.10.10 TEST TEMS
```

Exemple 2

L'exemple suivant présente la commande **postei fmsg** utilisée avec l'associateur de port.

```
postei fmsg -S 10.10.10.10 source=MYTEST severity=CRITICAL hostname=testhost
physical_node=testhost.us.ibm.com sub_source=postei fmsg poll_time=3600
host_type=UNIX msg="Événement de test critique" origin=10.10.10.10 TEST TEMS
```

Tâches associées:

«Configuration de l'utilitaire `postei fmsg` sous z/OS à l'aide d'USS», à la page 12
 Pour configurer l'utilitaire **postei fmsg** sous z/OS à l'aide d'USS, procédez comme suit :

Référence associée:

«Structure de répertoire», à la page 10

Structure de répertoire d'Event Integration Facility après l'extraction des fichiers compressés.

Information associée:

 Probe for Tivoli EIF

Classes Java

Utilisez l'utilitaire Java `com.tivoli.tec.event_delivery.TECAgent` pour envoyer un événement au serveur d'événements. Utilisez l'utilitaire Java `com.tivoli.tec.event_delivery.common.Encryption` pour créer une clé de chiffrement pour les communications SSL.

`com.tivoli.tec.event_delivery.TECAgent`

```
java com.tivoli.tec.event_delivery.TECAgent SENDER -S serveur | -f fichier_configuration [-m "message"] [-r gravité] [attribute=valeur...] class source
```

Where

- *serveur*
- *fichier de configuration* est le nom du fichier de configuration.
- *message* est le texte de l'événement, entre guillemets (" ").
- *gravité* est la gravité de l'événement. La gravité doit être définie pour la classe d'événement.
- *valeur* est la valeur d'un attribut valide. Vérifiez que l'attribut est défini pour la classe d'événement. Paires attribut-valeur multiples distinctes avec des espaces, par exemple `attribue1=1 attribute2=2`, etc.

Vérifiez que la classe de l'événement est configurée dans le serveur d'événements. Les classes sont définies par l'adaptateur. Si le nom de la classe contient des espaces, placez-le entre guillemets (" "). Si la source de l'événement contient des espaces, placez-la entre guillemets (" ").

L'exemple suivant envoie un message test qui affiche un événement `Su_Failure` dans la console d'événements :

```
java com.tivoli.tec.event_delivery.TECAgent SENDER -f myconfig.conf -m "su login failure." Su_Failure LOGFILE
```

`com.tivoli.tec.event_delivery.common.Encryption`

```
java -cp path_to_evd.jar com.tivoli.tec.event_delivery.common.Encryption createKey [-k chemin_fichier_clé_chiffrement] [-l longueur_clé] -o fichier_sortie -d texte_à_chiffrer [-f]
```

Les longueurs de clés valides sont 128, 192 et 256. Si vous ne spécifiez pas de longueur de clé, la valeur par défaut est 128. L'option `-f` active le mode FIPS 140-2. L'expéditeur et le récepteur EIF doivent être en mode FIPS 140-2. Pour créer un fichier de dissimulation, créez préalablement le fichier de clés de chiffrement. Lorsqu'un fichier de dissimulation est utilisé, un fichier de clés de chiffrement est obligatoire.

Java Utilisez la commande `createkey` pour générer le fichier de clés de chiffrement et la commande de chiffrement pour générer le fichier de dissimulation.

C Utilisez l'utilitaire de ligne de commande `nc_gskcmd` ou l'utilitaire graphique `iKeyman` pour créer le fichier de dissimulation.

L'exemple suivant montre la syntaxe de `com.tivoli.tec.event_delivery.common.Encryption` pour créer une clé de chiffrement.

```
java -cp ./evd.jar com.tivoli.tec.event_delivery.common.Encryption
createkey -l 128 -o ./mykey
```

La syntaxe de `com.tivoli.tec.event_delivery.common.Encryption` pour utiliser la clé de chiffrement afin de chiffrer le mot de passe et d'enregistrer les données chiffrées dans un fichier de dissimulation se présente ainsi :

```
java -cp chemin_evd.jar com.tivoli.tec.event_delivery.common.Encryption encrypt
-k chemin_fichier_clés_chiffrement -o chemin_fichier_sortie -d texte_à_chiffrer [-f]
```

Exemple :

```
java -cp ./evd.jar com.tivoli.tec.event_delivery.common.Encryption encrypt
-k ./mykey -o ./mypass -d mot_de_passe
```

Réception d'événements pour les applications

Outre envoyer des événements, Tivoli Event Integration Facility permet à d'autres applications de recevoir, c'est-à-dire d'écouter des événements.

Event Integration Facility peut instancier un ou plusieurs programmes d'écoute d'événement. Chaque programme d'écoute d'événement peut comprendre un ou plusieurs canaux. Cela permet aux informations de provenir de plusieurs sources. Vous pouvez indiquer plusieurs canaux et le programme d'écoute d'événement écoute tous ces canaux.

Le mécanisme d'interrogation permet à l'application de récupérer des événements de manière synchrone à l'aide de la méthode `get` de l'interface de programme d'application (API). Le mécanisme sans interrogation permet à l'application d'enregistrer un programme d'écoute ou un rappel et de recevoir des événements de manière asynchrone.

En outre, EIF utilise un cache pour la réception des événements. Si l'application dispose d'un programme d'écoute, EIF enregistre également un programme d'écoute dans le cache. Il notifie ensuite l'application et transfère l'événement au programme d'écoute de l'application. Si l'application n'utilise pas de programme d'écoute, l'application doit demander la récupération du prochain événement.

Voici un exemple de fichier de configuration qui permet à l'application de recevoir des événements à l'aide de sockets :

```
BufferEvents=YES
BufEvtPath=/tmp/eif_socket_recv.cache
TransportList=t1
t1Type=SOCKET
t1Channels=t_
t_ServerLocation=myserver.com
t_Port=5151
```

Voici un exemple de fichier de configuration qui permet à une application créée à l'aide de l'API C de recevoir des événements à l'aide du protocole SSL, avec le mode FIPS 140-2 activé :

```
TransportList=t1_
t1_Type=SSL
t1_Channels=c1_
c1_Port=3443

c1_ServerLocation=myserver.com
```

```
c1_SSLKeystore=/my/location/gbkeys.kdb
c1_SSLKeystorePW=password
c1_SSLEncipherList=SSL_RSA_WITH_3DES_EDE_CBC_SHA
```

```
c1_SSLEnforceMode=ON
```

Tâches associées:

«Activation du cache», à la page 28

Par défaut, le cache qui garantit la récupération après des échecs système stocke des événements. Les mots clés du fichier de configuration vous permettent de contrôler la configuration du cache.

Référence associée:

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

Envoi d'événements via des pare-feux

Vous pouvez envoyer des événements via des pare-feux en fonction des restrictions liées à votre environnement et à votre organisation concernant la sécurité par pare-feu.

Pour envoyer des événements via des pare-feux, configurez-les pour autoriser les connexions TCP/IP arbitraires sur le port indiqué pour votre adaptateur. Assurez-vous que le pare-feu autorise les communications entrantes à partir du système informatique hébergeant l'adaptateur.

Référence associée:

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

Distribution d'événements lors d'échecs système

Pour vous assurer que les événements sont distribués après des échecs système, Tivoli Event Integration Facility fournit un cache sur l'adaptateur.

Ce référentiel stocke des événements. Les événements sont supprimés du cache lorsqu'ils sont distribués. Event Integration Facility permet également de ne pas distribuer le même événement plusieurs fois.

Pour configurer votre environnement pour assurer une distribution fiable des événements, activez le cache et indiquez des serveurs de sauvegarde pour distribuer les événements. Vous pouvez aussi éviter les échecs de distribution en indiquant une liste de serveurs.

Tâches associées:

«Activation du cache», à la page 28

Par défaut, le cache qui garantit la récupération après des échecs système stocke des événements. Les mots clés du fichier de configuration vous permettent de contrôler la configuration du cache.

«Configuration des serveurs de sauvegarde pour la distribution d'événement», à la page 29

La configuration utilise les sockets TCP pour distribuer des événements. Indiquez comment contacter les serveurs de sauvegarde en définissant le canal de transfert comme étant le nom de serveur et le numéro de port.

Référence associée:

«Installation, configuration et test de l'adaptateur», à la page 44
Après avoir assemblé tous les fichiers de l'adaptateur, vous devez installer, configurer et tester l'adaptateur avant de l'exécuter.

Activation du cache

Par défaut, le cache qui garantit la récupération après des échecs système stocke des événements. Les mots clés du fichier de configuration vous permettent de contrôler la configuration du cache.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser les mots clés suivants dans le cache .

Tableau 4. Mots clés servant à la configuration du cache

Configuration du cache	Mots clés
Activation du cache	BufferEvents
Débit d'envoi des événements	BufferFlushRate MaxPacketSize
Taille du cache	BufEvtMaxSize

Vous pouvez configurer le cache de plusieurs manières différentes :

- Stockez des événements en mémoire uniquement avec le mot clé BufferEvents.
- Sauvegardez ces événements mis en mémoire tampon dans un fichier permanent en définissant le mot clé BufferEvents sur YES. Le mot clé BufEvtPath indique l'emplacement du fichier permanent.

La deuxième configuration garantit qu'aucun événement n'est perdu durant un échec système.

L'exemple suivant est tiré d'un fichier de configuration et explique l'utilisation des mots clés du cache :

```
BufferEvents=YES  
BufEvtPath=./buffer  
MaxPacketSize=130  
BufferFlushRate=96  
ConnectionMode=C0
```

Lorsque les connexions au serveur d'événements échouent, les événements attendent dans le cache. Event Integration Facility tente de se reconnecter à chaque serveur de sauvegarde l'un après l'autre. Si aucun serveur n'est disponible après toutes les nouvelles tentatives, l'interface de programme d'application (API) met en cache les événements jusqu'à établissement d'une connexion.

Lors de l'activation des fonctions de récupération, il est important de déterminer l'importance des performances et de la fiabilité. Dans certains environnements, l'utilisation du cache peut dégrader les performances en fonction de sa configuration. Par conséquent, vous pouvez ignorer la mise en cache des événements en définissant le mot clé BufferEvents sur NO.

Concepts associés:

«Distribution d'événements lors d'échecs système», à la page 27
Pour vous assurer que les événements sont distribués après des échecs système, Tivoli Event Integration Facility fournit un cache sur l'adaptateur.

«Performances et disponibilité», à la page 53
Tivoli Event Integration Facility peut contrôler les performances et la disponibilité du traitement d'événements.

Tâches associées:

«Configuration des serveurs de sauvegarde pour la distribution d'événement»
La configuration utilise les sockets TCP pour distribuer des événements. Indiquez comment contacter les serveurs de sauvegarde en définissant le canal de transfert comme étant le nom de serveur et le numéro de port.

«Filtrage des événements lors d'échecs des systèmes», à la page 48
Lorsqu'un adaptateur ne parvient pas à se connecter au serveur d'événements, il envoie les événements à un fichier si le mot clé BufferEvents est défini sur YES.

Référence associée:

«Réception d'événements pour les applications», à la page 26
Outre envoyer des événements, Tivoli Event Integration Facility permet à d'autres applications de recevoir, c'est-à-dire d'écouter des événements.

Configuration des serveurs de sauvegarde pour la distribution d'événement

La configuration utilise les sockets TCP pour distribuer des événements. Indiquez comment contacter les serveurs de sauvegarde en définissant le canal de transfert comme étant le nom de serveur et le numéro de port.

Pourquoi et quand exécuter cette tâche

Indiquez comment contacter les serveurs de sauvegarde en procédant comme suit :

Procédure

1. Créez des noms pour les connexions à établir avec les serveurs de sauvegarde. Le mot clé TransportList vous permet de créer cette liste.
2. Définissez une connexion SSL ou SOCKET pour chaque élément de cette liste à l'aide du mot clé Type.
3. Facultatif : Indiquez plusieurs canaux comme chemins alternatifs pour chaque type de transfert, à l'aide du mot clé ServerLocation.

Exemple

Voici un exemple de sauvegarde pour l'API C :

```
TransportList=t1,t2
t1Type=SSL
t1Channels=c1,c2
c1ServerLocation=sslhost1
c1Port=1111

c1_SSLKeystore=/my/location/gbkeys.kdb
c1_SSLKeystorePW=password
c1_SSLKeystoreStashFile=/my/location/gbkeys.sth
c1_SSLEncipherList=SSL_RSA_WITH_3DES_EDE_CBC_SHA

c1_SSLEncipherMode=OFF

c2ServerLocation=sslhost2
c2Port=2222

c2_SSLKeystore=/my/other/location/gbkeys.kdb
c2_SSLKeystorePW=password
c2_SSLEncipherList=SSL_RSA_WITH_3DES_EDE_CBC_SHA
```

```
c2_SSLFIPMode=OFF  
  
t2Type=SOCKET  
t2Channels=c3,c4  
c3ServerLocation=host1  
c3Port=1234  
c4ServerLocation=host2  
c4Port=5678
```

Concepts associés:

«Distribution d'événements lors d'échecs système», à la page 27
Pour vous assurer que les événements sont distribués après des échecs système, Tivoli Event Integration Facility fournit un cache sur l'adaptateur.

«Performances et disponibilité», à la page 53
Tivoli Event Integration Facility peut contrôler les performances et la disponibilité du traitement d'événements.

Tâches associées:

«Activation du cache», à la page 28
Par défaut, le cache qui garantit la récupération après des échecs système stocke des événements. Les mots clés du fichier de configuration vous permettent de contrôler la configuration du cache.

Utilisation des mots clés de l'associateur de port

Les mots clés du canal associateur de port permettent aux applications réceptrices d'enregistrer plusieurs ports sous différents noms de programme associateur de port et aux applications expéditrices d'accéder à ces noms enregistrés.

Remarque : Event Integration Facility ne prend pas en charge les mots clés de l'associateur de port pour la bibliothèque OS/390 à 64 bits.

Lorsque le mots clé `channel_namePortMapper` est défini sur YES, cela force le port indiqué à s'enregistrer auprès de l'associateur de port pour une application réceptrice. Par conséquent, le port indiqué est ignoré en faveur de l'associateur de port afin d'obtenir le port correct pour le code expéditeur. L'interface de programme d'application ignore toute autre valeur pour `channel_namePortMapper`.

Les valeurs par défaut des mots clés `channel_namePortMapper`, `channel_namePortMapperNumber` et `channel_namePortMapperVersion` sont les seules utilisées par le serveur d'événements. Si ces mots clés ne sont pas présents dans le fichier de configuration pour l'application expéditrice et si l'associateur de port est nécessaire, une tentative de connexion au serveur d'événements est exécutée. Dans une application réceptrice, le port est uniquement enregistré s'il est défini sur zéro (0). Dans ce cas, les valeurs par défaut sont utilisées.

Si un port est défini sur zéro (0), EIF utilise les mots clés `channel_namePortMapper`, `channel_namePortMapperNumber` et `channel_namePortMapperVersion`. Si l'une de leurs valeurs n'est pas définie, les valeurs par défaut sont utilisées.

Si un port est défini sur une valeur supérieure à zéro, l'associateur de port est uniquement utilisé si le mot clé `channel_namePortMapper` est défini sur YES. Dans ce cas, le port indiqué est ignoré du côté expéditeur, mais utilisé du côté récepteur. Le mot clé `channel_namePortMapper` permet d'indiquer le port utilisé pour l'associateur de port. Par conséquent, un port défini sur zéro (0) utilisera n'importe quel port disponible. Du côté récepteur, il est avantageux d'utiliser l'associateur de port afin que les application expéditrices puissent se connecter au récepteur au

moyen de l'associateur de port.

Configuration d'une application réceptrice générée avec l'interface de programme d'application (API) C

Une application réceptrice générée avec l'API C peut être configurée avec les mots clés du fichier de configuration.

Les mots clés suivants du fichier de configuration vous permettent de configurer une application de réception générée avec l'API C.

ActiveConnections=nn

Nombre de connexions actives que le processus de réception doit traiter.

Le nombre de connexions potentielles est compris entre 2 et 10 000.

Un nombre inférieur à la valeur minimale définit la valeur sur la valeur minimale, à moins que ce nombre soit égal à zéro, ce qui signifie un nombre de connexions illimité. Un nombre supérieur à la valeur maximale définit la valeur sur la valeur maximale. Si vous ne définissez aucune valeur, la valeur par défaut de 128 est utilisée.

S'applique uniquement à l'API C.

ActiveConnectionsSafety=nn

Le pourcentage ActiveConnections que le nombre de connexions réelles autorise doit être réduit à un nombre spécifique avant que les connexions puissent à nouveau être traitées. Il s'agit de la valeur de seuil.

Si vous définissez la valeur ActiveConnections, le nombre de connexions actives est limité par le processus de réception C d'Event Integration Facility.

Par exemple, si ActiveConnections équivaut à 20 et ActiveConnectionsSafety, à 80, le processus de réception arrête d'accepter des connexions lorsqu'il existe déjà 20 connexions. Le pourcentage peut aller de 10 à 90. Les nouvelles connexions reprennent lorsque le nombre de connexions actives est réduit à 16 (80 % de 20) ou moins.

La valeur par défaut est de 80 et est uniquement utilisée lorsque la valeur ActiveConnections a été définie.

Un nombre inférieur à la valeur minimale définit la valeur sur sa valeur minimale. Un nombre supérieur à la valeur maximale définit la valeur sur sa valeur maximale.

S'applique uniquement à l'API C.

ConnectionsQueued=nn

Nombre approximatif de connexions mises en file d'attente que le processus de réception traite.

Le socket de serveur met en file d'attente les connexions qui attendent d'être acceptées par le processus de réception. Les tentatives de connexion échouent une fois cette limite dépassée.

Cette option vous permet de limiter le nombre de connexions pouvant être mises en file d'attente. Le nombre réel de connexions mises en file d'attente peut être légèrement inférieur ou supérieur à la valeur de ConnectionsQueued. La valeur par défaut est 1000. La plage de valeurs peut être comprise entre 1 et 1000. Un nombre inférieur à la valeur minimale définit la valeur sur sa valeur minimale. Un nombre supérieur à la valeur maximale définit la valeur sur sa valeur maximale.

S'applique uniquement à l'API C.

Exemple d'application réceptrice générée avec l'API C et recevant des événements via une connexion SOCKET :

