

Tivoli Netcool/OMNibus
Version 8.1

Guide des sondes et passerelles



Tivoli Netcool/OMNibus
Version 8.1

Guide des sondes et passerelles



Important

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section «Remarques», à la page 371.

Deuxième édition - Novembre 2014

Réf. US : SC27-6266-01

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 1994, 2014.

Table des matières

Avis aux lecteurs canadiens vii

A propos de cette publication ix

Utilisateurs concernés	ix
Publications	ix
Accessibilité	xi
Formation technique Tivoli	xi
Informations de support	xi
Conventions utilisées dans la présente publication	xi

Chapitre 1. A propos des sondes 1

Enregistrement des sondes.	2
Types de sondes	3
Sondes de périphériques	3
Sondes de fichier journal	3
Sondes de base de données	4
Sondes API	4
Sondes CORBA	4
Sondes diverses	5
Composants de sonde	5
Fichier exécutable.	5
Fichier de propriétés.	6
Fichier de règles	8
Conventions de dénomination des fichiers de composant de sonde.	9
Architecture de sonde	10
Génération des identificateurs uniques pour les événements	11
Modes de fonctionnement des sondes	12
Mode stocker-et-transmettre pour les sondes	12
Mode de capture brute pour les sondes	16
Mode sécurisé pour les sondes	17
Mode de reprise en ligne d'égal à égal pour les sondes	17

Chapitre 2. Fichiers de règles des sondes. 21

Développement de fichiers de règles	21
Éléments, zones, propriétés et tableaux dans les fichiers de règles	22
Instructions de contrôle dans les fichiers de règles	26
Fonctions et opérateurs de fichier de règles.	33
Exemples de fichier de règles	62
Test de fichiers de règles	64
Débogage de fichiers de règles	65
Relecture du fichier de règles	66
Configurer une sonde pour le téléchargement de fichiers de règles éloignés	68
Activation de la mise en cache des fichiers de règles de sonde	69
Intégration de plusieurs fichiers de règles dans un fichier de règles	70
Probe Rules Utility	72

Variables d'environnement et échappement de caractères	76
--	----

Chapitre 3. Personnalisations du fichier de règles de sonde 77

Détection de débordements d'événements et de taux d'événements anormaux	77
Configuration des sondes pour détecter des débordements d'événements et des débits d'événements anormaux	78
Protection du serveur ObjectServer contre les afflux d'événements	80
Fichier de règles de configuration du débordement	82
Fichier de règles de débordement	85
Activation de l'auto-surveillance des sondes	87
Configuration de l'auto-surveillance des sondes	88
Fichiers de configuration Tivoli Netcool/OMNIbus pour l'auto-surveillance des sondes	90
Configuration des sondes pour l'auto-surveillance	91

Chapitre 4. Exécution des sondes 97

Utilisation des variables d'environnement OMNIHOME et NCHOME	98
Exécution de sondes sous UNIX	98
Exécution de sondes sous SUID root	99
Exécution de sondes sous Windows	101
Exécution d'une sonde en tant qu'application de console	101
Exécution d'une sonde en tant que service.	102

Chapitre 5. Administration à distance des sondes 105

Activation de l'administration à distance des sondes.	105
Configuration de l'authentification entre les systèmes distants et les sondes	107
Configuration des connexions SSL entre les systèmes distants et les sondes	108
Envoi de demandes à distance à des sondes (nco_http)	109
Rechargement de fichiers de règles (nco_probereloadrules)	114
Envoi de mises à jour de propriété à des sondes (nco_setprobeprop)	115
Génération d'événements à l'aide de sondes (nco_probeeventfactory)	116
A propos de l'identificateur URI commun	117
Extraire l'état en cours d'une sonde	118
Recharger le fichier de règles	119
Répertoire les propriétés de sonde	120
Créer un événement synthétique	121
Définir une propriété de sonde	123

Événement d'accusé de réception et données utiles	124
Définir les demandes PATCH ou POST comme bloquantes ou non bloquantes	127

Chapitre 6. Propriétés et options de ligne de commande de sonde communes 129

Chapitre 7. Netcool MIB Manager . . . 141

Démarrage de MIB Manager	141
Options de ligne de commande de MIB Manager	142
Utilisation de Netcool MIB Manager	146
Vue Modules MIB	148
Vue Arborecence d'ID objet	150
Importation de données MIB	152
Exportation des données MIB	154
Édition des messages d'alerte SNMP	157
Génération d'interceptions SNMP	158
Création et suppression d'unités	159
Configuration des préférences globales	160
Définition des préférences de répertoire	160
Définition des préférences d'exportation	161
Définition des préférences générales	161
Définition de préférences de journalisation	162
Définition des préférences de recherche	163
A propos de SNMP	164
Concepts et conception de MIB	165
Types d'objet MIB	168
Formats d'objet MIB valides	171