```
BufferEvents=YES  
BufEvtPath=/tmp/eif_socket_recv.cache
```

```
TransportList=t1  
t1Type=SOCKET  
t1Channels=t_  
t_ServerLocation=myserver.com  
t_Port=5151
```

Exemple d'application réceptrice générée avec l'API C et recevant des événements via une connexion SSL :

```
BufferEvents=YES  
BufEvtPath=/tmp/eif_socket_recv.cache
```

```
TransportList=t1  
t1Type=SSL  
t1Channels=t_  
t_ServerLocation=myserver.com  
t_Port=5151
```

```
t_SSLKeystore=/my/other/location/gbkeys.kdb  
t_SSLKeystorePW=password  
t_SSLEncipherList=SSL_RSA_WITH_3DES_EDE_CBC_SHA  
t_SSLFIPSMode=OFF
```

Remarque : La procédure de configuration d'un fichier de clés pour une application réceptrice est identique à celle d'une application expéditrice.

Chapitre 4. Génération d'un adaptateur

Avant de générer un adaptateur, vous devez identifier les événements à surveiller. Définissez ensuite la source d'événement et les classes d'événement, puis sélectionnez la méthode de distribution d'événement. Programmez l'adaptateur d'événements, installez-le, configurez-le puis testez-le. Vous êtes ensuite prêt à exécuter l'adaptateur.

Tâches associées:

«Migration des adaptateurs», à la page 12

Si vous mettez à niveau à partir d'une version précédente d'Event Integration Facility, faites migrer tous les adaptateurs que vous générez avec la version précédente. Pour migrer, assurez-vous que les fichiers qui sont requis par la dernière version d'Event Integration Facility sont inclus dans l'adaptateur.

Fichiers d'adaptateur

Outre le fichier d'en-tête ou le package Java, un certain nombre de fichiers sont connexes à l'adaptateur. Cette section répertorie et montre la relation entre ces fichiers et le traitement d'événement.

Il existe plusieurs fichiers d'adaptateur. Outre le fichier d'en-tête ou le package Java, les fichiers suivants sont connexes à l'adaptateur :

fichier .conf

Le fichier de configuration contrôle le filtrage et la mise en mémoire tampon des événements. Il contrôle également les communications.

fichier .rls

Le fichier de règles applique des règles personnalisées aux événements pour le filtrage, les tâches et d'autres actions. Certains fichiers de règles sont installés sur le serveur d'événements par défaut. Vous pouvez également indiquer d'autres règles.

Pour obtenir une description détaillée des fichiers d'adaptateur, voir le manuel *IBM Tivoli Enterprise Console Adapters Guide* dans le centre de documentation IBM Tivoli à l'adresse :

<http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp>

Référence associée:

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

Identification des événements à surveiller

Avant de créer un adaptateur, vous devez décider des types d'événements à surveiller.

Consultez les facteurs suivants pour vous aider à identifier des événements significatifs :

- Utilisateurs de ressources informatiques
- Contrat de service
- Exigences réseau pour les temps d'indisponibilité et de disponibilité
- Dépendances des applications et du réseau (par exemple, les bases de données, les sites Web d'e-commerce, les réseaux étendus, etc.)
- Exigences de performances
- Serveurs et ressources du réseau importants

Après avoir créé une liste d'événements significatifs, utilisez-la pour définir la source.

Définition de la source

La méthode utilisée pour récupérer les données sur l'événement dépend des ressources.

Lorsque vous développez un nouvel adaptateur, vous devez déterminer comment recueillir des informations sur la ressource surveillée. Vous devez également déterminer une méthode pour identifier les informations à envoyer au serveur d'événements. Par exemple, les informations nécessaires sur une ressource peuvent être recueillies à partir d'un fichier journal système. Ces informations doivent ensuite être formatées et, en option, filtrées, avant d'être envoyées au serveur d'événements.

Définitions de classes d'événement

Une tâche importante lors de la création d'un adaptateur est de déterminer les classes d'événement pour les informations à surveiller. Pour vous aider à écrire des règles pour gérer les événements, vous devez rendre les définitions d'événement aussi spécifiques que possible.

Probe for Tivoli EIF dispose d'un jeu de règles qui mappe des attributs d'événements d'Event Integration Facility à des colonnes de la table alerts.status. Vous devez définir ces attributs communs afin que la sonde puisse les traiter.

Remarque : Les noms de classe d'événement doivent être uniques.

La chaîne d'événement dépend de l'adaptateur. Vous pouvez posséder autant de paires obligatoires d'attributs et de valeurs. Le fragment de code suivant illustre l'assemblage d'une chaîne d'événement :

```
"MY_EVENT_CLASS;  
source=_ANY_DEFINED_SOURCE;  
application=myApp11;  
origin=9.179.1.234;  
msg=Hello World;  
END\n\001"
```

Dans le fragment de code, source et application sont spécifiques à l'application.

Pour plus d'informations sur les règles et les attributs de sonde, reportez-vous au guide IBM Tivoli Netcool/OMNIbus Probe for Tivoli EIF dans le centre de documentation d'IBM Tivoli Network Management à l'adresse suivante : http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/topic/com.ibm.netcool_OMNIbus.doc/probes/tivoli_eif_v11/tivoli_eif_v11/wip/reference/tveifv11_intro.html

Pour plus d'informations sur la hiérarchie de classes par défaut, voir le manuel *IBM Tivoli Enterprise Console Adapters Guide* dans le centre de documentation d'IBM Tivoli à l'adresse : <http://publib.boulder.ibm.com/infocenter/tivihelp/v3r1/index.jsp>

Concepts associés:

«Classes d'événement», à la page 3

Les classes d'événement sont des classifications spécifiques aux événements. Après avoir séparé les informations en classes d'événement, les adaptateurs formatent les informations en messages qui représentent des instances spécifiques de classes d'événement. Ils envoient ensuite les informations au serveur d'événements, qui les traite, ainsi que les informations reçues des autres adaptateurs.

Sélection des méthodes de distribution d'événements

Lors de la génération d'un adaptateur, vous devez également décider quelle méthode de distribution d'événement l'adaptateur utilise pour communiquer avec le serveur d'événements.

Vous devez définir les types de connexion et de transfert. Pour ce faire, éditez les mots clés appropriés dans le fichier de configuration d'EIF. Les types de connexion et les types de transfert suivants sont disponibles :

Options de connexion

Connection-oriented

Connectionless

Options de transfert

SOCKET

SSL

- Mode SSL normal (non FIPS)
- mode FIPS 140-2

Si vous choisissez d'utiliser le type de transfert SSL, vous devez générer des certificats de sécurité et des clés. Vous devez également décider de faire fonctionner ou non la connexion SSL en mode FIPS 140-2. Si oui, activez le mode FIPS 140-2 en éditant le fichier de configuration.

Concepts associés:

«Méthodes de distribution d'événement», à la page 15

Les événements sont distribués par un mécanisme de communication interprocessus. Pour indiquer les méthodes de distribution pour les adaptateurs et les applications à l'aide de Tivoli Event Integration Facility, modifiez le fichier de configuration et le lien vers la bibliothèque applicable. Pour distribuer des événements test directement depuis une invite de commande, utilisez l'interface de ligne de commande.

«Options de connexion», à la page 15

Les options de connexion sont soit "connection-oriented" (orienté connexion), soit "connectionless" (sans connexion). Dans des situations où vous souhaitez envoyer plusieurs événements, utilisez l'option orienté connexion. Dans des situations où vous souhaitez envoyer peu d'événements au cours d'une certaine période, utilisez l'option sans connexion.

Référence associée:

«Options de transfert», à la page 16

Une application peut utiliser l'interface de programme d'application d'Event Integration Facility pour agir en tant qu'expéditeur ou récepteur d'événements. Les options de transfert pour les connexions sont SOCKET ou Secure Sockets Layer (SSL). Elles sont définies dans le fichier de configuration d'EIF.

«Support SSL et FIPS 140-2», à la page 17

Tivoli Event Integration Facility prend en charge l'utilisation du protocole de chiffrement et d'authentification SSL (Secure Sockets Layer) pour envoyer et recevoir des événements. En outre, les connexions SSL d'EIF peuvent fonctionner en mode FIPS 140-2, ce qui implique l'utilisation de fournisseurs cryptographiques approuvés par la norme FIPS 140-2.

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

Configuration d'une application réceptrice EIF pour SSL

Afin d'utiliser la communication SSL entre les applications réceptrices et client EIF, vous devez configurer l'application réceptrice. Les récepteurs requièrent un certificat personnel autosigné ou signé par une autorité de certification.

Avant de commencer

Pourquoi et quand exécuter cette tâche

Pour configurer une application réceptrice EIF pour SSL, procédez comme suit :

Procédure

1. Créez une base de données de clés pour le récepteur EIF :
 - a. Démarrez l'utilitaire **nc_ikeyman** et créez un nouveau fichier base de données de clés.

Pour les applications EIF basées sur C, utilisez le type de base de données de clés CMS.

Pour les applications EIF basées sur Java, utilisez le type de base de données de clés JKS.
 - b. Sélectionnez un mot de passe pour la base de données de clés.

Pour les bases de données de clés CMS, vous pouvez coder le mot de passe dans un fichier de dissimulation.
2. Obtenez un certificat personnel de l'une des manières suivantes :
 - Utilisez l'utilitaire **nc_ikeyman** pour créer un nouveau certificat autosigné.
 - Utilisez l'utilitaire **nc_ikeyman** pour créer une demande de certificat et la soumettre à une autorité de certification à des fins de signature.
 - Obtenez un certificat signé auprès d'une autorité de certification à l'aide d'une autre méthode.
3. Ajoutez le nouveau certificat personnel à la base de données de clés de récepteurs et configurez-le afin qu'il soit le certificat par défaut.

4. Créez un fichier de configuration pour le récepteur EIF. Commencez par le code suivant :

```
TransportList=t1_  
t1_Type=SSL  
t1_Channels=c1_  
c1_Port=[Port d'écoute du récepteur EIF]  
c1_ServerLocation=[Nom d'hôte ou adresse IP du récepteur EIF]  
c1_SSLKeystore=[Emplacement du fichier base de données de clés de récepteurs]
```

5. Ajoutez le mot de passe de la base de données de clés de récepteurs au fichier de configuration d'EIF.
- Pour indiquer le mot de passe en texte en clair, ajoutez la ligne suivante au fichier de configuration d'EIF :
`c1_SSLKeystorePW=mot de passe de la base de données de clés de récepteurs`
 - Pour chiffrer le mot de passe à l'aide d'une base de données de clés CMS, vous pouvez utiliser l'emplacement du fichier de dissimulation associé à la base de données de clés. Ajoutez la ligne suivante au fichier de configuration d'EIF :
`c1_SSLKeystoreStashFile=emplacement du fichier de dissimulation`
 - Pour chiffrer le mot de passe lors de l'utilisation d'une base de données de clés JKS, vous pouvez faire appel à l'utilitaire EIF version Java, `com.tivoli.tec.event_delivery.common.Encryption`, pour créer un mot de passe chiffré et une clé de chiffrement qui décode le mot de passe. Ajoutez les lignes suivantes au fichier de configuration d'EIF :
`c1_SSLKeystoreStashFile=fichier contenant le mot de passe chiffré`
`c1_SSLKeystoreEncryptionKeyFile=fichier contenant la clé de chiffrement`
6. Facultatif : Sélectionnez les chiffrements que le récepteur EIF prend en charge. Par défaut, une application EIF basée sur C prend en charge tous les chiffrements disponibles dans IBM GSKit et une application EIF basée sur Java prend en charge tous les chiffrements disponibles dans IBM JRE. Ajoutez la liste de noms de chiffrement, séparés par des virgules, au fichier de configuration d'EIF, au format suivant :
`c1_SSLEncipherList=nom de chiffrement 1,nom de chiffrement 2, [...]`
7. Facultatif : Activez le mode FIPS 140-2. Par défaut, le mode FIPS 140-2 est désactivé. Pour activer le mode FIPS 140-2, ajoutez la ligne suivante au fichier de configuration d'EIF : `c1_SSLFIPSMODE=ON` Pour les applications EIF basées sur Java, l'environnement d'exécution Java doit également être configuré pour utiliser les fournisseurs FIPS. Ajoutez les entrées suivantes à la liste de fournisseurs de sécurité dans le fichier `lib/java/java.security` de votre installation de l'environnement d'exécution Java :
- ```
security.provider.numéro=com.ibm.fips.jsse.IBMJSSEFIPSProvider
security.provider.numéro=com.ibm.crypto.fips.provider.IBMJCEFIPS
```
- où *numéro* indique la séquence à allouer à chaque fournisseur de la liste.
8. Facultatif : Indiquez si les clients doivent présenter un certificat sécurisé. Par défaut, EIF ne demande pas aux clients de présenter un certificat sécurisé. Pour ajouter un certificat sécurisé, ajoutez la ligne suivante au fichier de configuration d'EIF :  
`c1_SSLRequireClientAuthentication=YES`
9. Pour les applications EIF basées sur C, mettez à jour la variable d'environnement requise avec le chemin d'accès aux fichiers de bibliothèque GSKit.

| Système d'exploitation | Variables d'environnement              |
|------------------------|----------------------------------------|
| UNIX                   | LIBPATH, SHLIB_PATH ou LD_LIBRARY_PATH |
| Windows                | PATH                                   |

GSKit est accessible à partir de \$NCHOME/bin sur les systèmes d'exploitation UNIX ou à partir de %NCHOME%\bin sur les systèmes d'exploitation Windows.

## Que faire ensuite

Vous devez ensuite configurer l'application client EIF pour SSL.

### Référence associée:

«Support SSL et FIPS 140-2», à la page 17

Tivoli Event Integration Facility prend en charge l'utilisation du protocole de chiffrement et d'authentification SSL (Secure Sockets Layer) pour envoyer et recevoir des événements. En outre, les connexions SSL d'EIF peuvent fonctionner en mode FIPS 140-2, ce qui implique l'utilisation de fournisseurs cryptographiques approuvés par la norme FIPS 140-2.

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

## Configuration d'une application client EIF pour SSL

Afin d'utiliser la communication SSL entre les applications réceptrices et client EIF, vous devez configurer l'application client EIF.

### Avant de commencer

Vous devez créer le certificat personnel du récepteur EIF avant de pouvoir configurer l'application client.

### Pourquoi et quand exécuter cette tâche

Pour configurer une application client EIF pour SSL, procédez comme suit :

### Procédure

1. Créez une base de données de clés pour le client EIF :
  - a. Démarrez l'utilitaire **nc\_ikeyman** et créez un nouveau fichier base de données de clés.  
Pour les applications EIF basées sur C, utilisez le type de base de données de clés CMS.  
Pour les applications EIF basées sur Java, utilisez le type de base de données de clés JKS.
  - b. Sélectionnez un mot de passe pour la base de données de clés.  
Pour les bases de données de clés CMS, vous pouvez chiffrer le mot de passe dans un fichier de dissimulation.
2. Ajoutez le certificat personnel du récepteur EIF à la base de données de clés de clients.
  - a. Utilisez l'utilitaire **nc\_ikeyman** pour exporter le certificat personnel par défaut depuis la base de données de clés de récepteurs.
  - b. Utilisez l'utilitaire **nc\_ikeyman** pour importer le certificat dans la base de données de clés de clients.

3. Créez un fichier de configuration pour le client EIF. Commencez par les opérations suivantes :

```
TransportList=t1_
t1_Type=SSL
t1_Channels=c1_
c1_Port=[Port d'écoute du récepteur EIF]
c1_ServerLocation=[Nom d'hôte ou adresse IP du récepteur EIF]
c1_SSLKeystore=[Emplacement du fichier base de données de clés de récepteurs]
```

4. Ajoutez le mot de passe de la base de données de clés de clients au fichier de configuration d'EIF.
  - Pour indiquer le mot de passe en texte en clair, ajoutez la ligne suivante au fichier de configuration d'EIF :  
`c1_SSLKeystorePW=mot de passe de la base de données de clés`
  - Pour chiffrer le mot de passe à l'aide d'une base de données de clés CMS, vous pouvez utiliser l'emplacement du fichier de dissimulation associé à la base de données de clés. Ajoutez la ligne suivante au fichier de configuration d'EIF :  
`c1_SSLKeystoreStashFile=emplacement du fichier de dissimulation`
  - Pour chiffrer le mot de passe lors de l'utilisation d'une base de données de clés JKS, vous pouvez faire appel à l'utilitaire EIF version Java, `com.tivoli.tec.event_delivery.common.Encryption`, pour créer un mot de passe chiffré et une clé de chiffrement qui décode le mot de passe. Ajoutez les lignes suivantes au fichier de configuration d'EIF :  
`c1_SSLKeystoreStashFile=fichier contenant le mot de passe codé`  
`c1_SSLKeystoreEncryptionKeyFile=fichier contenant la clé de chiffrement`
5. Facultatif : Indiquez les chiffrements obligatoires pour le récepteur EIF. Si le récepteur prend en charge tous les chiffrements, ignorez cette étape. Ajoutez la liste de noms de chiffrement, séparés par des virgules, au fichier de configuration d'EIF, au format suivant :  
`c1_SSLEncipherList=nom de chiffrement 1,nom de chiffrement 2, [...]`
6. Facultatif : Si le récepteur EIF est activé pour le mode FIPS 140-2, activez également le mode FIPS 140-2 pour le client. Sinon, n'activez pas le mode 140-2 pour le client.  
Par défaut, le mode FIPS 140-2 est désactivé. Pour activer le mode FIPS 140-2, ajoutez la ligne suivante au fichier de configuration d'EIF : `c1_SSLFIPSMODE=ON`  
Pour les applications EIF basées sur Java, l'environnement d'exécution Java doit également être configuré pour utiliser les fournisseurs FIPS. Ajoutez les entrées suivantes à la liste de fournisseurs de sécurité dans le fichier `lib/java/java.security` de votre installation de l'environnement d'exécution Java :  
`security.provider.numéro=com.ibm.fips.jsse.IBMJSSEFIPSProvider`  
`security.provider.numéro=com.ibm.crypto.fips.provider.IBMJCEFIPS`  
où *numéro* indique la séquence à allouer à chaque fournisseur de la liste.
7. Facultatif : Si le récepteur EIF demande aux clients de présenter un certificat sécurisé, procédez comme suit :
  - a. Obtenez un certificat sécurisé. Vous pouvez utiliser l'utilitaire `nc_ikeyman` pour créer un nouveau certificat autosigné ou l'utilitaire `nc_ikeyman` pour créer une demande de certificat et la soumettre à l'autorité de certification à des fins de signature. Sinon, vous pouvez obtenir un certificat signé auprès d'une autorité de certification à l'aide d'une autre méthode.
  - b. Ajoutez le nouveau certificat à la base de données de clés de clients et configurez-le afin qu'il soit le certificat par défaut.

- c. Ajoutez le nouveau certificat à la base de données de clés de récepteurs mais ne le configurez pas en certificat par défaut.
- 8. Pour les applications EIF basées sur C, mettez à jour la variable d'environnement requise avec le chemin d'accès aux fichiers de bibliothèque GSKit.

| Système d'exploitation | Variables d'environnement              |
|------------------------|----------------------------------------|
| UNIX                   | LIBPATH, SHLIB_PATH ou LD_LIBRARY_PATH |
| Windows                | PATH                                   |

GSKit est accessible à partir de \$NCHOME/bin sur les systèmes d'exploitation UNIX ou à partir de %NCHOME%\bin sur les systèmes d'exploitation Windows.

#### Référence associée:

«Support SSL et FIPS 140-2», à la page 17

Tivoli Event Integration Facility prend en charge l'utilisation du protocole de chiffrement et d'authentification SSL (Secure Sockets Layer) pour envoyer et recevoir des événements. En outre, les connexions SSL d'EIF peuvent fonctionner en mode FIPS 140-2, ce qui implique l'utilisation de fournisseurs cryptographiques approuvés par la norme FIPS 140-2.

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

---

## Programmation de l'adaptateur

Pour programmer un adaptateur, implémentez les interfaces et les paramètres préférés pour le fichier de configuration. Décidez de définir les valeurs d'attribut dans le fichier de configuration ou dans le code d'adaptateur. Compilez ensuite et établissez une liaison avec l'adaptateur, le cas échéant.

### Avant de commencer

#### Exemple de code d'adaptateur

Vous trouverez un exemple de code d'adaptateur pour les langages C et Java dans le répertoire EIFSDK.

#### Interfaces de programme d'application (API) de transfert de données

Les données sont transmises au serveur d'événements en assemblant une chaîne d'événement. La fonction `tec_put_event` ou la méthode `sendEvent` envoie la chaîne au serveur d'événements.

#### Compilation de l'adaptateur

Les adaptateurs peuvent être générés à l'aide de l'interface de programme d'application (API) C ou Java.

Les adaptateurs générés à l'aide d'Event Integration Facility ne sont pas thread-safe et ne peuvent pas être multiprocessus.

#### Compilation de l'adaptateur intégré à l'API Java

Pour compiler une source Java qui utilise l'API Java d'Event Integration Facility, importez `evd.jar` dans votre fichier source et vérifiez que `evd.jar` et `log.jar` sont disponibles dans le chemin d'accès aux classes de compilation.

## Pourquoi et quand exécuter cette tâche

Event Integration Facility fournit une API C et une API Java pour communiquer avec le serveur d'événements.

Tableau 5. Les API et leurs comportements

| Tâches préalables                                                                                                                                                                 | API Java                | API C                        |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|------------------------------|
| Accès aux fichiers de configuration et lecture des données de mot clé                                                                                                             | <b>TECAgent</b>         | <b>tec_agent_init</b>        |
|                                                                                                                                                                                   | <b>getConfigVal</b>     | <b>tec_agent_getenv</b>      |
| Etablir et fermer les communications avec le serveur d'événements                                                                                                                 | <b>TECAgent</b>         | <b>tec_create_handle</b>     |
|                                                                                                                                                                                   | <b>disconnect</b>       | <b>tec_destroy_handle</b>    |
| Envoyer et recevoir des événements                                                                                                                                                | <b>sendEvent</b>        | <b>tec_put_event</b>         |
|                                                                                                                                                                                   | <b>receiveEvent</b>     | <b>tec_get_event</b>         |
|                                                                                                                                                                                   | <b>registerListener</b> | <b>tec_register_callback</b> |
|                                                                                                                                                                                   | <b>onMessage</b>        | <b>tec_event_callback</b>    |
| <b>Remarque :</b> L'environnement local est défini indépendamment d'Event Integration Facility et l'appel de l'API <b>tec_create_handle</b> ne modifie pas l'environnement local. |                         |                              |

Utilisez les fonctions et méthodes pour gérer les fichiers de configuration, les communications et le transfert de données.

### Concepts associés:

Annexe A, «Interfaces de programme d'application (API)», à la page 59

Les API de Tivoli Event Integration Facility sont définies par les fonctions du langage C et les méthodes Java. Elles vous permettent de générer un adaptateur personnalisé ou une application qui reçoit des événements.

### Référence associée:

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

«Test et scriptage», à la page 21

Vous pouvez utiliser l'utilitaire de ligne de commande **postei fmsg** et une classe Java pour envoyer des événements manuellement. **postei fmsg** et la classe Java sont utiles pour utiliser des scripts de shell afin de développer de nouveaux adaptateurs et tester des adaptateurs après que vous avez créé des groupes et des affectations, édité des règles ou modifié le mode de traitement des événements par un serveur. Vous pouvez également les utiliser pour identifier et résoudre les problèmes de distribution d'événement après avoir développé un nouvel adaptateur.

## Mise à niveau d'adaptateurs existants

Pour mettre à niveau des adaptateurs existants, établissez une liaison entre vos adaptateurs et la bibliothèque `libeif.a`.

Lors de la mise à niveau d'adaptateurs existants, établissez à nouveau une liaison entre votre code d'adaptateur Event Integration Facility et la bibliothèque de module de remplacement `libeif.a`. Si vous recevez un message d'erreur `undefined symbol`, cela signifie que la fonction EIF que vous utilisez n'est plus prise en charge. Mettez à niveau, compilez et établissez à nouveau la liaison avec votre code.

### Concepts associés:

Annexe A, «Interfaces de programme d'application (API)», à la page 59  
Les API de Tivoli Event Integration Facility sont définies par les fonctions du langage C et les méthodes Java. Elles vous permettent de générer un adaptateur personnalisé ou une application qui reçoit des événements.

### Référence associée:

«Liaison de l'adaptateur généré avec l'interface de programme d'application (API C)», à la page 44

Le tableau suivant répertorie les bibliothèques requises pour lier des adaptateurs développés avec l'API C.

## Interface de programme d'application (API) des fichiers de configuration

La première tâche exécutée par les API est d'accéder aux informations des fichiers de configuration.

Pour activer la fonction `tec_agent_getenv` ou la méthode `getConfigVal`, appelez d'abord l'interface de programme d'application d'initialisation (`tec_agent_init` ou `TECAgent`). L'API d'initialisation lit les informations de configuration utilisées par toutes les fonctions ou méthodes suivantes.

### Référence associée:

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

## Interface de programme d'application (API) de communication

Les API de communication fournissent un mécanisme pour communiquer avec le serveur d'événements. Le descripteur est indiqué lors de l'envoi d'un événement.

Quel que soit le type de mécanisme de transfert, utilisez un appel unique de `tec_create_handle` (ou `TECAgent`) pour établir des communications avec le serveur. L'exemple de code suivant lance `TECAgent` et transmet le fichier de configuration, le mode de distribution et le mécanisme d'indication des erreurs en tant que paramètres :

```
public TECAgent(Reader configStream, int deliveryMode, int oneway)
```

Appelez la fonction `tec_destroy_handle` ou la méthode `disconnect` lorsque vous ne souhaitez plus communiquer avec le serveur d'événements. Cet appel est facultatif car le canal se déconnecte automatiquement lorsque l'adaptateur quitte.

## Remarques spécifiques pour Microsoft Windows

Un adaptateur créé pour les systèmes d'exploitation Microsoft Windows doit initialiser Winsock avant d'appeler `tec_create_handle`.

L'exemple suivant d'adaptateur Microsoft Windows initialise Winsock avant d'appeler `tec_create_handle` :

```
#ifdef WIN32
#include <winsock.h>
 WSADATA wsaData;

 if ((rc = WSStartup(MAKEWORD(1, 1), &wsaData)) != 0) {
 printf("error %d starting winsock.dll\n", rc);
 exit(1);
 }

 else
 printf("Winsock initialized successfully...\n");
#endif
```

Une application doit appeler la fonction `WSStartup` pour initialiser Winsock, quelle que soit la version de Winsock utilisée.

`WSStartup` a initialisé `Winsock2.dll` et une structure `WSADATA` qui contient les détails de l'implémentation de Winsock. Lorsqu'une application ou une bibliothèque de liaison dynamique a terminé d'utiliser `Winsock2.dll`, elle doit appeler `WSACleanup` pour permettre à `Ws2.dll` de libérer les ressources pour l'application. Pour chaque appel de `WSStartup`, un appel de `WSACleanup` doit exister.

Si cette opération aboutit, après le renvoi d'informations par `WSStartup`, une application ne peut pas appeler `WSAGetLastError` pour déterminer la valeur de l'erreur.

## Compilation de l'adaptateur généré avec l'API C

Pour compiler un fichier source qui utilise l'API C d'Event Integration Facility, vous devez inclure le fichier d'en-tête `tec_eef.h`.

Lors de la compilation du modèle d'adaptateur sous Windows, vous devez définir l'indicateur `PC` en tant qu'argument du programme de compilation pour éviter les erreurs d'heure de compilation.

Pour établir une liaison avec le modèle d'adaptateur, vous devez utiliser l'option `NODEFAULTLIB:LIBC.LIB` lors de la liaison de l'adaptateur sous Windows. Cela permet à l'éditeur de liens d'éviter les conflits avec les bibliothèques par défaut.

S'il existe plusieurs fichiers `wsock32.lib`, vous devez utiliser l'option `FORCE:MULTIPLE` lors de la compilation de l'adaptateur Windows. Cela force le compilateur à choisir un fichier pour éliminer les erreurs d'heure de compilation. En association avec l'option `FORCE:MULTIPLE`, l'option `INCREMENTAL:NO` doit également être utilisée.

### Exemple

L'exemple suivant montre comment compiler et lier un modèle d'adaptateur à l'aide des options mentionnées ci-dessus :

```

unsecure: adapter.c \
$ (CC) -nologo -Ze -W3 -MD -DUNSECURE -D_WIN32 -DWIN32 -DPC \
-Ic:/Tivoli/include/w32-ix86/TME/TEC -Id:/msdev/include -FosampleAdapter.obj
-c sampleAdapter.c \
slashes link -subsystem:console -L. -Ld:/msdev/lib -Lc:/Tivoli/lib/w32-ix86 \
-out:sampleAdapter.exe sampleAdapter.obj msvcrt.lib libeif.a \
libsunrpc.a -NODEFAULTLIB:LIBC.LIB -INCREMENTAL:NO -FORCE:MULTIPLE wsock32.lib

```

## Liaison de l'adaptateur généré avec l'interface de programme d'application (API C)

Le tableau suivant répertorie les bibliothèques requises pour lier des adaptateurs développés avec l'API C.

*Tableau 6. Bibliothèques pour les adaptateurs développés avec l'API C d'Event Integration Facility*

| Bibliothèques                          | Fourni par                        | Détails complémentaires                                          |
|----------------------------------------|-----------------------------------|------------------------------------------------------------------|
| libeif.a                               | Tivoli Event Integration Facility | Aucune                                                           |
| libdl.a                                | Système d'exploitation            | Pas sous Windows et HP-UX                                        |
| libpthreads.a                          | Système d'exploitation            | Pour les adaptateurs sous AIX                                    |
| libpthread.a                           | Système d'exploitation            | Pour les adaptateurs sous Linux                                  |
| libnsl.a<br>libsocket.a<br>libthread.a | Système d'exploitation            | Pour les adaptateurs sous l'environnement d'exploitation Solaris |
| libsunrpc.a                            | Tivoli Event Integration Facility | Pour les adaptateurs sous Windows                                |
| Bibliothèques C standard               | Système d'exploitation            | Aucune                                                           |

### Référence associée:

«Options de transfert», à la page 16

Une application peut utiliser l'interface de programme d'application d'Event Integration Facility pour agir en tant qu'expéditeur ou récepteur d'événements. Les options de transfert pour les connexions sont SOCKET ou Secure Sockets Layer (SSL). Elles sont définies dans le fichier de configuration d'EIF.

«Mise à niveau d'adaptateurs existants», à la page 42

Pour mettre à niveau des adaptateurs existants, établissez une liaison entre vos adaptateurs et la bibliothèque libeif.a.

---

## Installation, configuration et test de l'adaptateur

Après avoir assemblé tous les fichiers de l'adaptateur, vous devez installer, configurer et tester l'adaptateur avant de l'exécuter.

Après avoir installé l'adaptateur, vous pouvez également lui ajouter d'autres fonctions. Par exemple, vous pouvez modifier l'adaptateur pour activer les fonctions suivantes :

- Filtrage d'événements
- Mise en cache d'événements pour les échecs et la reprise

Avant d'installer votre nouvel adaptateur dans un environnement de production, testez au moins les fonctionnalités suivantes :

- L'adaptateur est-il installé avec succès ?
- Les événements sont-ils mappés aux classes d'événement appropriées ?
- Les événements arrivent-ils et s'affichent-ils sur la console, le cas échéant ?
- Les événements sont-ils correctement filtrés ?

**Concepts associés:**

«Distribution d'événements lors d'échecs système», à la page 27

Pour vous assurer que les événements sont distribués après des échecs système, Tivoli Event Integration Facility fournit un cache sur l'adaptateur.

**Référence associée:**

«Test et scriptage», à la page 21

Vous pouvez utiliser l'utilitaire de ligne de commande **postei fmsg** et une classe Java pour envoyer des événements manuellement. **postei fmsg** et la classe Java sont utiles pour utiliser des scripts de shell afin de développer de nouveaux adaptateurs et tester des adaptateurs après que vous avez créé des groupes et des affectations, édité des règles ou modifié le mode de traitement des événements par un serveur. Vous pouvez également les utiliser pour identifier et résoudre les problèmes de distribution d'événement après avoir développé un nouvel adaptateur.

---

## Exécution d'adaptateurs générés avec l'interface de programme d'application (API) Java d'Event Integration Facility

L'API Java d'Event Integration Facility dépend d'autres classes pour exécuter ses tâches. Outre la configuration de l'environnement adéquat à l'aide des commandes **setup\_env**, vous devez ajouter le chemin au fichier exécutable Java dans la variable d'environnement de chemin d'accès à la bibliothèque.

Le tableau suivant répertorie les fichiers jar et les bibliothèques Java nécessaires pour exécuter un adaptateur généré à l'aide de l'API Java d'Event Integration Facility.

Pour exécuter un adaptateur qui utilise l'API Java d'Event Integration Facility, vous devez ajouter les bibliothèques requises à votre environnement CLASSPATH :

*Tableau 7. Bibliothèques requises pour les adaptateurs développés avec l'API Event Integration Facility Java :*

| Bibliothèques obligatoires              | Fournies par                      | Détails complémentaires                                |
|-----------------------------------------|-----------------------------------|--------------------------------------------------------|
| evd.jar<br>log.jar                      | Tivoli Event Integration Facility | Aucun.                                                 |
| ibmjssefips.jar<br>ibmjsseprovider2.jar | Tivoli Event Integration Facility | Facultatif, pour la prise en charge de SSL uniquement. |

**Référence associée:**

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

---

## Configuration d'adaptateurs pour des environnements internes

Le serveur d'événements peut recevoir des événements au codage UTF-8 ou au codage de l'hôte du serveur d'événements. Le serveur d'événements détermine automatiquement le type de codage (UTF-8 ou non-UTF-8) d'un événement en évaluant un indicateur spécifique dans les données d'événement.

Pour tous les adaptateurs utilisant les types de transfert SOCKET et SSL, utilisez le codage UTF-8 pour envoyer des événements au serveur d'événements.

**Remarque :** Si l'adaptateur envoie des événements à un hôte de serveur d'événements exécutant une version de Tivoli Enterprise Console antérieure à la version 3.7, les fichiers de format situés dans les répertoires de localisation doivent rester en anglais.

**Référence associée:**

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

---

## Chapitre 5. Filtrage des événements à la source

L'un des problèmes liés à la gestion des événements est l'utilisation du volume élevé d'événements que les unités peuvent générer. Vous pouvez résoudre ce problème en filtrant des événements à la source.

Un volume élevé d'événements peut être généré, par exemple, si un routeur a dépassé un seuil de performance clé, notamment la durée nécessaire pour renvoyer une commande ping. En règle générale, le routeur est défini pour générer un événement toutes les 30 secondes jusqu'à ce qu'un opérateur détecte et adresse le problème sous-jacent. Cela provoque le débordement de la console d'événements en raison des événements redondants, ce qui a un impact sur le processus de résolution des problèmes.

Tivoli Event Integration Facility adresse ce problème grâce à deux techniques puissantes pour analyser, récapituler et distribuer les données d'événement entrantes :

### **Filtrage à l'aide des fichiers de configuration**

A l'aide des fichiers de configuration sur les adaptateurs et les passerelles, vous pouvez filtrer des événements en fonction des correspondances aux classes d'événements.

Il s'agit de l'option à la fois la plus simple et la plus limitée en termes de capacités.

Lorsqu'ils sont définis correctement, les fichiers de configuration optimisent la gestion des événements en réduisant le nombre d'événements que doit surveiller chaque opérateur.

### **Concepts associés:**

«Filtrage d'événements», à la page 6

Le filtrage d'événements réduit la complexité pour les opérateurs de la console et améliore les temps de réponse pour des erreurs système complexes.

---

## Filtrage à l'aide des fichiers de configuration

Normalement, un adaptateur envoie tous les événements au serveur d'événements. Vous pouvez facultativement répertorier les événements que l'adaptateur peut ou non envoyer au serveur d'événements à l'aide des mots clés Filter et FilterCache. De même, vous pouvez modifier le fichier de configuration pour filtrer les événements.

### **Pourquoi et quand exécuter cette tâche**

Un fichier de configuration peut contenir autant d'entrées de filtre que nécessaire. Indiquez la classe et les données d'événement, notamment l'origine, la gravité ou tout autre attribut ou paire de valeurs défini(e) pour la classe d'événement. En fonction de la valeur que vous attribuez aux mots clés Filter et FilterMode, les événements filtrés sont envoyés au serveur d'événements ou supprimés.

### **Procédure**

Définissez des filtres d'événement de la manière suivante :

- Pour envoyer tous les événements au serveur d'événements (le comportement par défaut) :
  1. Définissez FilterMode sur OUT.
  2. Ne spécifiez pas d'instruction Filter.
- Pour envoyer des événements spécifiques au serveur d'événements, procédez comme suit :
  1. Définissez FilterMode sur IN.
  2. Créez des instructions Filter pour correspondre à des événements spécifiques à envoyer.
- Pour supprimer tous les événements, procédez comme suit :
  1. Définissez FilterMode sur IN.
  2. Ne spécifiez pas d'instruction Filter.
- Pour supprimer des événements spécifiques, procédez comme suit :
  1. Définissez FilterMode sur OUT (la valeur par défaut).
  2. Créez des instructions Filter pour correspondre à des événements spécifiques à supprimer.

## Exemple

L'exemple suivant présente deux filtres. Le premier filtre supprime tous les événements possédant la classe disk\_event. Le second filtre supprime tous les événements possédant la classe Su\_Success à partir de l'adresse IP 126.32.2.14.