Chapitre 8. A propos des passerelles 173

Types de passerelles	174
Passerelles ObjectServer unidirectionnelles	175
Passerelles ObjectServer bidirectionnelles	176
Base de données, centre d'assistance et autres passerelles	176
Composants de passerelle	176
Passerelles unidirectionnelles	177
Passerelles bidirectionnelles	177
Mode stocker-et-transmettre pour les passerelles	179
Mode sécurisé pour les passerelles	180
Programmes d'écriture et reprise par restauration de la passerelle	181

Chapitre 9. Configuration de passerelles 183

Utilisation de fichiers de configuration multiples	183
Fichier de définition de mappe	184
Fichier de définition de répllication de table	187
Fichier de commandes de démarrage	190
Propriétés et options de ligne de commande de passerelle communes	192
Utilisation de commandes sur des passerelles en cours d'exécution	196
Utilisation de fichiers de configuration uniques	197
Configuration du programme de lecture	198
Configuration du programme d'écriture	198

Configuration de route	199
Configuration de mappage	199
Configuration de filtre	200
Options de ligne de commande de passerelle communes	201
Configuration de passerelles en cours d'exécution	203
Commandes de passerelle	205
Création de tables de conversion	217

Chapitre 10. Exécution des passerelles 219

Utilisation des variables d'environnement OMNIHOME et NCHOME	219
Exécution des passerelles	219
Traitement des incidents de passerelle	220

Chapitre 11. Configuration des passerelles ObjectServer 221

Passerelles ObjectServer bidirectionnelles	221
Passerelles ObjectServer unidirectionnelles	223
Configuration des propriétés de passerelles ObjectServer	224
Contrôle de la taille de la mémoire cache	225
Définition du niveau des messages de débogage	225
Vérifier si la passerelle s'exécute sous le contrôle de processus	226
Modification de la taille de mémoire tampon	227
Modification du mécanisme d'authentification	227
Connexion à des serveurs ObjectServer sécurisés	228
Resynchronisation des données de sécurité avec des serveurs ObjectServer sécurisés	228
Configuration des opérations de reprise par restauration	229
Configuration des opérations de stockage et de retransmission	232
Configuration de la resynchronisation	233
Propriétés de passerelles ObjectServer génériques	238
Propriétés des passerelles unidirectionnelles	242
Propriétés des passerelles bidirectionnelles	249
Exécution des passerelles ObjectServer	254
Mappage de passerelles ObjectServer	255
Mappage des attributs	256
Exemple de mappage	257
Autres commandes d'exécution de passerelle	262
GET CONFIG	263
SHOW PROPS	263
FAILOVER SYNCH	263
SET LOG LEVEL	264
Fichier de définition de répllication de table	264
Effets de la transmission des suppressions sur la taille de mémoire	267
Exemple de fichier de définition de répllication de table	268

Chapitre 12. Mappage de passerelles ObjectServer. 269

Mappage des attributs	270
Exemple de mappage	271

Chapitre 13. Configuration des propriétés de passerelles ObjectServer.	277
Contrôle de la taille de la mémoire cache	277
Définition du niveau des messages de débogage	278
Vérifier si la passerelle s'exécute sous le contrôle de processus.	279
Modification de la taille de mémoire tampon.	279
Modification du mécanisme d'authentification	280
Connexion à des serveurs ObjectServer sécurisés	281
Resynchronisation des données de sécurité avec des serveurs ObjectServer sécurisés	281
Configuration des opérations de reprise par restauration	281
Configuration des opérations de reprise par restauration pour les passerelles ObjectServer unidirectionnelles	282
Configuration des opérations de reprise par restauration pour les passerelles ObjectServer bidirectionnelles	283
Configuration des opérations de stockage et de retransmission	285
Configuration des opérations de stockage et de retransmission pour les passerelles ObjectServer unidirectionnelles	285
Configuration des opérations de stockage et de retransmission pour les passerelles ObjectServer bidirectionnelles	285
Configuration de la resynchronisation	286
Configuration de la resynchronisation pour les passerelles ObjectServer unidirectionnelles.	286
Configuration de la resynchronisation pour les passerelles ObjectServer bidirectionnelles	288
Propriétés de passerelles ObjectServer génériques	291
Propriétés des passerelles unidirectionnelles	295
Propriétés des passerelles bidirectionnelles	303
Chapitre 14. Autres commandes d'exécution de passerelle	309
GET CONFIG	309
SHOW PROPS	310
FAILOVER SYNCH	310
SET LOG LEVEL	310
Chapitre 15. Fichier de définition de réplication de table	313
Effets de la transmission des suppressions sur la taille de mémoire	315

Exemple de fichier de définition de réplication de table	316
--	-----

Annexe A. Messages d'erreur des sondes et techniques d'identification et de résolution des problèmes . . . **319**