```
#
Filtres d'événements
#
Filter:Class=disk_event
Filter:Class=Su_Success;origin=126.32.2.14
```

### Référence associée:

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

## Filtrage des événements lors d'échecs des systèmes

Lorsqu'un adaptateur ne parvient pas à se connecter au serveur d'événements, il envoie les événements à un fichier si le mot clé BufferEvents est défini sur YES.

### Pourquoi et quand exécuter cette tâche

Vous pouvez filtrer les événements envoyés à un fichier de cache (fonctionnement identique au filtrage des événements pour le serveur d'événements), à l'aide du mot clé FilterCache.

**Remarque :** Pour empêcher la perte de données, le cache doit être suffisamment grand pour stocker tous les événements générés en cas d'indisponibilité.

Les procédures suivantes décrivent comment filtrer les événements avec les mots clés FilterCache et FilterMode lorsque le serveur d'événements est indisponible :

- Pour mettre en cache des événements spécifiques :
  1. Définissez FilterMode sur IN.
  2. Définissez BufferEvents sur YES (la valeur par défaut).
  3. Créez des instructions Filter et FilterCache pour correspondre à des événements spécifiques à envoyer.

- Pour supprimer des événements spécifiques :
  1. Définissez FilterMode sur OUT.
  2. Créez des instructions Filter et FilterCache pour correspondre à des événements spécifiques à supprimer.
- Pour mettre en cache tous les événements (le comportement par défaut) :
  1. Définissez FilterMode sur OUT.
  2. Définissez BufferEvents sur YES.
  3. Ne spécifiez pas d'instruction FilterCache.

**Remarque :** Tous les événements sont supprimés lorsque la configuration est la suivante :

1. FilterMode est défini sur IN.
2. Aucune instruction FilterCache n'est spécifiée.

**Tâches associées:**

«Activation du cache», à la page 28

Par défaut, le cache qui garantit la récupération après des échecs système stocke des événements. Les mots clés du fichier de configuration vous permettent de contrôler la configuration du cache.

## Expressions régulières dans les filtres

Vous pouvez utiliser les expressions régulières Tcl dans des instructions de filtrage.

Le format d'une expression régulière est le suivant :

re: '*fragment\_valeur*'

Une expression régulière comprend zéro branche ou plus, séparées par une barre verticale (|). Une expression régulière correspond à tout ce qui concorde avec l'une de ses branches.

Une branche comprend zéro élément ou plus concaténés. Elle correspond à une correspondance pour le premier élément, suivie d'une correspondance pour le deuxième élément, etc.

Un élément est un atome pouvant être suivi d'un astérisque (\*), d'un signe plus (+) ou d'un point d'interrogation (?). Un atome suivi d'un astérisque (\*) correspond à une séquence de zéro correspondance ou plus de l'atome. Un atome suivi d'un signe plus (+) correspond à une séquence d'une ou plusieurs correspondances de l'atome. Un atome suivi d'un point d'interrogation (?) correspond à une correspondance de l'atome ou à la chaîne null.

Un atome est une expression régulière entre parenthèses (correspondant à une correspondance de l'expression régulière), un intervalle, un point (.) (correspondant à n'importe quel caractère unique), un caret (^) (correspondant à la chaîne null au début de la chaîne d'entrée), un signe dollar (\$) (correspondant à la chaîne null à la fin de la chaîne d'entrée), une barre oblique inversée (\) suivie d'un caractère unique (correspondant à ce caractère) ou un caractère unique sans autre signification (correspondant à ce caractère).

Un intervalle est une séquence de caractères entre crochets [ ]. Un intervalle correspond à n'importe quel caractère unique de la séquence. Si la séquence commence par un caret (^), elle correspond à n'importe quel caractère unique n'appartenant pas au reste de la séquence. Si deux caractères de la séquence sont séparés par un signe moins (-), cela représente la liste complète des caractères

ASCII compris entre ces deux caractères. Par exemple, l'intervalle [0-9] correspond à n'importe quel nombre décimal. Pour inclure un crochet littéral droit (]) dans la séquence, utilisez le crochet droit (]) en tant que premier caractère, suivi d'un caret (^). Pour inclure un signe moins littéral (-), utilisez le signe moins (-) comme premier ou dernier caractère.

Un modèle de programme appelé `regtest` et un modèle de fichier en entrée appelé `regtest.data` sont fournis dans le répertoire EIFSDK pour tester les expressions régulières. Le modèle est fourni pour chaque système d'exploitation dans le répertoire `<eifsdk>/bin/$INTERP`.

**Restriction :**

1. Le code d'expression régulière est une version légèrement modifiée du code initialement écrit par Henry Spencer, Copyright 1986 by the University of Toronto. Il n'est pas dérivé d'un logiciel sous licence. Toute personne a le droit d'utiliser ce logiciel à n'importe quelle fin et sur n'importe quel système informatique et de le redistribuer librement, en respectant les restrictions suivantes :
  - a. L'auteur n'est pas responsable des conséquences de l'utilisation de ce logiciel, même si elles proviennent de défauts de ce dernier.
  - b. L'origine de ce logiciel ne doit pas être faussement représentée, soit par réclamation explicite, soit par omission.
  - c. Les versions modifiées doivent être entièrement marquées comme telles, et ne doivent pas être faussement représentées comme logiciel initial.
2. Event Integration Facility utilise une exception à la syntaxe de l'expression régulière Tcl. La barre oblique inversée (\) dans Event Integration Facility indique que le caractère littéral suivant est le caractère à filtrer, et non un caractère spécifique comme une tabulation. Par exemple, `\t` signifie le caractère de tabulation dans Tcl, mais signifie `t` dans Event Integration Facility.

L'exemple suivant présente une instruction `Filter` avec une expression régulière. Cette instruction `filter` correspond à tous les événements portant un nom de classe qui commence par `TEC_`  
`Filter:Class=re:'TEC_.*'`

L'exemple suivant présente une instruction `FilterCache` avec un intervalle plus limité. L'instruction `filter` correspond à tous les événements portant un nom de classe qui commence par `TEC_` et a une gravité critique :  
`FilterCache:Class=re:'TEC_.*';severity=CRITICAL`

Pour de plus amples informations sur les expressions régulières Tcl, voir un guide d'utilisation de Tcl.

---

## Chapitre 6. Résolution des problèmes

Vous pouvez identifier et résoudre des problèmes liés à l'installation et à l'utilisation de Tivoli Event Integration Facility. Les journaux et les utilitaires de trace peuvent vous aider à déterminer les sources de problèmes lorsqu'ils se produisent. Avant l'échec des systèmes, vous pouvez configurer l'interface de programme d'application (API) pour assurer la distribution d'événements. En outre, vous pouvez améliorer les performances de la distribution d'événements avec les configurations de l'API.

---

### Journaux des messages

Dans des situations de résolution de problèmes, vous devez comprendre comment interpréter les messages et connaître les actions à exécuter pour résoudre un problème. Les fichiers journaux des messages vous permettent d'identifier et de résoudre les problèmes au sein de votre environnement.

Pour générer des journaux des messages pour l'interface de programme d'application (API) Java, indiquez des mots clés dans le fichier de configuration.

Les mots clés `LogLevel` et `LogFileName` vous permettent d'indiquer la quantité et la destination des messages.

Pour l'API C, indiquez le mot clé `ed_diag_config_file`.

#### Référence associée:

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

---

### Journaux de trace

Les journaux de trace vous aident à déterminer pourquoi un problème se produit. Les journaux de trace capturent des informations sur l'environnement d'exploitation lorsque Tivoli Event Integration Facility ne fonctionne pas comme souhaité.

Le personnel du service clients utilise les informations capturées par les consigneurs de trace pour tracer un problème jusqu'à sa source ou déterminer pourquoi une erreur se produit. Ces outils ne sont pas activés par défaut. Etant donné que les messages de trace sont destinés au service clients, ils sont généralement écrits dans un fichier qui peut être consulté à des fins d'examen ultérieur.

**Remarque :** La consignation de trace produit un grand nombre d'informations et peut générer de grands fichiers de trace. C'est pourquoi, n'activez la consignation de trace que temporairement et désactivez-la lorsque vous possédez suffisamment d'informations.

Pour générer des messages de trace pour l'interface de programme d'application (API) Java, vous devez indiquer des mots clés dans le fichier de configuration **postei fmsg**. Les mots clés `TraceLevel` et `TraceFileName` vous permettent d'indiquer la quantité et la destination du traçage.

Pour l'API C, vous devez indiquer le mot clé `ed_diag_config_file`.

Généralement, vous envoyez le fichier `/tmp/tec_ed` et le journal de débogage de la sonde EIF au service clients.

## Génération de messages de trace pour l'API C

Pour activer la consignation de trace pour l'API C, procédez comme suit :

1. Dans le répertoire `/tmp`, écrivez le contenu suivant dans un fichier et nommez ce fichier `.ed_diag_config` :

```
Highest_level trace2
Truncate_on_restart true

tec_ed Highest_level trace2

cache and buffering

tec_ed ed_buffer trace2 /tmp/tec_ed
tec_ed ed_cache trace2 /tmp/tec_ed
tec_ed ed_cache_controller trace2 /tmp/tec_ed

cMain drivers

tec_ed ed_agent_comm trace2 /tmp/tec_ed
tec_ed ed_agent_util trace2 /tmp/tec_ed
tec_ed ed_tc trace2 /tmp/tec_ed

TRANSPORTS

tec_ed ed_fwkw_t trace2 /tmp/tec_ed
tec_ed ed_lcf_t trace2 /tmp/tec_ed
tec_ed ed_socket_t trace2 /tmp/tec_ed

SOCKETS

tec_ed ed_eipc trace2 /tmp/tec_ed
tec_ed ed_socket_impl trace2 /tmp/tec_ed
tec_ed ed_socket_impl trace2 /tmp/tec_ed
tec_ed ed_socket_rcv trace2 /tmp/tec_ed

API calls to eipc

tec_ed ed_c_recv_msg trace2 /tmp/tec_ed
tec_ed ed_c_ipc_poll trace2 /tmp/tec_ed
tec_ed ed_c_ipc trace2 /tmp/tec_ed
tec_ed ed_c_ipc_accept trace2 /tmp/tec_ed
tec_ed ed_c_pool trace2 /tmp/tec_ed
tec_ed ed_c_ipc_server trace2 /tmp/tec_ed

API calls to ipc

tec_ed ed_recv_msg trace2 /tmp/tec_ed
tec_ed ed_ipc_poll trace2 /tmp/tec_ed
tec_ed ed_ipc trace2 /tmp/tec_ed
tec_ed ed_ipc_accept trace2 /tmp/tec_ed
tec_ed ed_pool trace2 /tmp/tec_ed
tec_ed ed_ipc_server trace2 /tmp/tec_ed
tec_ed ed_evmsg trace2 /tmp/tec_ed

misc

tec_ed hn_target trace2 /tmp/tec_ed
tec_ed EdDiag trace2 /tmp/tec_ed
```

2. Ajoutez le paramètre de propriété suivant au fichier de configuration **postei fmsg** :  
`ed_diag_config_file=/tmp/.ed_diag_config`
3. Répliquez le problème qui nécessite les informations de trace.
4. Supprimez le paramètre de propriété `ed_diag_config_file` du fichier de configuration **postei fmsg**.

**Référence associée:**

Annexe D, «Mots clés de configuration», à la page 77

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

«**postei fmsg**», à la page 22

L'utilitaire **postei fmsg** vous permet de publier des événements sur l'ObjectServer via la sonde Probe for Tivoli EIF. Après avoir installé Event Integration Facility, vous pouvez copier le fichier binaire **postei fmsg** sur d'autres hôtes.

---

## Performances et disponibilité

Tivoli Event Integration Facility peut contrôler les performances et la disponibilité du traitement d'événements.

Pour éviter toute surcharge lors de la distribution d'événement, vous pouvez ajouter des minuteurs pour la distribution d'événement et indiquer le nombre maximum d'événements dans le cache.

Lorsque vous utilisez des minuteurs, Event Integration Facility informe le récepteur de l'événement que des événements sont mis en cache. Cela se produit lorsque le minuteur expire ou lorsque le nombre maximum d'événements est dépassé. Par conséquent, le récepteur peut traiter tout le contenu du cache en une seule fois.

En outre, vous pouvez augmenter la disponibilité des événements en définissant des serveurs de sauvegarde.

**Tâches associées:**

«Activation du cache», à la page 28

Par défaut, le cache qui garantit la récupération après des échecs système stocke des événements. Les mots clés du fichier de configuration vous permettent de contrôler la configuration du cache.

«Configuration des serveurs de sauvegarde pour la distribution d'événement», à la page 29

La configuration utilise les sockets TCP pour distribuer des événements. Indiquez comment contacter les serveurs de sauvegarde en définissant le canal de transfert comme étant le nom de serveur et le numéro de port.

## Paramètres de connexion pour le processus de réception

Le processus de réception crée un socket de serveur qui écoute sur un port indiqué. Le processus d'écoute marque un socket en mode connexion comme acceptant les connexions et limite le nombre de connexions en attente dans la file d'attente du socket d'écoute à la valeur indiquée par la valeur `ConnectionsQueued`.

L'implémentation peut inclure des connexions incomplètes dans la file d'attente sujette à une limite. Les implémentations peuvent limiter la longueur du socket d'écoute mis en file d'attente en définissant la valeur `ConnectionsQueued`.

TCP/IP autorise les données à être envoyées et la connexion, à être fermée par l'expéditeur avant que l'application réceptrice ne puisse recevoir de données. Le processus de réception accepte finalement la connexion et traite les données. `ConnectionsQueued` permet à ces types de connexion d'être limités. Les tentatives de connexion sont refusées lorsque TCP/IP ne peut plus mettre en file d'attente les connexions.

Le nombre de connexions actives traitées par le processus de réception peut être limité par le mot clé `ActiveConnections`. Cela empêche le processus de réception d'accumuler trop de connexions qui utilisent des ressources système.

Si toutes les connexions initiales sont `ConnectionOriented` et si la valeur `ActiveConnections` a été atteinte, aucune connexion supplémentaire n'est acceptée par le processus de réception. Cependant, les connexions sont toujours considérées par le protocole TCP/IP jusqu'à ce que la valeur `ConnectionsQueued` ait été atteinte. Si toutes les connexions `ActiveConnections` sont `ConnectionOriented`, l'état reste inchangé jusqu'à ce que le nombre de connexions actives diminue pour atteindre la valeur indiquée par le pourcentage `ActiveConnections` via l'utilisation de `ActiveConnectionsSafety`.

**Remarque :** Les connexions traitées par le protocole TCP/IP (`ConnectionsQueued`) sont supprimées lorsque le programme se termine. Les connexions se trouvant dans la liste `ActiveConnections` sont toutes fermées et les données de ces connexions sont supprimées.

---

## Incidents et scénarios usuels

Lorsque vous utilisez Tivoli Event Integration Facility, vous pouvez rencontrer un certain nombre de problèmes et de scénarios usuels.

### Génération et exécution d'adaptateurs

Un certain nombre de scénarios peuvent se produire lorsque vous rencontrez des problèmes lors de la génération, de la compilation ou de l'exécution de l'adaptateur.

#### Mon adaptateur sur un système d'exploitation Windows rencontre des erreurs d'heure de compilation

**Cause** Il existe plusieurs fichiers `wsock32.lib`.

#### Solution

Utilisez l'option `FORCE:MULTIPLE` lors de la compilation de l'adaptateur Windows NT.

Cela force le compilateur à choisir un fichier.

### **J'ai modifié le fichier de configuration, mais mes modifications ne sont pas appliquées**

**Cause** Des instructions indiquées de manière incorrecte peuvent se trouver dans le fichier de configuration.

#### **Solution**

Consultez les éléments suivants pour les mots clés modifiés :

- Recherchez des erreurs de typographie dans l'orthographe des mots clés.
- Vérifiez que les blancs sont mis entre guillemets simples.

### **J'ai reçu une erreur LOG0014E**

**Cause** Le système ne retrouve pas le fichier indiqué par le mot clé LogFileName.

#### **Solution**

Corrigez le nom de chemin indiqué par le mot clé LogFileName.

## **Etablissement de connexions au serveur d'événements**

Un certain nombre de scénarios peuvent se produire lorsque vous rencontrez des problèmes lors de l'établissement de la connexion au serveur d'événements.

### **J'ai reçu une erreur de connexion lorsque j'ai utilisé posteifmsg**

**Cause** L'erreur indique que vous utilisez peut-être un ID utilisateur différent de Administrator ou root. Par conséquent, votre ID ne possède pas les droits corrects pour créer et écrire le fichier indiqué par le mot clé BufEvtPath.

#### **Solution**

Vérifiez que vous disposez des droits corrects pour créer le fichier indiqué par le mot clé BufEvtPath.

#### **Référence associée:**

«postEIFmsg», à la page 22

L'utilitaire **postEIFmsg** vous permet de publier des événements sur l'ObjectServer via la sonde Probe for Tivoli EIF. Après avoir installé Event Integration Facility, vous pouvez copier le fichier binaire **postEIFmsg** sur d'autres hôtes.

## **Envoi d'événements**

Problèmes communs pouvant se produire lorsque des événements sont envoyés au serveur d'événements.

### **Mon adaptateur n'envoie pas tous les événements au serveur d'événements.**

**Cause** Un problème de communication s'est produit entre l'adaptateur et le serveur d'événements ou vous rencontrez un problème interne lié au code d'adaptateur.

#### **Solution**

Envoyez les événements à un fichier au lieu de les envoyer directement au serveur d'événements. Vérifiez ensuite la distribution d'événement en procédant comme suit :

1. Définissez le mot clé TestMode sur YES dans le fichier de configuration d'adaptateur.
2. Indiquez le fichier dans lequel recevoir les événements avec le mot clé ServerLocation.
3. Redémarrez l'adaptateur.

4. Consultez le fichier indiqué par le mot clé `ServerLocation` et vérifiez si tous les événements s'y affichent.

Si tous les événements apparaissent dans le fichier, vous rencontrez un problème de communication avec le serveur d'événements. Si les événements sont manquants dans le fichier, vous rencontrez un problème interne lié à l'adaptateur. Vérifiez le code d'adaptateur.

Après avoir testé l'adaptateur, redéfinissez le mot clé `TestMode`, afin que les événements soient envoyés au serveur d'événements et ne soient plus envoyés à un fichier.

### **J'utilise la commande `postei fmsg` pour envoyer des événements, mais ils ne parviennent pas au serveur d'événements**

**Cause** Les événements sont envoyés au cache sur l'adaptateur ou la passerelle en raison de l'une des situations suivantes :

- Un associateur de port est en cours d'utilisation du côté expéditeur, mais pas sur le serveur d'événements. Pour les systèmes Windows, il n'existe aucun démon d'associateur de port.
- Un port, un nom d'hôte ou un serveur d'événements non valide a été indiqué.

#### **Solution**

Résolvez les problèmes de nom ou de port non valide.

### **Les caractères multi-octets dans un événement généré par `postei fmsg` sont illisibles**

Pour les environnements locaux qui utilisent des caractères multi-octets, un message dans un événement généré par `postei fmsg` est illisible. Le message illisible peut être affiché dans la liste d'événements, le journal de débogage de la sonde Probe for Tivoli EIF et le fichier cache de l'utilitaire `postei fmsg`.

**Cause** L'utilitaire `postei fmsg` ne parvient pas à convertir les messages locaux codés sur deux octets (DBCS), en japonais par exemple, en caractères codés en UTF-8 si la variable d'environnement `TISDIR` n'est pas définie.

#### **Solution**

Définissez la variable d'environnement `TISDIR` sur un chemin d'accès au répertoire codeset avant d'exécuter l'utilitaire `postei fmsg`. Ce répertoire est inclus dans le SDK de la fonction d'intégration d'événements (EIF, Event Integration Facility) pour Tivoli Netcool/OMNIBus.

Pour définir cette variable d'environnement :

1. Exécutez la commande suivante :