Messages d'erreur génériques	319
Messages de niveau fatal	319
Messages de niveau erreur	320
Messages de niveau avertissement	323
Messages de niveau information	323
Messages de niveau débogage.	324
Messages ProbeWatch et TSMWatch	325
Identification et résolution des problèmes de sonde	327
Causes de problèmes communes	327
Que faire si	328
Vérifier la synchronisation d'un fichier de règles	332

Annexe B. Messages d'erreur de passerelle communs . . . **335**

Annexe C. Expressions régulières . . **345**

Bibliothèque d'expressions régulières NETCOOL	345
Bibliothèque d'expressions régulières TRE	347
Métacaractères	348
Quantificateur minimal ou non gourmand.	349
Expressions entre crochets	350
Constructions pour le support multiculturel	352
Séquences d'échappement avec barres obliques inversées	353

Annexe D. Tables et types de données ObjectServer. . . . **355**

Table alerts.status	355
Table alerts.details	364
Table alerts.journal	365
Table service.status	366
Table registry.probes	366
Types de données ObjectServer	368

Remarques . . . **371**

Marques	374
---------	-----

Index . . . **375**

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

Tivoli Netcool/OMNIBus est un système de gestion du niveau de service (SLM) qui offre une surveillance centralisée et en temps réel des réseaux complexes et des domaines informatiques.

Le *Guide des sondes et des passerelles d'IBM Tivoli Netcool/OMNIBus* contient des informations de présentation et de référence sur les analyses, notamment la syntaxe des fichiers de règles d'analyse, les propriétés et les options des lignes de commande, les messages d'erreur, et les techniques d'identification et de traitement des incidents. Cette publication contient également des informations de présentation et de référence sur les passerelles, notamment les commandes de passerelles, les options de lignes de commande et les messages d'erreur.

Utilisateurs concernés

Cette publication concerne les utilisateurs et les administrateurs qui ont besoin de configurer des sondes et des passerelles.

Les sondes et les passerelles font partie de Tivoli Netcool/OMNIBus, et vous êtes supposés connaître le fonctionnement de Tivoli Netcool/OMNIBus.

Publications

Cette section liste les publications dans la bibliothèque Tivoli Netcool/OMNIBus et les documents associés. Cette section décrit également comment accéder aux publications Tivoli en ligne et comment commander 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*,
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*,
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*,
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*,
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*,
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*,
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* ,
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*,
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*,
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*,
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*,
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.

Accès à la terminologie en ligne

Le site Web de terminologie IBM consolide la terminologie à partir des bibliothèques de logiciels IBM à un emplacement pratique. Vous pouvez accéder au site Web de terminologie à l'adresse 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 des téléchargements Tivoli à l'adresse suivante :

<ftp://public.dhe.ibm.com/software/tivoli/Netcool/NetcoolOmnibus/library/>

Remarque : Si vous imprimez les documents PDF sur du papier qui n'est pas au format lettre, configurez l'option dans la fenêtre **Fichier > Imprimer** qui permet à

Adobe Reader d'imprimer des pages au format lettre sur le papier que vous utilisez.

Accessibilité

Les fonctions d'accessibilité permettent aux utilisateurs souffrant d'un handicap physique, tel qu'une mobilité réduite ou une vision limitée, d'utiliser avec succès les produits logiciels.

Ce produit vous permet d'utiliser des technologies d'assistance pour écouter et naviguer dans l'interface. Vous pouvez également utiliser le clavier à la place de la souris pour faire fonctionner certaines fonctionnalités de l'interface graphique.

Formation technique Tivoli

Pour des informations sur la formation technique Tivoli, consultez le site Web IBM Tivoli Education :

<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 la présente publication

Cette publication utilise plusieurs conventions pour les dispositions particulières et les actions spéciales, pour les commandes et les chemins d'accès liés au système d'exploitation.

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.

Emplacement d'origine de Netcool

L'emplacement d'origine de Netcool est le répertoire de base dans lequel Tivoli Netcool/OMNIBus est installé. L'emplacement d'origine de Netcool est défini par la variable d'environnement NCHOME, dont la valeur est la suivante :

- **UNIX** **Linux** \$NCHOME a pour valeur par défaut /opt/IBM/tivoli/netcool
- **Windows** %NCHOME% a pour valeur par défaut C:\IBM\Tivoli\Netcool

Lorsqu'un répertoire ou un chemin de commande commence par la variable NCHOME, l'information s'applique à tous les systèmes d'exploitation pris en charge.