```
export TISDIR=/chemin_package_eif-sdk_extraite/codeset
```

2. Exécutez ensuite la commande suivante :

```
postei fmsg -f /chemin_fichier_configuration -r CRITICAL
-m "message inclus DBCS" CLASS SOURCE
```

### **Des processus `postei fmsg` ne se terminent pas**

Si un fichier cache est partagé par plusieurs processus `postei fmsg`, certains processus `postei fmsg` ne se terminent pas. Ils se bloquent.

**Cause** Plusieurs instances de l'utilitaire `postei fmsg` ne peuvent pas partager un fichier cache unique.

### Solution

Indiquez des fichiers de configuration uniques pour chaque instance de **postei fmsg**. Vous pouvez spécifier un fichier cache dans le fichier de configuration de chaque instance de **postei fmsg**.

Vous pouvez également utiliser l'utilitaire **nco\_postmsg** qui peut partager un fichier cache unique dans plusieurs instances. Si vous utilisez cette solution, veillez à indiquer un fichier journal unique pour chaque instance de **nco\_postmsg** à l'aide de l'option de ligne de commande `-message log`.

### Référence associée:

«postei fmsg», à la page 22

L'utilitaire **postei fmsg** vous permet de publier des événements sur l'ObjectServer via la sonde Probe for Tivoli EIF. Après avoir installé Event Integration Facility, vous pouvez copier le fichier binaire **postei fmsg** sur d'autres hôtes.



---

## Annexe A. Interfaces de programme d'application (API)

Les API de Tivoli Event Integration Facility sont définies par les fonctions du langage C et les méthodes Java. Elles vous permettent de générer un adaptateur personnalisé ou une application qui reçoit des événements.

### Tâches associées:

«Programmation de l'adaptateur», à la page 40

Pour programmer un adaptateur, implémentez les interfaces et les paramètres préférés pour le fichier de configuration. Décidez de définir les valeurs d'attribut dans le fichier de configuration ou dans le code d'adaptateur. Compilez ensuite et établissez une liaison avec l'adaptateur, le cas échéant.

### Référence associée:

«Mise à niveau d'adaptateurs existants», à la page 42

Pour mettre à niveau des adaptateurs existants, établissez une liaison entre vos adaptateurs et la bibliothèque `libeif.a`.

---

## Interface de programme d'application (API) du langage C

Afin de générer un adaptateur personnalisé dans le langage C, vous devez posséder un compilateur C.

### `tec_agent_getenv`

La fonction `tec_agent_getenv` récupère la valeur de la variable contenue dans le fichier de configuration.

### Synopsis

```
char *tec_agent_getenv(char *mot clé)
```

### Arguments

#### mot clé

La variable mot clé à récupérer.

### Exemples

```
#include "tec_eef.h"
char *serverLoc=tec_agent_getenv("ServerLocation")
```

### Codes retour

Renvoie un pointeur vers une chaîne qui correspond à la valeur de la variable. Ne libérez pas ce pointeur. Renvoie une valeur NULL si le mot clé n'apparaît pas dans le fichier de configuration.

## tec\_agent\_init

La fonction `tec_agent_init` est une fonction d'initialisation qui lit le fichier de configuration et met en cache les informations.

**Remarque :** La fonction `tec_agent_init` est la première fonction appelée pour initialiser le module Tivoli Event Integration Facility. Appelez la fonction `tec_agent_init` une seule fois pour chaque adaptateur et appelez-la avant toutes les autres fonctions.

### Synopsis

```
int tec_agent_init(char *fichier de configuration)
```

### Arguments

#### Fichier de configuration

Chemin d'accès complet au fichier de configuration.

### Exemples

```
#include "tec_eeif.h"
tec_agent_init("config");
```

### Codes retour

Renvoie 0 si l'appel de fonction aboutit.

## tec\_create{EIF\_handle

La fonction `tec_create{EIF_handle` établit un descripteur pour envoyer des événements au serveur d'événements ou recevoir des événements d'une source, le premier argument indiquant un fichier de configuration. Un descripteur est créé à l'aide des informations de configuration indiquées dans le fichier de configuration. Cette fonction est similaire à `tec_create_handle`.

### Synopsis

```
descripteur_tec_t tec_create{EIF_handle(char *fichier de configuration, entier
unidirectionnel, mode mode_distribution)
```

### Arguments

#### fichier de configuration

Chemin d'accès complet au fichier de configuration.

#### unidirectionnel

Utilisé pour les adaptateurs de noeud géré uniquement, pour désigner si les appels de la fonction `tec_put_event` renvoient des exceptions à l'appelant en cas d'échec. Une valeur de un (1) signifie que les exceptions ne sont, le cas échéant, pas renvoyées à l'appelant de la fonction `tec_put_event` car l'appelant n'attend pas une réponse du processus `oserv`. Une valeur de zéro (0) signifie que les exceptions sont renvoyées à l'appelant car ce dernier attend que le processus `oserv` renvoie la réussite ou l'échec de la méthode.

**mode** Les valeurs possibles sont les suivantes :

- submission
- reception

Utilisez le mode submission lorsque le descripteur est utilisé à des fins de transmission. Utilisez le mode reception lorsque le descripteur est utilisé pour la réception.

## Exemples

```
#include "tec_eeif.h"
if((th =tec_create{EIF_handle("config",0,submission))==NULL){
 fprintf(stderr,"%s:tec_create_handle failed errno=%d \n ",
 progname,tec_errno);
 exit(1);
}
```

## Codes retour

Un descripteur à une structure de données interne. Le descripteur est utilisé dans les appels d'autres fonctions de l'API.

## tec\_create\_handle

La fonction `tec_create_handle` établit un descripteur pour envoyer des événements au serveur d'événements.

### Synopsis

```
descripteur_tec_t tec_create_handle(char *emplacement, port court non signé,
entier unidirectionnel, type type_distribution_tec)
```

### Arguments

#### emplacement

Nom d'hôte ou adresse du protocole hôte du serveur d'événements.

#### unidirectionnel

Utilisé pour les adaptateurs de noeud géré uniquement, pour désigner si les appels de la fonction `tec_put_event` renvoient des exceptions à l'appelant en cas d'échec. Une valeur de un (1) signifie que les exceptions ne sont, le cas échéant, pas renvoyées à l'appelant de la fonction `tec_put_event` car l'appelant n'attend pas une réponse du processus `oserv`. Une valeur de zéro (0) signifie que les exceptions sont renvoyées à l'appelant car ce dernier attend que le processus `oserv` renvoie la réussite ou l'échec de la méthode.

**port** Port écouté par le serveur d'événements.

**type** Les valeurs possibles sont les suivantes :

- `connection_less`
- `connection_oriented`
- `use_default`

La valeur `use_default` lit le paramètre dans le fichier de configuration pour le mot clé `ConnectionMode` et configure un descripteur sans connexion si le mot clé `ConnectionMode` n'est pas défini.

## Exemples

```
#include "tec_eeif.h"
if((th = tec_create_handle(tec_server, port, oneway, type)) == NULL) {
 fprintf(stderr,"%s: tec_create_handle failed errno=",
 progname, tec_errno);
 exit(1);
}
```

## Codes retour

Un descripteur à une structure de données interne. Le descripteur est utilisé dans les appels de la fonction `tec_put_event`. Si l'emplacement est NULL, l'entrée `ServerLocation` du fichier de configuration est utilisée pour déduire l'emplacement. Si le port est zéro, l'entrée `ServerPort` est utilisée, le cas échéant. Sinon, l'associateur de port est interrogé pour connaître le port d'écoute du serveur d'événements.

## `tec_create_handle_c`

La fonction `tec_create_handle_c` établit un descripteur pour envoyer des événements au serveur d'événements ou recevoir des événements d'une source, le premier argument indiquant un fichier de configuration. Un descripteur est créé à l'aide des informations de configuration indiquées dans le fichier de configuration. Cette fonction est similaire à `tec_create_handle`.

### Synopsis

```
descripteur_tec_t tec_create_handle_c (char *fichier de configuration, char *emplacement,
port court non signé, entier unidirectionnel, type type_distribution_tec, mode mode_distribution)
```

### Arguments

#### fichier de configuration

Chemin d'accès complet au fichier de configuration.

#### emplacement

Nom d'hôte ou adresse du protocole hôte du serveur d'événements.

**mode** Les valeurs possibles sont les suivantes :

- `submission`
- `reception`

Utilisez le mode `submission` lorsque le descripteur est utilisé à des fins de transmission. Utilisez le mode `reception` lorsque le descripteur est utilisé pour la réception.

#### unidirectionnel

Utilisé pour les adaptateurs de noeud géré uniquement, pour désigner si les appels de la fonction `tec_put_event` renvoient des exceptions à l'appelant en cas d'échec. Une valeur de un (1) signifie que les exceptions ne sont, le cas échéant, pas renvoyées à l'appelant de la fonction `tec_put_event` car l'appelant n'attend pas une réponse du processus `oserv`. Une valeur de zéro (0) signifie que les exceptions sont renvoyées à l'appelant car ce dernier attend que le processus `oserv` renvoie la réussite ou l'échec de la méthode.

**port** Port d'écoute du serveur d'événements pour les versions non TME.

**type** Les valeurs possibles sont les suivantes :

- `connection_less`
- `connection_oriented`
- `use_default`

La valeur `use_default` lit le paramètre dans le fichier de configuration pour le mot clé `ConnectionMode` et configure un descripteur sans connexion si le mot clé `ConnectionMode` n'est pas défini.

## Exemples

```
#include "tec_eeif.h"
if((th
=tec_create_handle_c("config","tecserver.com",5529,0,connection_less,submission))
==NULL){
 fprintf(stderr,"%s:tec_create_handle failed errno=%d \n ",
 progname,tec_errno);
 exit(1);
}
```

## Codes retour

Un descripteur à une structure de données interne. Le descripteur est utilisé dans les appels de la fonction `tec_put_event`. Si l'emplacement est NULL, l'entrée `ServerLocation` du fichier de configuration est utilisée pour déduire l'emplacement. Si le port est zéro, l'entrée `ServerPort` est utilisée, le cas échéant. Sinon, l'associateur de port est interrogé pour connaître le port d'écoute du serveur d'événements.

## tec\_create\_handle\_r

La fonction `tec_create_handle_r` établit un descripteur pour envoyer des événements au serveur d'événements, le premier argument indiquant un fichier de configuration. Un descripteur est créé à l'aide des informations de configuration indiquées dans le fichier de configuration. Similaire à `tec_create_handle`.

## Synopsis

```
descripteur_tec_t tec_create_handle_r (char *fichier de configuration, char *emplacement,
port court non signé, entier unidirectionnel, type type_distribution_tec)
```

## Arguments

### fichier de configuration

Chemin d'accès complet au fichier de configuration.

### emplacement

Nom d'hôte ou adresse du protocole hôte du serveur d'événements.

### unidirectionnel

Utilisé pour les adaptateurs de noeud géré uniquement, pour désigner si les appels de la fonction `tec_put_event` renvoient des exceptions à l'appelant en cas d'échec. Une valeur de un (1) signifie que les exceptions ne sont, le cas échéant, pas renvoyées à l'appelant de la fonction `tec_put_event` car l'appelant n'attend pas une réponse du processus `oserv`. Une valeur de zéro (0) signifie que les exceptions sont renvoyées à l'appelant car ce dernier attend que le processus `oserv` renvoie la réussite ou l'échec de la méthode.

**port** Port d'écoute du serveur d'événements pour les versions non TME.

**type** Les valeurs possibles sont les suivantes :

- `connection_less`
- `connection_oriented`
- `use_default`

La valeur `use_default` lit le paramètre dans le fichier de configuration pour le mot clé `ConnectionMode` et configure un descripteur sans connexion si le mot clé `ConnectionMode` n'est pas défini.

## Exemples

```
#include "tec_eeif.h"
th = tec_create_handle_r("config","localhost",1234, 0, use_default) == NULL) */
```

## Codes retour

Un descripteur à une structure de données interne. Le descripteur est utilisé dans les appels de la fonction `tec_put_event`. Si l'emplacement est NULL, l'entrée `ServerLocation` du fichier de configuration est utilisée pour déduire l'emplacement. Si le port est zéro, l'entrée `ServerPort` est utilisée, le cas échéant. Sinon, l'associateur de port est interrogé pour connaître le port d'écoute du serveur d'événements.

## tec\_destroy\_handle

La fonction `tec_destroy_handle` détruit le descripteur du serveur d'événements créé par `tec_create_handle`, `tec_create_handle_c`, `tec_create{EIF_handle` et `tec_create_handle_r`, puis ferme toutes les connexions établies.

### Synopsis

```
annuler tec_destroy_handle (descripteur_tect_t th)
```

### Arguments

**th** Descripteur tec renvoyé par un appel de la fonction `create_handle`.

### Exemples

```
#include "tec_eeif.h"
tec_destroy_handle(th);
```

## tec\_errno

Lorsqu'une fonction renvoie une erreur, la fonction `tec_errno` est définie sur le code d'erreur approprié.

### Synopsis

```
externe entier tec_errno
```

## tec\_get\_event

La fonction `tec_get_event` permet à une application de recevoir des événements. Elle reçoit des événements sur demande, à partir du mode de transfert configuré. Les données renvoyées peuvent contenir plusieurs événements. Utilisez l'utilitaire `ed_scan_n` pour déterminer le nombre d'événements. La mémoire allouée pour l'événement doit être libérée.

### Synopsis

```
long tec_get_event (descripteur_tec_t unsigned char ** event_message);
```

### Arguments

**th** Descripteur du serveur d'événements renvoyé par un appel de la fonction `create_handle`.

#### **event\_message**

Contient les données d'événement du message reçues d'un transfert.

## Exemples

```
#include "tec_eeif.h"
char *event;
long event_len;
int rc;

event=NULL;
event_len = tec_get_event(th, &event);
if (event && event_len)
{
 n =ed_scan_n (event,event_len);
}
if (event)
 free(event)
```

## Codes retour

Renvoie la longueur du message d'événement. Renvoie 0 (zéro) lorsqu'aucun événement n'est disponible.

## tec\_put\_event

La fonction `tec_put_event` envoie un événement au serveur d'événements.

**Remarque :** Si la mise en mémoire tampon est activée dans le fichier de configuration, les événements sont placés dans une mémoire tampon, puis envoyés sur une unité d'exécution distincte. Si un appel de la fonction `tec_put_event` est suivi d'un appel de la fonction `tec_destroy_handle`, le descripteur peut être détruit avant l'envoi de l'événement. Pour empêcher la survenue de telles situations, utilisez la fonction de programme utilitaire `ed_sleep` après la fonction `tec_put_event` pour laisser du temps afin que l'événement puisse être envoyé :

```
ed_sleep(0,100);
```

## Synopsis

```
long tec_put_event (descripteur_tec_t th, char *événement)
```

## Arguments

### événement

Chaîne de caractères représentant l'événement.

**th** Descripteur du serveur d'événements renvoyé par un appel de la fonction `create_handle`.

## Exemples