Les autres produits qui utilisent la variable d'environnement NCHOME, tels que IBM Tivoli Network Manager IP Edition, peuvent être installés dans l'emplacement d'origine de Netcool. Chaque produit installe ses composants et fichiers spécifiques dans un sous-répertoire de produits dédié dans l'emplacement d'origine de Netcool. Les fichiers qui sont communs à tous les produits sont installés dans des sous-répertoires partagés dans l'emplacement d'origine de Netcool.

Noms de répertoire spécifiques au système d'exploitation

Lorsque les fichiers Tivoli Netcool/OMNIBus sont identifiés comme se trouvant dans un répertoire *arch* sous NCHOME, *arch* est une variable qui représente le répertoire de votre système d'exploitation, comme indiqué dans le tableau suivant.

Tableau 1. Noms de répertoire pour la variable arch

Nom de répertoire représenté par arch	Système d'exploitation
aix5	Systèmes AIX
hpux11hpa	Systèmes HP-UX Itanium
linux2x86	Systèmes Red Hat Linux et SUSE
linux2s390	Linux for System z
solaris2	Systèmes Solaris
win32	Systèmes Windows

Emplacement OMNIHOME

Les sondes et les passerelles, ainsi que les anciennes versions de Tivoli Netcool/OMNIbus, utilisent la variable d'environnement OMNIHOME dans de nombreux fichiers de configuration. Définissez la valeur d'OMNIHOME de la manière suivante :

- **UNIX** **Linux** Définissez \$OMNIHOME avec la valeur \$NCHOME/omnibus
- **Windows** Définissez %OMNIHOME% avec la valeur %NCHOME%\omnibus

Répertoire de base de interface graphique Web

REP_INSTALL_WEBGUI

Fait référence au répertoire où l'interface graphique Web est installée. Ce répertoire est connu comme le répertoire de base de l'interface graphique Web. Les valeurs par défaut sont les suivantes :

UNIX **Linux** /opt/IBM/tivoli/netcool/omnibus_webgui

Windows C:\IBM\tivoli\netcool\omnibus_webgui

Le répertoire de base de l'interface graphique Web est distinct des répertoires de base de Jazz for Service Management.

Répertoire de base de Jazz for Service Management

REP_INSTALL_WAS

Fait référence à l'emplacement où WebSphere Application Server est installé. Cet emplacement peut être spécifié lors de l'installation. Les valeurs par défaut sont les suivantes :

UNIX **Linux** /opt/IBM/WebSphere/AppServer

Windows C:\Program Files\IBM\WebSphere\AppServer

REP_INSTALL_JazzSM

Fait référence à l'emplacement où Jazz for Service Management est installé. Cet emplacement peut être spécifié lors de l'installation. Les valeurs par défaut sont les suivantes :

UNIX **Linux** /opt/IBM/JazzSM

Windows C:\Program Files\IBM\JazzSM

REP_INSTALL_JazzSM

Fait référence à l'emplacement du profil de serveur d'applications utilisé pour Jazz for Service Management. Cet emplacement se trouve dans le sous-répertoire /profile du répertoire de base de Jazz for Service Management.

UNIX **Linux** *REP_INSTALL_JazzSM*/profile

Windows *REP_INSTALL_JazzSM*\profile

REP_INSTALL_DASH

Fait référence à l'emplacement où Concentrateur des services d'application du tableau de bord est installé. Cet emplacement peut être spécifié lors de l'installation. Les valeurs par défaut sont les suivantes :

UNIX **Linux** /opt/IBM/JazzSM/ui

















Windows C:\Program Files\IBM\JazzSM\ui

Pour les autres répertoires d'installation de Jazz for Service Management, consultez le centre de documentation de Jazz for Service Management à <http://www-01.ibm.com/support/knowledgecenter/SSEKCU/welcome>.

Conventions relatives aux graphiques de repérage

La documentation de Tivoli Netcool/OMNIbus contient des graphiques de repérage qui indiquent les parties d'une rubrique ou d'une instruction qui ne s'appliquent que dans certaines conditions. Le tableau suivant décrit la signification de chacun de ces graphiques :

Tableau 2. Graphiques de repérage pour Tivoli Netcool/OMNIbus.

Graphique	Description
	Le texte ou l'instruction ne s'applique qu'au composant de l'interface graphique Web.
	Le texte ou l'instruction ne s'applique qu'aux systèmes d'exploitation UNIX, ce qui inclut AIX et Solaris. Les graphiques de repérage suivants sont utilisés pour AIX et Solaris lorsque cela est nécessaire :  
	Le texte ou l'instruction ne s'applique qu'aux systèmes d'exploitation Linux.
	Le texte ou l'instruction ne s'applique qu'aux systèmes d'exploitation Windows.
	Le texte ou l'instruction ne s'applique qu'aux systèmes d'exploitation 32 bits.
	Le texte ou l'instruction ne s'applique qu'aux systèmes d'exploitation 64 bits.
	Le texte ou l'instruction ne s'applique qu'à l'utilisation ou à la configuration du chiffrement FIPS 140-2.
	Le texte ou l'instruction décrit le comportement par défaut ou ne s'applique qu'aux configurations par défaut.
	Le texte ou l'instruction ne s'applique qu'au numéro de groupe de correctifs indiqué par le graphique. Les fonctionnalités ou les améliorations décrites dans le texte ne sont disponibles qu'après l'installation de ce groupe de correctifs. Remarque : Les groupes de correctifs pour le composant serveur et le composant de l'interface graphique Web sont publiés séparément. Les groupes de correctifs de l'interface graphique Web sont indiqués comme suit :  
	Le texte ou l'instruction ne s'applique qu'aux administrateurs de l'interface graphique Web. C'est-à-dire aux utilisateurs qui disposent des rôles ncw_user et ncw_admin.
	Le texte ou l'instruction ne s'applique qu'au langage de programmation C.
	Le texte ou l'instruction ne s'applique qu'au langage de programmation Java™.

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

Chapitre 1. A propos des sondes

Les sondes et les gestionnaires TSM (Telco Service Manager) se connectent à une source d'événement, détectent et acquièrent des données d'événement et les transmettent au serveur ObjectServer sous forme d'alertes. Les gestionnaires TSM fonctionnent de la même manière que les sondes mais comportent des fonctions supplémentaires. Les sondes utilisent la logique spécifiée dans un fichier de règles pour manipuler les éléments d'événement avant de les convertir en zones d'une alerte dans la table alerts.status du serveur ObjectServer.

La figure suivante montre l'intégration des sondes dans une architecture Tivoli Netcool/OMNIBus.

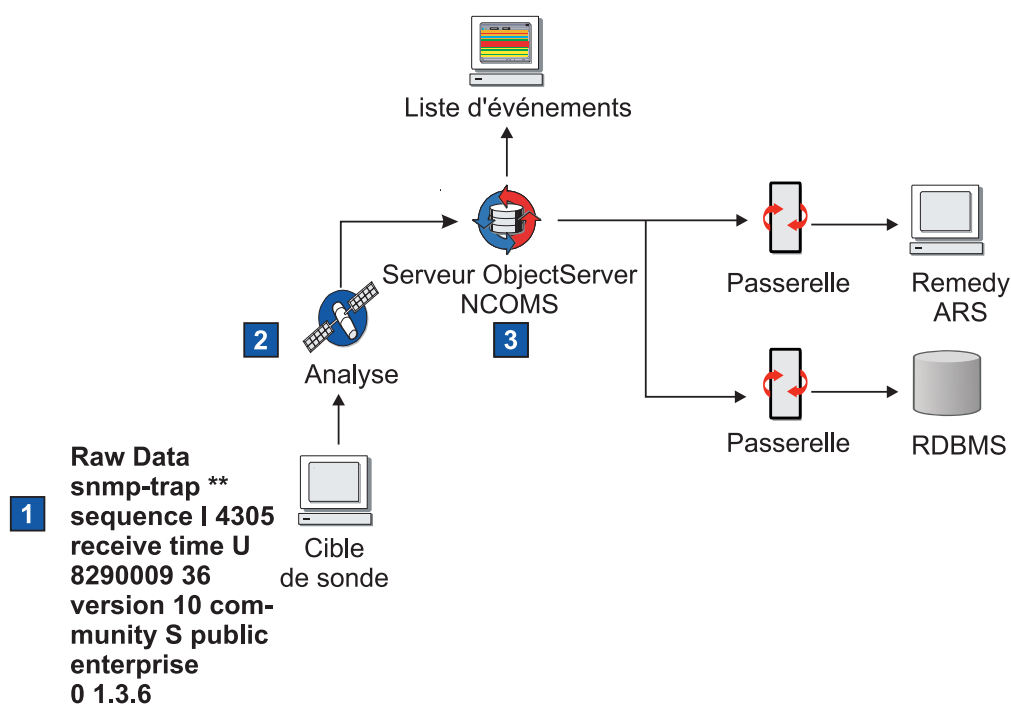


Figure 1. Traitement d'événement dans Tivoli Netcool/OMNIBus

Le flux des données d'événement est le suivant :

- 1** Les données d'événement sont générées par la cible de la sonde.
- 2** La sonde segmente les données d'événement, ajoute des informations supplémentaires à l'événement et affecte les valeurs aux zones de la table ObjectServer alerts.status. Elle transmet ensuite les données traitées au serveur ObjectServer sous forme d'alerte.
- 3** Le serveur ObjectServer stocke et gère les alertes, qui peuvent être affichées dans la liste d'événements et facultativement transmises à une ou plusieurs passerelles.