```
#include "tec_eeif.h"
if (tec_put_event(th, event_string) == -1 {
 fprintf(stderr, "%s: tec_put_event failed, errno=,
 progname, tec_errno);
 exit(1);
}
```

## Codes retour

Renvoie le nombre d'octets envoyés au serveur d'événements, d'autres applications écoutant les événements ou le fichier de cache. Un zéro renvoyé signifie que l'événement est filtré. Une valeur négative renvoyée indique une erreur.

## tec\_register\_callback

La fonction `tec_register_callback` permet à une application de recevoir des événements via un appel ascendant. L'application enregistre un rappel et transmet en tant que paramètre la méthode qui gère les événements reçus.

### Syntaxe de la fonction `tec_event_callback`

La syntaxe de la fonction `tec_event_callback` est la suivante :

```
int (*tec_event_callback)(tec_handle_t h, unsigned char *msg, long _msg_len);
```

Les données renvoyées peuvent contenir plusieurs événements. Utilisez l'utilitaire `ed_scan_n` pour déterminer le nombre d'événements renvoyés. La mémoire allouée pour l'événement doit être libérée.

La fonction `tec_event_callback` renvoie `-1` ou zéro (`0`). Un zéro indique qu'aucune erreur n'a été rencontrée et que l'événement a été traité. `-1` indique qu'un problème a été rencontré lors du traitement de l'événement et qu'il ne faut pas le supprimer du cache si un cache est configuré.

### Synopsis

```
annuler tec_register_callback(descripteur_tec_t th, rappel_événement_tec *fn)
```

### Arguments

**th**      Descripteur du serveur d'événements renvoyé par un appel de la fonction `create_handle`.

**fn**      Fonction à appeler lorsque l'événement se produit.

### Exemples

```
#include "tec_eef.h"
int on_message (tec_handle_t th, unsigned char *event, long event_len)
{
 long n;
 if (event && event_len)
 {
 int i;
 char *ev;
 long len;
 long idx = 1;
 n = ed_scan_n (event, event_len);
 for(i = 0; i < n; i++, idx++)
 {
 ev = (char *)ed_scan_get_n ((char *)event, idx, event_len, &len);
 free (ev);
 }
 }
 return 0;
}
tec_register_callback(th, on_message)
```

---

## Annexe B. Utilitaires de l'interface de programme d'application (API) C

Tivoli Event Integration Facility fournit un certain nombre d'utilitaires pour l'API C.

---

### ed\_scan\_get\_n

Les événements reçus via l'interface de programme d'application (API) de réception peuvent contenir plusieurs événements. Utilisez l'utilitaire `ed_scan_get_n` pour obtenir l'événement *n* du paquet.

#### Synopsis

```
char * ed_scan_get_n (char *paquet, index long, longueur_paquet long, long *longueur_résultante);
```

#### Arguments

##### paquet

Pointeur renvoyé par `tec_get_event` ou transmis au rappel lorsque la fonction `tec_register_callback` est utilisée.

**index** Élément *n* du paquet commençant par 1.

##### Longueur\_paquet

Longueur maximale du paquet à rechercher.

##### Longueur\_résultante

Contient la longueur résultante du paquet.

#### Exemples

```
#include "tec_eeif.h"
char *package
char *ev;
int i;
long n, len, idx=1;
package=tec_get_event(th);
n = ed_scan_n (package, strlen (package));
for (i = 0; i < n; i++, idx++)
{
ev =(char *) ed_scan_get_n ((char *) package, idx, strlen (package), &len);
free (ev);
}
```

#### Codes retour

Renvoie un pointeur à une mémoire tampon récemment allouée contenant l'événement souhaité. Ce pointeur doit être libéré.

---

## ed\_scan\_n

Les événements reçus via l'interface de programme d'application (API) de réception peuvent contenir plusieurs événements. Utilisez l'utilitaire `ed_scan_n` pour déterminer le nombre d'événements contenus dans le paquet.

### Synopsis

```
long ed_scan_n (char *paquet, longueur_paquet long);
```

### Arguments

#### paquet

Pointeur renvoyé par `tec_get_event` ou transmis au rappel lorsque la fonction `tec_register_callback` est utilisée.

#### Longueur\_paquet

Longueur maximale du paquet à rechercher.

### Exemples

```
#include "tec_eeif.h"
char *package
char *ev;
int i;
long n,len,idx=1;
package=tec_get_event(th);
n = ed_scan_n (package, strlen (package));
for (i = 0; i < n; i++, idx++)
{
ev =(char *) ed_scan_get_n ((char *) package, idx, strlen (package),&len);
free (ev);
}
```

### Codes retour

Renvoie le nombre d'événements : zéro (0) si aucun événement n'est détecté et -1 lorsqu'une erreur se produit.

---

## ed\_sleep

L'utilitaire `ed_sleep` arrête temporairement l'exécution pendant la durée indiquée et autorise les unités d'exécution à basculer. Cet utilitaire est appelé par les adaptateurs de noeud géré dans la boucle principale pour libérer l'exécution des unités d'exécution internes.

### Synopsis

```
entier ed_sleep (durée en secondes, durée en millisecondes);
```

### Arguments

#### durée en millisecondes

Indique le nombre de millisecondes.

#### durée en secondes

Indique le nombre de secondes.

## Exemples

```
#include "tec_eef.h"
/* pause for 3.5 seconds */
ed_sleep (3, 500);
/* sleep duration is zero but yields briefly so other threads can run */
ed_sleep (0, 0);
```

## Codes retour

Renvoie zéro (0).



---

## Annexe C. Interface de programme d'application (API) du langage Java

Pour générer un adaptateur dans Java, vous devez posséder le compilateur Java version 1.6.0 (ou ultérieure). L'API Java est fournie dans les fichiers Jar.

---

### disconnect

La fonction disconnect ferme toutes les connexions ouvertes vers le serveur d'événements.

#### Synopsis

```
disconnect()
```

#### Exemples

```
public synchronized void disconnect()
```

---

### disconnect(time)

La fonction disconnect(time) vide le cache puis ferme toutes les connexions ouvertes vers le serveur d'événements.

#### Synopsis

```
disconnect (délai d'attente maximum en secondes)
```

#### Arguments

**Remarque :** Pour toutes les valeurs, la commande est renvoyée à l'appelant dès que le cache est vidé.

- < 0     Attente sans fin ou jusqu'au vidage du cache. Si le serveur est hors tension, cette routine ne sera pas renvoyée.
- = 0     Envoi d'événements du cache alors que la connexion est toujours ouverte. Cela peut prendre un certain temps si le cache est volumineux.
- >= 0    Arrêt du vidage du cache après le nombre de secondes indiqué.

#### Exemples

```
public synchronized void disconnect(int délai d'attente maximum en secondes)
```

---

### getConfigVal

La fonction getConfigVal extrait la valeur d'une variable contenue dans le fichier de configuration.

#### Synopsis

```
getConfigVal (String clé)
```

clé     Indique l'étiquette de mot clé de configuration.

#### Exemples

```
public String getConfigVal(String clé)
```

## Codes retour

Renvoie la valeur de chaîne associée à la variable *clé* dans le fichier de configuration utilisé pour initialiser la fonction TECAgent.

Si la variable *clé* ne figure pas dans le fichier de configuration, la valeur null est renvoyée. Pour les mots clés tels que Filter, qui ont plusieurs valeurs, seule la dernière valeur indiquée dans les fichiers de configuration est renvoyée.

---

## onMessage

La fonction onMessage traite les événements reçus de manière asynchrone depuis l'interface de programme d'application (API).

### Synopsis

```
onMessage (String événement)
```

### Arguments

#### événement

Chaîne contenue dans l'événement, qui est renvoyée à l'application.

### Exemples

```
public void onMessage(String événement)
```

## Codes retour

L'application renvoie la valeur true si l'événement reçu a été traité avec succès, ce qui indique à l'API de supprimer l'événement du cache et/ou du journal permanent afin d'éviter de renvoyer l'événement à l'avenir.

L'application renvoie la valeur false si l'événement n'a pas été traité avec succès, ce qui indique à l'API de renvoyer l'événement au serveur d'événements à des fins de traitement. Dans ce cas, l'événement n'est pas supprimé du cache et/ou du journal permanent.

---

## receiveEvent

La fonction receiveEvent permet à une application de recevoir des événements de manière synchrone. Elle reçoit des événements de tous les serveurs d'événements indiqués dans le fichier de configuration utilisé pour initialiser la fonction TECAgent.

### Synopsis

```
String receiveEvent()
```

### Exemples

```
public synchronized String receiveEvent()
```

## Codes retour

Renvoie des données d'événement reçues par cette interface de programme d'application (API). Le délimiteur TECEvent.TECAD\_EVENT\_END\_CHAR (A) sépare les événements renvoyés dans cette chaîne.

**Référence associée:**

«registerListener»

La fonction registerListener enregistre l'application appelante en tant que programme d'écoute, permettant ainsi la réception asynchrone d'événements. Vous devez transmettre un objet qui implémente l'interface IEventProcessing en tant que paramètre.

---

## registerListener

La fonction registerListener enregistre l'application appelante en tant que programme d'écoute, permettant ainsi la réception asynchrone d'événements. Vous devez transmettre un objet qui implémente l'interface IEventProcessing en tant que paramètre.

### Synopsis

registerListener (IEventProcessing)

### Arguments

#### IEventProcessing

Classe d'application qui implémente le rappel IEventProcessing.

### Exemples

```
public void registerListener (IEventProcessing)
```

#### Référence associée:

«receiveEvent», à la page 72

La fonction receiveEvent permet à une application de recevoir des événements de manière synchrone. Elle reçoit des événements de tous les serveurs d'événements indiqués dans le fichier de configuration utilisé pour initialiser la fonction TECAgent.

---

## sendEvent

Cette fonction envoie des événements aux serveurs d'événements indiqués dans le fichier de configuration utilisé pour initialiser la fonction TECAgent. Vous devez transmettre une fonction TECEvent sérialisée en tant que paramètre. Si le mot clé BufferEvents=YES est indiqué, les événements sont mis en cache et conservés jusqu'à leur envoi.

### Synopsis

sendEvent (String *événement*)

### Arguments

#### événement

Données d'événement à envoyer au serveur d'événements. Si la valeur événement est non-null, elle doit être d'une longueur minimale de TECEvent.MIN\_EVENT\_LEN caractères ou sendEvent renvoie immédiatement une erreur.

### Exemples

```
public synchronized int sendEvent(String événement)
```

## Codes retour

Renvoie le nombre d'octets envoyés au serveur d'événements, d'autres applications écoutant les événements ou le fichier de cache. Un zéro renvoyé signifie que l'événement est filtré. Une valeur négative renvoyée indique une erreur.

---

## TECAgent

La fonction TECAgent accède au fichier de configuration et définit le mécanisme de transfert. C'est l'objet de niveau supérieur qui permet l'envoi et la réception d'événements vers et depuis le serveur d'événements.

### Synopsis

```
TECAgent (Reader configStream, int mode_distribution, boolean oneway)
```

### Arguments

#### configStream

Objet qui lit les mots clés de configuration.

#### deliveryMode

Indique le mode de distribution. Les valeurs sont SENDER\_MODE et RECEIVE\_MODE.

#### oneway

Ce paramètre n'a pas de signification.

### Exemples

```
public TECAgent(Reader configStream, int mode_distribution, boolean oneway)
```

### Codes retour

Une exception est émise si TECAgent ne peut pas être créé.

---

## TECEvent

La fonction TECEvent encapsule le code pour analyser les définitions d'événement sur le plan syntaxique en paires nom de classe/attribut=valeur.

### Synopsis

```
TECEvent()
init(String événement)
```

### Arguments

#### événement

La chaîne d'événement doit être analysée sur le plan syntaxique. Voici quelques exemples de chaînes d'événement valides :

```
Class1;msg='text.';hostname=artemis;source=TEC;END
Class2;END
Class3;msg=theMessage;END
```

Une chaîne d'événement valide a le format suivant :

```
ID SEMICLN (ID = (STRING | VALUE | EMPTY_STRING) SEMICLN)* "END"[CTRL_A]
```

Les jetons de la grammaire de la chaîne d'événement sont expliqués ci-dessous :

```
SEMICLN := ";"
```

```
EQUALS := "="
CNTRL_A := "\001"
```

ID := toute séquence non vide de caractères appartenant au jeu  
a-z, A-Z, 0-9, \_, -, .

et contenant au moins un caractère du jeu  
a-z, A-Z

```
STRING :=
```

commence et se termine par des guillemets simples ou des guillemets normaux. Tous les guillemets imbriqués sont identiques aux guillemets utilisés pour délimiter la chaîne. Ils doivent être échappés par le même guillemet. Par exemple :

- 'embedded single(') s'écrirait 'embedded single(')'
- "embedded double(") s'écrirait "embedded double(")"
- "embedded single(') s'écrirait "embedded single(')"

**Remarque :** Les jetons STRING ne peuvent pas contenir le caractère NUL ('\\000') ou Ctrl-A ('\\001')

```
VALUE :=
```

toute séquence non vide de caractères excluant les éléments suivants :

- tous les caractères de commande ASCII (" - ")
- le caractère espace (" ")
- les guillemets simples (" ")
- Le signe égal ("=")
- le point-virgule (";")

EMPTY\_STRING := ce jeton représente toute chaîne vide. Les guillemets ne sont pas obligatoires pour cette valeur.

Le premier jeton identificateur est le nom de classe de l'événement. La séquence d'instances de "ID=(STRING|VALUE);" indique les paires emplacement/valeur et "END" marque la fin de l'événement. Le caractère de fin ^A est facultatif. Chaque ID utilisé en tant que nom d'emplacement doit être unique par rapport à tous les autres ID utilisés comme noms d'emplacement. Le nom d'emplacement ne peut pas être "END". Une quantité arbitraire de blancs (caractères " ", "\\t", "\\r" et "\\n") peut se trouver avant et après les jetons dans une chaîne d'événement, avec l'exception suivante. Si un caractère de fin ^A est présent, rien ne peut apparaître après. Voici des exemples de chaînes d'événement valides :

```
Class1;
 msg='embedded quote ''.' ;
 hostname=artemis;
END

Class2;END^A
Class3; msg = theMessage ; END
Class4;
 msg='Here''s a newline
 rest of msg';
END
```

## Exemples