Remarque : Les informations de cette publication sont générales et communes à toutes les sondes. Pour obtenir des informations spécifiques aux sondes, consultez les publications sur les sondes individuelles dans le centre de documentation IBM Tivoli Network Management :

1. Accédez à <http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp>.
2. Développez le noeud *IBM Tivoli Netcool/OMNIBus* dans le panneau de navigation située sur la gauche.
3. Développez le noeud *Tivoli Netcool/OMNIBus probes and TSMs*.
4. Recherchez la publication appropriée.

Concepts associés:

«Types de sondes», à la page 3

Chaque sonde est conçue de manière unique pour acquérir des données d'événement à partir d'une source spécifique. Toutefois, les sondes peuvent être catégorisées en fonction de la manière dont elles acquièrent les événements.

Enregistrement des sondes

Lorsqu'une sonde se connecte au serveur ObjectServer, elle enregistre des informations la concernant dans la table registry.probes.

La table registry.probes est utilisée pour effectuer le suivi des informations d'exécution dynamiques relatives aux sondes. La sonde détermine les données qui sont entrées dans la table.

Si vous possédez plusieurs instances d'une sonde s'exécutant sur un ordinateur et si chaque instance possède le même nom, seule une instance sera enregistrée dans la table registry.probes. Pour permettre l'enregistrement de toutes les instances d'une sonde s'exécutant sur le même ordinateur, vous devez utiliser des valeurs uniques pour la propriété **Name** de chaque sonde.

Si une sonde ne parvient pas à redémarrer après s'être précédemment connectée à l'ObjectServer, son statut n'est pas mis à jour dans la table registry.probes. La table continue de représenter le statut de la sonde au moment de sa dernière connexion à l'ObjectServer.

Remarque : Lorsqu'une sonde est connectée au serveur ObjectServer via un serveur proxy, l'ID connexion de la sonde peut varier dans le temps et risque donc d'être enregistré incorrectement. Cela s'explique par le fait que le serveur proxy optimise ses connexions au serveur ObjectServer et déplace les connexions de sonde de façon dynamique. Toutefois, l'ID connexion stocké dans la table registry.probes reste le même. Il n'est pas mis à jour lorsqu'une sonde est déplacée vers une autre connexion sur le même serveur proxy.

Si vous utilisez la commande `kill -9` pour arrêter un processus de sonde et que la sonde est connectée via un serveur proxy, les données de sonde existantes dans la table registry.probes sont conservées et ne sont pas mises à jour au redémarrage de la sonde. Ce problème ne se pose pas lorsqu'un processus de sonde est arrêté par la commande `kill` sans l'option `-9`.

Référence associée:

«Table registry.probes», à la page 366

La table registry.probes est utilisée pour effectuer le suivi des informations d'exécution dynamiques relatives aux sondes. Lorsqu'une sonde se connecte au serveur ObjectServer, elle enregistre des informations la concernant dans la table registry.probes. La sonde contrôle les données entrées dans la table.

Types de sondes

Chaque sonde est conçue de manière unique pour acquérir des données d'événement à partir d'une source spécifique. Toutefois, les sondes peuvent être catégorisées en fonction de la manière dont elles acquièrent les événements.

Les types de sondes sont :

- Périphérique
- Fichier journal
- Base de données
- API
- CORBA
- Diverses

Le type de sonde est déterminé par la méthode avec laquelle la sonde détecte des événements. Par exemple, la sonde pour Agile ATM Switch Management détecte les événements produits par un périphérique (un commutateur ATM), mais elle acquiert des événements à partir d'un fichier journal et pas directement du commutateur. Par conséquent, cette sonde est classée comme sonde de fichier journal et non pas comme sonde de périphérique. De même, la sonde pour Oracle obtient des données d'événement d'une table de base de données et elle est donc classée comme sonde de base de données.

Sondes de périphériques

Une sonde de périphérique acquiert des événements en se connectant à un périphérique distant, comme un commutateur ATM.

Les sondes de périphérique sont souvent exécutées sur une machine différente de celle qu'elles sondent et se connectent à la machine cible via une liaison réseau, un modem ou un câble physique. Certaines sondes de périphérique peuvent utiliser plusieurs méthodes pour se connecter à la machine cible.

Après s'être connectée à la machine cible, la sonde détecte des événements et les transmet au serveur ObjectServer. Certaines sondes de périphérique sont passives et attendent de détecter un événement avant de le transmettre au serveur ObjectServer. D'autres sondes de périphérique sont plus actives et émettent des commandes sur le périphérique cible afin d'acquérir des événements.

Sondes de fichier journal

Une sonde de fichier journal acquiert des événements en lisant un fichier journal créé par le système cible.

Par exemple, la sonde pour Heroix RoboMon Element Manager lit le fichier d'événements Heroix RoboMon Element Manager.

La majorité des sondes de fichier journal est exécutée sur la machine sur laquelle se trouve le fichier journal ; il ne s'agit pas nécessairement de la même machine que celle du système cible. Le système cible ajoute des événements au fichier journal. La sonde ouvre régulièrement le fichier journal, acquiert et traite les événements qu'il stocke puis transmet les événements pertinents au serveur ObjectServer sous forme d'alertes. Vous pouvez configurer la fréquence à laquelle la sonde vérifie le fichier journal à la recherche de nouveaux événements et comment ces derniers sont traités.

Sondes de base de données

Une sonde de base de données acquiert des événements à partir d'une seule table de base de données : la table *source*. En fonction de la configuration, toute modification (insertion, mise à jour ou suppression) d'une ligne de la table source peut produire un événement.

Par exemple, la sonde pour Oracle acquiert les données des transactions consignées dans une table de base de données Oracle.

Au démarrage d'une sonde de base de données, elle crée une table de consignation temporaire et ajoute un déclencheur à la table source. Lorsqu'une modification est apportée à la table source, le déclencheur transmet l'événement à la table de consignation. Les événements stockés dans cette table sont régulièrement transmis au serveur ObjectServer sous forme d'alertes et le contenu de la table de consignation est supprimé. Vous pouvez configurer la fréquence de vérification de la table de consignation par la sonde à la recherche de nouveaux événements.

Avertissement : Les déclencheurs existants dans la table source peuvent être écrasés à l'installation de la sonde.

Les sondes de base de données traitent chaque ligne de la table source comme une seule entité. Même si une seule zone d'une ligne de la table est modifiée, toutes les zones de la ligne sont transmises à la table de consignation puis au serveur ObjectServer. Si une ligne de la table source est supprimée, la sonde transmet son contenu avant la suppression. Si une ligne de la table source est insérée ou mise à jour, la sonde transmet son contenu après l'insertion ou la mise à jour.

Sondes API

Une sonde API acquiert des événements via l'interface de programmation d'application (API) d'une autre application.

Par exemple, la sonde pour Sun Management Center utilise l'API Java Sun Management Center pour se connecter à distance au centre Sun Management Center.

Les sondes API utilisent des bibliothèques spécialement conçues pour l'acquisition d'événements d'une autre application ou d'autres systèmes de gestion. Ces bibliothèques contiennent des fonctions qui se connectent au système cible et gèrent l'extraction des événements. Les sondes API appellent ces fonctions, qui se connectent au système cible et renvoient des événements à la sonde. Cette dernière traite les événements et les transmet au serveur ObjectServer en tant qu'alertes.

Sondes CORBA

L'architecture Common Object Request Broker Architecture (CORBA) permet aux systèmes répartis d'être définis indépendamment d'un langage de programmation spécifique. Les sondes CORBA utilisent des interfaces CORBA pour se connecter à la source de données, qui est généralement un système de gestion des éléments (Element Management System, EMS).

Les fournisseurs d'équipement publient les détails de leur interface CORBA spécifique sous forme de fichier en langage IDL. Ces fichiers IDL sont utilisés pour créer les applications CORBA client et serveur. Une sonde spécifique est requise pour chaque interface CORBA spécifique. Certaines sondes CORBA utilisent la fonction ORB (Object Request Broker) IBM pour communiquer avec les fonctions

ORB d'autres fournisseurs. La fonction ORB IBM est fournie avec Tivoli Netcool/OMNIbus. Certaines sondes CORBA utilisent les fonctions ORB d'autres fournisseurs. Voir la documentation de la sonde concernée pour plus de détails sur la fonction ORB requise.

La majorité des sondes CORBA est écrite à l'aide de Java, et elles nécessitent l'installation de composants Java spécifiques pour leur exécution, comme décrit dans les publications individuelles pour ces sondes. Les sondes écrites en Java utilisent les processus supplémentaires suivants :

- La sonde **probe-nco-p-nonnative**, qui permet la communication entre les sondes écrites en Java et la bibliothèque C de sondes standard (libOpl)
- Les bibliothèques d'exécution Java

Sondes diverses

Toutes les sondes diverses ont des caractéristiques qui les différencient des autres types de sondes et les unes des autres. Chacune de ces sondes effectue une tâche spécialisée qui requiert qu'elle fonctionne de manière unique.

Par exemple, la sonde Email se connecte au serveur de messagerie, extrait les courriers électroniques, les traite, les supprime puis se déconnecte. Elle est utile sur un poste de travail qui ne dispose pas des ressources nécessaires pour permettre de conserver un serveur SMTP et le système de livraison de messages local associé résidents et fonctionnant en continu.

Un autre exemple de sonde de la catégorie diverses est la sonde Ping. Elle est utilisée pour les applications à but général sur les systèmes d'exploitation UNIX et ne nécessite pas de matériel spécial. Vous pouvez utiliser la sonde Ping pour surveiller tout périphérique qui prend en charge le protocole ICMP, comme les commutateurs, les routeurs, les PC et les hôtes UNIX.