```
public boolean init(String événement);
```

## Codes retour

Le code retour de l'appel `init()` est `true` lorsque l'analyse syntaxique de la chaîne d'événement aboutit, et `false` lorsque ce n'est pas le cas.

---

## Annexe D. Mots clés de configuration

Les mots clés suivants du fichier de configuration vous permettent de configurer le comportement des adaptateurs.

**Important :** Tous les mots clés ne s'appliquent pas à tous les adaptateurs. Certains adaptateurs disposent de mots clés supplémentaires qui leur sont spécifiques. Les adaptateurs n'émettent pas de messages d'erreur pour les mots clés mal orthographiés ou définis sur une valeur non valide.

### Format

Les mots clés utilisent le format suivant :

keyword=value

Entrez chaque mot clé sur une ligne distincte. N'utilisez pas de blancs dans les instructions de mots clés à moins de les placer entre guillemets simples.

Le tableau suivant répertorie les mots clés contenus dans un fichier de configuration. Ces mots clés sont communs à la plupart des adaptateurs. Certains mots clés sont spécifiques de l'API C ou de l'API Java.

Tableau 8. Mots clés de configuration pour la fonction d'intégration d'événements

| Mot clé              | Obligatoire ou facultatif                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------|-------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>BufEvtMaxSize</b> | Facultatif                                                                                                  | Indique la taille maximale, en kilooctets, du fichier cache de l'adaptateur. La valeur par défaut est de 64 et la taille minimale est de 8. Les tailles de fichier en-deçà de ce niveau sont ignorées et la valeur 8 Ko est utilisée. Il n'existe aucune limite supérieure de taille de fichier. Le fichier cache stocke les événements sur le disque lorsqu'ils ne peuvent pas être envoyés au serveur d'événements et que le mot clé <b>BufferEvents</b> est défini sur YES.<br><b>A faire :</b> Si le fichier cache existe déjà, supprimez le fichier afin que les modifications prennent effet. |
| <b>BufEvtPath</b>    | Obligatoire lorsque le mot clé <b>BufferEvents</b> est défini sur YES. Si tel n'est pas le cas, facultatif. | Indique le chemin d'accès complet du fichier cache de l'adaptateur. La valeur par défaut est cache.dat.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

Tableau 8. Mots clés de configuration pour la fonction d'intégration d'événements (suite)

| Mot clé                | Obligatoire ou facultatif | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>BufferEvents</b>    | Facultatif                | <p>Indique si et comment le stockage temporaire d'événements fonctionne. Les valeurs possibles sont les suivantes :</p> <ul style="list-style-type: none"> <li>• YES (par défaut) : les événements sont stockés dans le fichier spécifié par le mot clé <b>BufEvtPath</b>.</li> <li>• MEMORY_ONLY : les événements sont mis en mémoire tampon.</li> <li>• NO : aucun événement n'est stocké ni mis en mémoire tampon.</li> </ul> <p>Si le mot clé <b>BufferEvents</b> est défini sur YES, l'API stocke également les événements dans les fichiers spécifiés par le mot clé <b>BufEvtPath</b>.</p>                                                                                                                                                                                                    |
| <b>BufferFlushRate</b> | Facultatif                | <p>Indique le nombre d'événements envoyés par minute.</p> <p>Si une connexion est perdue et la mémoire tampon contient des événements, ces derniers sont envoyés à cette vitesse une fois la connexion rétablie. La valeur par défaut est 0, ce qui signifie que tous les événements sont envoyés simultanément.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>ConnectionMode</b>  | Facultatif                | <p>Indique le mode de connexion utilisé pour se connecter au serveur d'événements. Les valeurs possibles sont les suivantes :</p> <ul style="list-style-type: none"> <li>• <code>connection_oriented</code> (par défaut) : une connexion est établie lorsque l'adaptateur est initialisé. La connexion est conservée pour tous les événements qui sont envoyés. Une nouvelle connexion est établie uniquement si la connexion initiale est perdue. La connexion est supprimée après l'arrêt de l'adaptateur.</li> <li>• <code>connection_less</code> : une nouvelle connexion est établie et supprimée pour chaque événement ou groupe d'événements envoyé.</li> </ul> <p><b>Conseil :</b> Vous pouvez abrégier l'option <code>connection_oriented</code> en <code>co</code> ou <code>C0</code>.</p> |

Tableau 8. Mots clés de configuration pour la fonction d'intégration d'événements (suite)

| Mot clé                                   | Obligatoire ou facultatif                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>C</b> <code>ed_diag_config_file</code> | Facultatif                                                                                                 | <p>Écrit des messages de journal et de trace dans le fichier <code>./diag_config</code>, qui est le fichier par défaut de ce mot clé. Ce fichier doit être présent pour permettre la consignation et le traçage.</p> <p>Pour permettre la consignation, spécifiez l'un des niveaux de consignation suivants dans le fichier :</p> <ul style="list-style-type: none"> <li>• error</li> <li>• warning</li> </ul> <p>Pour permettre le traçage, spécifiez l'un des niveaux de traçage suivants dans le fichier :</p> <ul style="list-style-type: none"> <li>• trace0</li> <li>• trace1</li> <li>• trace2</li> </ul> <p>Si vous augmentez le niveau de traçage, la sortie de trace augmente également.</p> <p>Lorsque Event Integration Facility redémarre, l'API tronque le fichier de trace <code>/tmp/tec_ed</code>. Pour éviter la troncature, définissez le mot clé <b>Truncate_on_restart</b> sur <code>false</code>.</p> |
| <b>Filter</b>                             | Obligatoire lorsque le mot clé <b>FilterCache</b> est défini sur YES. Si tel n'est pas le cas, facultatif. | <p>Détermine le mode de filtrage des événements. Ce mot clé est utilisé avec le mot clé <b>FilterMode</b>.</p> <p>Un événement correspond à une instruction filter lorsque chaque paire <code>attribut=valeur</code> de l'instruction filter est identique à la paire <code>attribut=valeur</code> correspondante dans l'événement.</p> <p>Une instruction filter doit contenir la classe d'événement. Eventuellement, elle peut aussi contenir toute autre paire <code>attribut=valeur</code> définie pour la classe d'événement.</p> <p>Le format d'une instruction filter est le suivant.</p> <pre>Filter:Class=nom_classe; [attribut=valeur;...;attribut=valeur]</pre> <p>Chaque instruction doit se trouver sur une ligne distincte. La paire <code>attribut=valeur</code> est sensible à la casse.</p>                                                                                                                |

Tableau 8. Mots clés de configuration pour la fonction d'intégration d'événements (suite)

| Mot clé            | Obligatoire ou facultatif                                                                                                            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FilterCache</b> | Facultatif                                                                                                                           | <p>Détermine quels événements sont stockés dans le cache lorsque des événements ne peuvent pas être envoyés au serveur d'événements. Ce mot clé est utilisé avec le mot clé <b>FilterMode</b> et le mot clé <b>Filter</b>.</p> <p>Pour activer la mise en cache d'événements, définissez le mot clé <b>BufferEvents</b> sur YES.</p> <p>Un événement correspond à une instruction <b>FilterCache</b> lorsque chaque paire attribut=valeur de l'instruction <b>FilterCache</b> est identique à la paire attribut=valeur correspondante dans l'événement.</p> <p>Une instruction <b>FilterCache</b> doit contenir la classe d'événement (nom_classe) et peut inclure toute paire attribut=valeur définie pour la classe d'événement.</p> <p>Le format d'une instruction de filtrage est le suivant.</p> <pre>FilterCache:Class=nom_classe; [attribution=valeur;...;attribution=valeur]</pre> <p>Chaque instruction doit se trouver sur une ligne distincte. La paire attribut=valeur est sensible à la casse. L'instruction <b>FilterCache</b> doit indiquer la même classe ou un sous-ensemble de classes que ceux indiqués par l'instruction filter.</p> <p><b>Remarque :</b> Si vous utilisez ce mot clé avec des adaptateurs de noeud final, vérifiez que les instructions filter sur les deux noeuds sont identiques.</p> |
| <b>FilterMode</b>  | Obligatoire si une ou plusieurs instructions filter ou instructions filter cache sont définies. Si tel n'est pas le cas, facultatif. | <p>Indique si les événements correspondant à une instruction filter ou filter sont envoyés au serveur d'événements ou supprimés. Les valeurs possibles sont les suivantes :</p> <ul style="list-style-type: none"> <li>• IN : les événements sont envoyés au serveur d'événements.</li> <li>• OUT (par défaut) : les événements sont supprimés.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

Tableau 8. Mots clés de configuration pour la fonction d'intégration d'événements (suite)

| Mot clé                        | Obligatoire ou facultatif                                                                                                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FQDomain</b>                | Facultatif                                                                                                                | <p>Valide pour OpenView, SNMP, le fichier journal d'UNIX et les adaptateurs de journal d'événements Windows.</p> <p>Ce mot clé spécifie comment l'adaptateur définit la valeur de l'attribut fqhostname des événements envoyés au serveur d'événements. Cet attribut est utilisé pour indiquer le nom d'hôte qualifié complet de l'hôte d'origine. Les valeurs possibles sont les suivantes :</p> <ul style="list-style-type: none"> <li>• YES : l'adaptateur tente de déterminer le nom d'hôte qualifié complet. Si le nom d'hôte peut être déterminé, l'attribut fqhostname est défini sur ce nom d'hôte. Si tel n'est pas le cas, cet attribut a une valeur NULL.</li> <li>• NO : l'attribut fqhostname a une valeur NULL.</li> <li>• <i>suffixe.domaine.qualifié.complet</i> : si vous définissez ce mot clé sur un nom de système hôte qualifié complet, l'attribut fqhostname est défini sur le nom d'hôte ajouté à cette valeur.</li> </ul> |
| <b>Java</b> <b>LogFileName</b> | Facultatif                                                                                                                | <p>Spécifie le nom du chemin d'accès complet et du fichier journal. La valeur par défaut est \$TIVOLIHOME/tec/eif.log. Si vous indiquez un nom de chemin non valide, l'API renvoie l'erreur suivante :</p> <pre>LOG0014E Unable to open the handler output file &lt;nom de fichier&gt;. java.io.FileNotFoundException: &lt;nom de fichier&gt; (The system cannot find the path specified)</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>MaxPacketSize</b>           | Facultatif                                                                                                                | <p>Indique le nombre d'octets à envoyer au débit indiqué par le mot clé <b>BufferFlushRate</b>. La valeur par défaut est 0, ce qui signifie qu'un événement est envoyé simultanément.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>NO_UTF8_CONVERSION</b>      | Facultatif                                                                                                                | <p>Indique si Event Integration Facility code des données au format UTF-8. Les valeurs possibles sont les suivantes :</p> <ul style="list-style-type: none"> <li>• YES : les données d'événement ne sont pas codées au format UTF-8. Les données sont supposées déjà être au codage UTF-8 lors de leur transmission à Event Integration Facility. Les données sont ajoutées en préfixe avec l'indicateur précisant que les données sont au codage UTF-8 si cet indicateur n'existe pas au début des données d'événement.</li> <li>• NO (par défaut) : les données d'événement sont codées au format UTF-8.</li> </ul>                                                                                                                                                                                                                                                                                                                              |
| <b>Pre37Server</b>             | Obligatoire si vous définissez le mot clé <b>Pre37ServerEncoding</b> sur un langage. Si tel n'est pas le cas, facultatif. | <p>Indique si l'adaptateur envoie des événements dans le langage de l'hôte du serveur d'événements ou au codage UTF-8. Les valeurs possibles sont les suivantes :</p> <ul style="list-style-type: none"> <li>• YES : désactive le codage UTF-8.</li> <li>• NO (par défaut) : l'adaptateur envoie des événements codés en UTF-8.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

Tableau 8. Mots clés de configuration pour la fonction d'intégration d'événements (suite)

| Mot clé                    | Obligatoire ou facultatif                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Pre37ServerEncoding</b> | Obligatoire si vous définissez le mot clé <b>Pre37Server</b> sur YES. Si tel n'est pas le cas, facultatif.                                                                                                                                                                                                                                                                                                                                                                                                                 | Détermine le langage à utiliser lorsqu'un adaptateur non TME communique avec un hôte de serveur d'événements non UTF-8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>RetryInterval</b>       | Facultatif                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <p>Utilisé lorsque le mot clé <b>ConnectionMode</b> est défini sur <code>connection_oriented</code>. Ce mot clé spécifie la durée, en secondes, pendant laquelle un adaptateur attend après la perte de la connexion vers le serveur d'événements principal ou de secours ou mettre les événements en mémoire tampon. L'adaptateur ne traite aucun nouvel événement pendant cet intervalle. La valeur par défaut est 120.</p> <p>Ce mot clé est utile car l'adaptateur peut envoyer tous les événements au serveur d'événements principal, y compris si le serveur d'événements est brièvement arrêté, par exemple si une nouvelle base de règles est chargée.</p> <p><b>Conseil :</b> Si vous utilisez ce mot clé pour attendre un serveur d'événements redémarré, vérifiez que l'intervalle est défini sur une durée supérieure celle nécessaire à l'arrêt et au redémarrage du serveur d'événements.</p> |
| <b>ServerLocation</b>      | Facultatif. Ce mot clé est ignoré si le mot clé <b>TransportList</b> est défini.                                                                                                                                                                                                                                                                                                                                                                                                                                           | <p>Spécifie l'hôte sur lequel le récepteur EIF est installé. Définissez la valeur de ce mot clé sur le nom d'hôte ou l'adresse IP. Si vous utilisez l'adresse IP, utilisez le format à points. La valeur par défaut est localhost.</p> <p>Vous pouvez définir au maximum 8 valeurs, séparées par des virgules (,). La première valeur définit le serveur d'événements principal et les valeurs supplémentaires sont traitées comme des serveurs de secours.</p> <p><b>Remarque :</b> Le mot clé <b>ServerLocation</b> définit le chemin d'accès et le nom du fichier de consignation des événements, et non le serveur d'événements, lorsqu'il est utilisé avec le mot clé <b>TestMode</b>.</p>                                                                                                                                                                                                             |
| <b>ServerPort</b>          | <p>Dépend du système d'exploitation sur lequel est installé le serveur d'événements :</p> <ul style="list-style-type: none"> <li>• <span style="background-color: #800000; color: white; padding: 2px;">UNIX</span> <span style="background-color: #800000; color: white; padding: 2px;">Linux</span><br/>Facultatif</li> <li>• <span style="background-color: #800000; color: white; padding: 2px;">Windows</span><br/>Obligatoire</li> </ul> <p>Ce mot clé est ignoré si le mot clé <b>TransportList</b> est défini.</p> | <p>Indique le numéro de port sur un adaptateur non-TME sur lequel le serveur d'événements écoute uniquement les événements.</p> <p>La valeur par défaut est 0, ce qui signifie que le numéro de port est extrait de l'associateur de port. Modifiez cette valeur uniquement si l'associateur de port n'est pas disponible sur le serveur d'événements.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

Tableau 8. Mots clés de configuration pour la fonction d'intégration d'événements (suite)

| Mot clé                                        | Obligatoire ou facultatif | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>TestMode</b>                                | Facultatif                | Active ou désactive le mode de test. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> <li>• YES : le mode de test est activé. Les événements sont consignés dans le fichier spécifié par le mot clé <b>ServerLocation</b>. Les événements ne sont pas envoyés au serveur d'événements.</li> <li>• NO (par défaut). Le mot de test est désactivé. Les événements sont envoyés au serveur d'événements.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Java <b>TraceFileName</b>                      | Facultatif                | Spécifie le nom du chemin d'accès complet et du fichier de trace. La valeur par défaut est <code>\$TIVOLIHOMTEC/eif.trc</code> . Si vous indiquez un chemin non valide, l'erreur suivante est renvoyée : <code>LOG0014E Unable to open the handler output file nom de fichier.</code><br><code>java.io.FileNotFoundException: nom de fichier (The system cannot find the path specified)</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Java <b>TraceLevel</b>                         | Facultatif                | Active le traçage. Pour générer les messages de trace, définissez ce mot clé sur ALL. Si vous définissez le mot clé sur une autre valeur, aucun message de trace n'est généré.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Linux Windows AIX Solaris <b>TransportList</b> | Facultatif                | Définit le nom des mécanismes de transfert. Spécifiez les mécanismes de transfert dans une liste de valeurs séparées par des virgules. Exemple : <code>TransportList=t1,t2,t3</code><br><br>Si un mécanisme de transfert échoue lorsqu'il est utilisé par une application émettrice, l'API se tourne vers les mécanismes de transfert qui sont spécifiés par ce mot clé, dans l'ordre où ils sont indiqués. Pour les applications réceptrices, l'API crée et utilise tous les mécanismes de transfert.<br><br>Ce mot clé est pris en charge uniquement pour les adaptateurs sur les systèmes d'exploitation qui sont spécifiés.<br><br>Si vous spécifiez ce mot clé, indiquez également le type de transfert et le canal de transfert du mécanisme de transfert. Pour spécifier ces informations, utilisez les mots clés <b>Type</b> et <b>Channels</b> . Pour plus d'informations sur ces mots clés, voir tableau 9, à la page 84. |

Le tableau suivant décrit les mots clés du type de transfert et du canal de transfert pour chaque mécanisme de transfert défini par le mot clé **TransportList**. Dans ce tableau, chaque mécanisme de transfert est représenté par **type\_name**.

Tableau 9. Mots clés pour les mécanismes de transfert définis par le mot clé **TransportList**

| En-tête                                                                                                                                                                                                                            | En-tête     | En-tête                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>type_nameType</b>                                                                                                                                                                                                               | Obligatoire | Indique le type de transfert d'un mécanisme de transfert défini par le mot clé <b>TransportList</b> . Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> <li>• SOCKET</li> <li>• SSL</li> </ul> Le serveur pour chaque mécanisme de transfert, tel qu'il est défini par <b>type_name</b> , est spécifié par le mot clé <b>ServerLocation</b> . Le numéro de port est spécifié par le mot clé <b>Port</b> .<br><br>Si vous définissez ce mot clé sur <b>SOCKET</b> , définissez également les mots clés dans tableau 10. Si vous définissez ce mot clé sur <b>SSL</b> , définissez également les mots clés dans tableau 11, à la page 85. |
| <b>type_nameChannels</b>                                                                                                                                                                                                           | Obligatoire | Indique les noms définis par l'utilisateur des canaux pour les mécanismes de transfert définis par le mot clé <b>TransportList</b> , dans une liste de valeurs séparées par des virgules. Exemple :<br>t1Channels=c1,c2,c3,c4<br><br>Pour chaque nom de canal, définissez également le mot clé <b>channel_namePort</b> .                                                                                                                                                                                                                                                                                                                                            |
| <b>channel_namePort</b>                                                                                                                                                                                                            | Obligatoire | Indique le numéro de port sur lequel le mécanisme de transfert écoute un canal, tel que défini par le mot clé <b>type_nameChannels</b> .<br><br>Si ce mot clé a pour valeur 0, cela signifie que l'associateur de port est utilisé.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <i>nom_canal</i> <b>ServerLocation=</b><br><i>serveur[région]</i>                                                                                                                                                                  |             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Indique le nom d'hôte ou l'adresse IP du serveur et de la région où le serveur des mécanismes de transfert se trouve pour le canal indiqué.<br><br>Le canal est défini par le mot clé <b>Channel</b> . Ce mot clé est obligatoire. |             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

Tableau 10. Mots clés supplémentaires à définir pour les types de transfert **SOCKET**

| Mot clé                             | Obligatoire ou facultatif | Description                                                                                                                                                                                                                            |
|-------------------------------------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>channel_namePortMapper</b>       | Facultatif                | Active ou désactive l'associateur de port. Les valeurs possibles sont les suivantes : <ul style="list-style-type: none"> <li>• YES : l'associateur de port est activé.</li> <li>• NO : l'associateur de port est désactivé.</li> </ul> |
| <b>channel_namePortMapperName</b>   | Facultatif                | Spécifie le nom de l'associateur de port, si ce dernier a été activé par la définition du mot clé <b>channel_namePortMapper</b> .                                                                                                      |
| <b>channel_namePortMapperNumber</b> | Facultatif                | Indique l'ID enregistré par l'appel de procédure distante (RPC).                                                                                                                                                                       |

Tableau 10. Mots clés supplémentaires à définir pour les types de transfert SOCKET (suite)

| Mot clé                              | Obligatoire ou facultatif | Description                                                                                                                           |
|--------------------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>channel_namePortMapperVersion</b> | Facultatif                | Spécifie la version de l'associateur de port, si ce dernier a été activé par la définition du mot clé <b>channel_namePortMapper</b> . |

Si le type de transfert pour un mécanisme de transfert, tel qu'il est défini par le mot clé **TransportList**, est défini sur SSL, définissez les mots clés décrits dans le tableau ci-après pour ce mécanisme de transfert.

Tableau 11. Mots clés supplémentaires à définir pour les types de transfert SSL

| Mot clé                                              | Obligatoire ou facultatif                                                     | Description                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------|-------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>channel_nameSSLKeystore</b>                       | Obligatoire                                                                   | Indique le chemin d'accès au fichier de clés contenant les clés et les certificats utilisés pour l'authentification SSL.                                                                                                                                                                                              |
| <b>channel_nameSSLKeystorePW</b>                     | Obligatoire                                                                   | Spécifie le mot de passe du fichier de clés qui est défini par le mot clé <b>channel_nameSSLKeystore</b> . Ce mot clé ou le mot clé <b>channel_nameSSLKeystoreStashFile</b> doit être utilisé.                                                                                                                        |
| <b>channel_nameSSLKeystoreStashFile</b>              | Obligatoire                                                                   | Spécifie le chemin d'accès au fichier de dissimulation qui contient le mot de passe du fichier de clés défini par le mot clé <b>channel_nameSSLKeystore</b> . Ce mot clé ou le mot clé <b>channel_nameSSLKeystorePW</b> doit être utilisé.                                                                            |
| <b>Java channel_nameSSLKeystoreEncryptionKeyFile</b> | Obligatoire si le mot clé <b>channel_nameSSLKeystoreStashFile</b> est défini. | Spécifie le chemin d'accès au fichier qui contient la clé de chiffrement utilisée pour décoder les données d'un fichier défini par le mot clé <b>channel_nameSSLKeystoreStashFile</b> .                                                                                                                               |
| <b>Java channel_nameSSLTruststore</b>                | Obligatoire                                                                   | Indique le chemin d'accès au fichier de clés contenant les clés et les certificats qui sont certifiés par Event Integration Facility pour l'authentification SSL. Si ce mot clé n'est pas défini, le fichier défini par le mot clé <b>channel_nameSSLKeystore</b> est utilisé en tant que fichier de clés certifiées. |
| <b>Java channel_nameSSLTruststorePW</b>              | Obligatoire                                                                   | Spécifie le mot de passe du fichier de clés qui est défini par le mot clé <b>channel_nameSSLTruststore</b> .                                                                                                                                                                                                          |
| <b>Java channel_nameSSLTruststoreStashFile</b>       | Obligatoire                                                                   | Spécifie le chemin d'accès au fichier de dissimulation qui contient le mot de passe du fichier de clés défini par le mot clé <b>channel_nameSSLTruststore</b> .                                                                                                                                                       |

Tableau 11. Mots clés supplémentaires à définir pour les types de transfert SSL (suite)

| Mot clé                                                              | Obligatoire ou facultatif                                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Java</p> <p><b>channel_nameSSLTruststoreEncryptionKeyFile</b></p> | Obligatoire si le mot clé <code>channel_nameSSLTruststoreStashFile</code> est spécifié. | Spécifie le chemin d'accès au fichier qui contient la clé de chiffrement utilisée pour décoder les données du fichier défini par le mot clé <b>channel_nameSSLTruststoreStashFile</b> .                                                                                                                                                                                                                                                                                                                                                               |
| <b>channel_nameSSLCipherList</b>                                     | Facultatif                                                                              | <p>Indique les chiffrements autorisés pour l'authentification SSL. Utilisez une liste de valeurs séparées par des virgules pour plusieurs chiffrements. Par exemple :</p> <pre>c1SSLCipherList=SSL_RC2_CBC_128_CBC_WITH_MD5,SSL_RC2_CBC_128_CBC_EXPORT40_WITH_MD5,SSL_DES_64_CBC_WITH_MD5</pre> <p>Si aucun chiffrement n'est indiqué, tous les chiffrements disponibles sont autorisés.</p>                                                                                                                                                          |
| <b>channel_nameSSLFIPSMODE</b>                                       | Facultatif                                                                              | <p>Contrôle si l'établissement de liaison SSL s'exécute en mode FIPS 140-2. Les valeurs possibles sont les suivantes :</p> <ul style="list-style-type: none"> <li>• ON : l'établissement de liaison est en mode FIPS 140-2.</li> <li>• OFF (par défaut) : l'établissement de liaison n'est pas en mode FIPS 140-2.</li> </ul> <p>Le mode FIPS 140-2 limite les types de chiffrement pouvant être utilisés.</p>                                                                                                                                        |
| <b>channel_nameSSLRequireClientAuthentication</b>                    |                                                                                         | <p>Indique si une application Event Integration Facility agissant comme un serveur nécessite un certificat d'un client durant un établissement de liaison SSL.</p> <p>Ce mot clé est valide uniquement dans la configuration SSL du serveur. Dans la configuration client, ce mot clé est ignoré.</p> <p>Les valeurs possibles sont les suivantes :</p> <ul style="list-style-type: none"> <li>• YES : un établissement de liaison SSL est obligatoire.</li> <li>• NO (par défaut) : aucun établissement de liaison SSL n'est obligatoire.</li> </ul> |

L'exemple suivant illustre 2 mécanismes de transfert définis par le mot clé **TransportList** et les mots clés associés pour les mots clés **type\_name** et **channel\_name**.

```
TransportList=t1,t2
t1Type=SSL
t1Channels=c1,c2
c1ServerLocation=sslhost1
c1Port=1111

c1_SSLKeystore=/my/location/gbkeys.kdb
c1_SSLKeystorePW=password
c1_SSLKeystoreStashFile=/my/location/gbkeys.sth
c1_SSLEncipherList=SSL_RSA_WITH_3DES_EDE_CBC_SHA

c1_SSLEIPSMODE=OFF

c2ServerLocation=sslhost2
c2Port=2222

c2_SSLKeystore=/my/other/location/gbkeys.kdb
c2_SSLKeystorePW=password
c2_SSLEncipherList=SSL_RSA_WITH_3DES_EDE_CBC_SHA
c2_SSLEIPSMODE=OFF

t2Type=SOCKET
t2Channels=c3,c4
c3ServerLocation=host1
c3Port=1234
c4ServerLocation=host2
c4Port=5678
```

#### Concepts associés:

«Fichiers de configuration», à la page 3

Utilisez le fichier de configuration pour contrôler le comportement des adaptateurs. Vous ne devez pas modifier plusieurs instances d'un adaptateur à exécuter dans différents environnements. Modifiez uniquement les fichiers de configuration.

«Options de connexion», à la page 15

Les options de connexion sont soit "connection-oriented" (orienté connexion), soit "connectionless" (sans connexion). Dans des situations où vous souhaitez envoyer plusieurs événements, utilisez l'option orienté connexion. Dans des situations où vous souhaitez envoyer peu d'événements au cours d'une certaine période, utilisez l'option sans connexion.

«Sélection des méthodes de distribution d'événements», à la page 35

Lors de la génération d'un adaptateur, vous devez également décider quelle méthode de distribution d'événement l'adaptateur utilise pour communiquer avec le serveur d'événements.

«Interface de programme d'application (API) des fichiers de configuration», à la page 42

La première tâche exécutée par les API est d'accéder aux informations des fichiers de configuration.

«Configuration d'adaptateurs pour des environnements internes», à la page 46

Le serveur d'événements peut recevoir des événements au codage UTF-8 ou au codage de l'hôte du serveur d'événements. Le serveur d'événements détermine automatiquement le type de codage (UTF-8 ou non-UTF-8) d'un événement en évaluant un indicateur spécifique dans les données d'événement.

«Journaux des messages», à la page 51

Dans des situations de résolution de problèmes, vous devez comprendre comment interpréter les messages et connaître les actions à exécuter pour résoudre un problème. Les fichiers journaux des messages vous permettent d'identifier et de résoudre les problèmes au sein de votre environnement.

«Journaux de trace», à la page 51

Les journaux de trace vous aident à déterminer pourquoi un problème se produit. Les journaux de trace capturent des informations sur l'environnement d'exploitation lorsque Tivoli Event Integration Facility ne fonctionne pas comme souhaité.

**Tâches associées:**

«Programmation de l'adaptateur», à la page 40

Pour programmer un adaptateur, implémentez les interfaces et les paramètres préférés pour le fichier de configuration. Décidez de définir les valeurs d'attribut dans le fichier de configuration ou dans le code d'adaptateur. Compilez ensuite et établissez une liaison avec l'adaptateur, le cas échéant.

«Filtrage à l'aide des fichiers de configuration», à la page 47

Normalement, un adaptateur envoie tous les événements au serveur d'événements. Vous pouvez facultativement répertorier les événements que l'adaptateur peut ou non envoyer au serveur d'événements à l'aide des mots clés Filter et FilterCache. De même, vous pouvez modifier le fichier de configuration pour filtrer les événements.

«Configuration d'une application réceptrice EIF pour SSL», à la page 36

Afin d'utiliser la communication SSL entre les applications réceptrices et client EIF, vous devez configurer l'application réceptrice. Les récepteurs requièrent un certificat personnel autosigné ou signé par une autorité de certification.

«Configuration d'une application client EIF pour SSL», à la page 38

Afin d'utiliser la communication SSL entre les applications réceptrices et client EIF, vous devez configurer l'application client EIF.

**Référence associée:**

«Envoi d'événements via des pare-feux», à la page 27

Vous pouvez envoyer des événements via des pare-feux en fonction des restrictions liées à votre environnement et à votre organisation concernant la sécurité par pare-feu.

«Réception d'événements pour les applications», à la page 26

Outre envoyer des événements, Tivoli Event Integration Facility permet à d'autres applications de recevoir, c'est-à-dire d'écouter des événements.

«Support SSL et FIPS 140-2», à la page 17

Tivoli Event Integration Facility prend en charge l'utilisation du protocole de chiffrement et d'authentification SSL (Secure Sockets Layer) pour envoyer et recevoir des événements. En outre, les connexions SSL d'EIF peuvent fonctionner en mode FIPS 140-2, ce qui implique l'utilisation de fournisseurs cryptographiques approuvés par la norme FIPS 140-2.

«Fichiers d'adaptateur», à la page 33

Outre le fichier d'en-tête ou le package Java, un certain nombre de fichiers sont connexes à l'adaptateur. Cette section répertorie et montre la relation entre ces fichiers et le traitement d'événement.

«Exécution d'adaptateurs générés avec l'interface de programme d'application (API) Java d'Event Integration Facility», à la page 45

L'API Java d'Event Integration Facility dépend d'autres classes pour exécuter ses tâches. Outre la configuration de l'environnement adéquat à l'aide des commandes **setup\_env**, vous devez ajouter le chemin au fichier exécutable Java dans la variable d'environnement de chemin d'accès à la bibliothèque.

---

## Remarques

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays.

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

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

IBM 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

Pour obtenir des informations sur les licences concernant les produits utilisant un jeu de caractères codé sur deux octets, contactez le service de propriété intellectuelle d'IBM de votre pays ou envoyez vos demandes par écrit à l'adresse suivante :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, 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" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAULT

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 éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

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

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

IBM Corporation  
958/NH04  
IBM Centre, St Leonards  
601 Pacific Hwy  
St Leonards, NSW, 2069  
Australia

IBM Corporation  
896471/H128B  
76 Upper Ground  
London SE1 9PZ  
United Kingdom

IBM Corporation  
JBF1/SOM1  
294 Route 100  
Somers, NY, 10589-0100  
United States of America

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

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

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

aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

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

#### LICENCE DE COPYRIGHT :

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

Des parties de ce produit contiennent du logiciel développé par Daniel Veillard.

- libxml2-2.7.8

Le logiciel libxml2-2.7.8 est distribué conformément au contrat de licence suivant :

© Copyright 1998-2003 Daniel Veillard.

All Rights Reserved. Toute personne possédant une copie de ce Logiciel et des fichiers de documentation associés (le «Logiciel») est autorisée gratuitement à exploiter le Logiciel sans restriction, y compris et sans limitation à utiliser, copier, modifier, fusionner, publier, distribuer, octroyer une sous-licence, et/ou vendre des copies du logiciel et à autoriser les personnes auxquelles le Logiciel est fourni à en faire de même, sous réserve des conditions suivantes :

Les déclarations relatives au copyright ci-dessus et cette déclaration de permission, doivent être incluses dans toutes les copies ou toute partie substantielle du Logiciel.

LE LOGICIEL EST FOURNI «EN L'ETAT» SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. DANIEL VEILLARD NE POURRA EN AUCUN CAS ÊTRE TENU RESPONSABLE POUR TOUT DOMMAGE, QUELLES QU'EN SOIT LA CAUSE ET LES RESPONSABILITES, CONTRACTUELLES OU NON, OU PAR FAUTE LIEE A L'UTILISATION DE CE LOGICIEL.

Sauf indication contraire dans cette notice, le nom de Daniel Veillard ne doit pas être utilisé à des fins de publicité ou de promotion de ce Logiciel sans autorisation écrite préalable de Daniel Veillard.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

---

## Marques

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

AIX  
IBM  
Netcool  
OS/390  
Passport Advantage  
System z  
Tivoli  
Tivoli Enterprise  
Console  
TME  
z/OS  
zSeries

Adobe, Acrobat, Portable Document Format (PDF), PostScript ainsi que toutes les marques incluant Adobe sont des marques d'Adobe Systems Incorporated aux Etats-Unis et/ou dans certains autres pays.

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

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

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque de The Open Group aux Etats-Unis et/ou dans certains autres pays.

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

# Index

## A

- accessibilité ix
- adaptateurs
  - API C 59
  - API Java 71
- classes d'événement 34
- compilation 40
- configuration 44
- considérations Windows 43
- exemple de code 40
- fichiers 33
- génération 40
  - présentation 33
- généré avec l'API C
  - compilation 43
  - liaison à des bibliothèques 44
- identification des événements 34
- installation 44
- internationalisation 46
- méthodes de distribution d'événement 35
- mise à niveau 42
- multiprocessus 40
- présentation 2
- test 44

## API

### C

- tec\_agent\_getenv 59
- tec\_agent\_init 60
- tec\_create{EIF\_handle 60
- tec\_create\_handle 61
- tec\_create\_handle\_c 62
- tec\_create\_handle\_r 63
- tec\_destroy\_handle 64
- tec\_errno 64
- tec\_get\_event 64
- tec\_put\_event 65
- tec\_register\_callback 66

- communications 42

- fichiers de configuration 42

### Java

- disconnect 71
- disconnect(time) 71
- getConfigVal 71
- onMessage 72
- receiveEvent 72
- registerListener 73
- sendEvent 73
- TECAgent 74
- TECEvent 74

- présentation 59

## API C

- configuration de l'application
  - réceptrice 31
  - utilitaires 67

## API Java

- présentation 71

## attributs

- présentation 1

## B

- bibliothèque libeif.a 16

- bibliothèques 64 bits 1

## C

- cache 27

- activation 28

### cartes

- fichiers d'API Java obligatoires 45

- centre de documentation de Tivoli vii

- certificat 17

- chiffrements 17

- classes d'événement 34

- commande de publications vii

- commandes d'interface de ligne de

- commande 21

### communications

- API 42

### connexions

- distribution orientée connexion 15

- distribution sans connexion 15

- conventions, typographiques x

- conventions typographiques x

## D

### démon

- associateur de port 30

- démon associateur de port 30

- détermination des problèmes

- identification et résolution des

- problèmes 51

- disconnect 71

- disconnect(time) 71

- disponibilité

- événements 53

- distribution d'événement

- options de connexion 15

- options de transfert 16

- sélection de méthodes 35

- documentation en ligne vii

## E

- échecs système 27

- ed\_scan\_get\_n 67

- ed\_scan\_n 68

- ed\_sleep 68

- événements 48

- classes 3

- corrélation d'état 47

- définition de classes 34

- disponibilité 53

- distribution lors d'échecs système 27

- exemple 4

- filtrage 6, 47

- flux 33

- identification 4, 34

- événements (*suite*)

- méthodes de distribution 15

- paramètres de connexion pour la

- réception 54

- pare-feux 27

- réception

- exemple 26

- surveillance 34

- translaté 1

## Event Integration Facility

- les nouveautés

- bibliothèques 64 bits 1

- FIPS 140-2 1

- prise en charge d'IPv6 1

- SSL 1

- types de transfert 1

- présentation 1

- expressions régulières

- filtrage 49

- expressions régulières Tcl

- filtrage des événements 49

## F

- fichier de clés 17

- fichier de clés certifiées 17

- fichier de configuration

- mots clés 77

- fichier de dissimulation 17

- fichiers

- root.baroc 3

- fichiers BAROC 3

- fichiers de configuration

- API 42

- filtrage des événements 47

- filtrage

- corrélations d'état 47

- expressions régulières 49

- fichiers de configuration 47

- filtrage des événements

- échecs système 48

- filtrage lors d'un échec système 48

- filtres

- présentation du filtrage

- d'événements 6

- FIPS 140-2 1, 17

- fonctions

- disconnect 71

- disconnect(time) 71

- getConfigVal 71

- onMessage 72

- receiveEvent 72

- registerListener 73

- sendEvent 73

- tec\_agent\_getenv 59

- tec\_agent\_init 60

- tec\_create{EIF\_handle 60

- tec\_create\_handle 61

- tec\_create\_handle\_c 62

- tec\_create\_handle\_r 63

- tec\_destroy\_handle 64

fonctions (*suite*)  
  tec\_errno 64  
  tec\_get\_event 64  
  tec\_put\_event 65  
  tec\_register\_callback 66  
  TECAgent 74  
  TECEvent 74  
formation  
  voir formation technique Tivoli x  
formation, technique Tivoli x  
formation technique Tivoli x

## G

getConfigVal 71

## I

identification et résolution des problèmes  
  erreur de connexion lors de  
    l'utilisation de posteifmsg 55  
  erreur liée aux événements envoyés  
    généralités 55  
    utilisation de posteifmsg 55  
  génération et exécution  
    d'adaptateurs 54  
  journaux de trace 51  
  journaux des messages 51  
  paramètres de connexion pour le  
    processus de réception 54  
  présentation 51  
  scénarios usuels 54  
iKeyman 16, 17  
informations de support x  
installation 10

## J

journaux  
  messages 51  
  trace 51  
journaux de trace 51  
journaux des messages 51

## M

manuels vii  
mots clés  
  fichier de configuration 77  
  fichiers de configuration 3  
  liste 77

## O

onMessage 72

## P

paramètres de connexion  
  réception d'événements 54  
pare-feux  
  envoi d'événements 27  
performances  
  prévention de surcharge 53

prise en charge d'IPv6 1  
public vii  
publications vii

## R

receiveEvent 72  
récupération  
  systèmes 27  
registerListener 73  
règles 7

## S

sauvegarde  
  configuration de serveurs 29  
sendEvent 73  
serveurs d'événements 4  
SOCKET 16  
SSL 1, 16, 17  
structure de répertoire 10  
surveillance  
  événements 34

## T

tec\_agent\_getenv 59  
tec\_agent\_init 60  
tec\_create{EIF\_handle 60  
tec\_create\_handle 61  
tec\_create\_handle\_c 62  
tec\_create\_handle\_r 63  
tec\_destroy\_handle 64  
tec\_errno 64  
tec\_get\_event 64  
tec\_put\_event 65  
tec\_register\_callback 66  
TECAgent 74  
TECEvent 74  
transfert  
  options 16  
transfert des événements  
  présentation 15  
types de transfert 1

## U

utilitaires  
  API C 67  
  ed\_scan\_get\_n 67  
  ed\_scan\_n 68  
  ed\_sleep 68  
utilitaires de l'API  
  C  
    ed\_scan\_get\_n 67  
    ed\_scan\_n 68  
    ed\_sleep 68

## V

variables, notation pour x  
variables d'environnement, notation x

## W

Windows  
  considérations spécifiques 43





SC43-0828-00