Composants de sonde

Une sonde est constituée des composants principaux suivants : un fichier exécutable, un fichier de propriétés et un fichier de règles.

Certaines sondes ont des composants supplémentaires. Si tel est le cas, ces composants sont décrits dans les publications des sondes individuelles.

Fichier exécutable

Le fichier exécutable est le coeur d'une sonde. Il se connecte à la source d'événements, acquiert et traite les événements puis les transmet au serveur ObjectServer sous forme d'alerte.

Sur les systèmes d'exploitation UNIX et Linux 32 bits, les sondes sont installées dans le répertoire `$NCHOME/omnibus/probes/arch`.

Sur les systèmes d'exploitation UNIX et Linux 64 bits, les sondes sont installées dans le répertoire `$NCHOME/omnibus/platform/arch/probes64`.

Sur les systèmes d'exploitation Windows, les sondes sont installées dans le répertoire `%NCHOME%\omnibus\probes\win32`.

Sur tous les systèmes d'exploitation, vous devez utiliser les scripts d'encapsuleur `nco_p_*` dans `$NCHOME/omnibus/probes/` pour démarrer les sondes. Par exemple :

`$NCHOME/omnibus/probes/nco_p_ping`

Lorsque la sonde démarre, elle obtient des informations sur le mode de configuration de son environnement à partir de ses fichiers de propriétés et de règles. La sonde utilise ces informations de configuration pour personnaliser les données qu'elle transmet au serveur ObjectServer.

Concepts associés:

«Fichier de propriétés»

Les propriétés de sonde définissent l'environnement dans lequel la sonde est exécutée.

«Fichier de règles», à la page 8

Le fichier de règles définit comment la sonde traite les données d'événement pour créer une alerte significative. Pour chaque alerte, le fichier de règles crée également un identificateur qui identifie de manière unique la source du problème.

Fichier de propriétés

Les propriétés de sonde définissent l'environnement dans lequel la sonde est exécutée.

Par exemple, la propriété **Server** spécifie le serveur ObjectServer auquel la sonde transmet des alertes. Les fichiers de propriétés de sonde sont stockés dans le répertoire suivant :

`$NCHOME/omnibus/probes/arch`

Où *arch* représente le répertoire de votre système d'exploitation.

Les fichiers de propriétés sont identifiés par l'extension `.props`. Par exemple, le fichier de propriétés de la sonde Ping est : `$NCHOME/omnibus/probes/solaris2/ping.props`.

Les fichiers de propriétés sont constitués de paires nom-valeur séparées par deux-points, comme illustré dans l'exemple suivant :

Server : "NCOMS"

Dans cette paire nom-valeur, **Server** est le nom de la propriété et NCOMS est la valeur sur laquelle la propriété est définie. Les valeurs de chaîne doivent être placées entre guillemets (" "). Les autres valeurs n'en requièrent pas.

Important : Si vous modifiez les propriétés de sonde, il est peut-être nécessaire de redémarrer la sonde pour que la modification prenne effet. La plupart des propriétés de sonde sont en lecture seule lorsque la sonde s'initialise. Si vous modifiez une propriété de sonde pendant qu'une sonde est en cours d'exécution, la modification est stockée en tant que variable dans le fichier de règles de sonde. Vous ne pouvez pas forcer une sonde à enregistrer la valeur nouvelle ou modifiée de la propriété lorsque cette sonde est en cours d'exécution.

Concepts associés:

«Fichier exécutable», à la page 5

Le fichier exécutable est le coeur d'une sonde. Il se connecte à la source d'événements, acquiert et traite les événements puis les transmet au serveur ObjectServer sous forme d'alerte.

Chapitre 4, «Exécution des sondes», à la page 97

Lorsque vous exécutez une sonde, vous pouvez spécifier des propriétés dans un fichier de propriétés ou des commandes sur une ligne de commande pour

configurer les paramètres de la sonde.

Types de propriétés de sonde

Les propriétés de sonde peuvent être divisées en deux catégories : les propriétés communes et les propriétés spécifiques à la sonde.

Les propriétés communes concernent toutes les sondes. Par exemple, la propriété **Server** est une propriété commune car chaque sonde doit connaître le serveur ObjectServer auquel elle envoie les alertes.

Les propriétés spécifiques à la sonde varient en fonction de la sonde. Certaines sondes n'ont pas de propriété spécifique, mais la plupart possède de telles propriétés qui sont liées à leur environnement d'exécution. Par exemple, la sonde Ping a la propriété **Pingfile** qui spécifie le nom d'un fichier qui contient une liste des machines sur lesquelles une commande PING doit être émise.

Les propriétés spécifiques à la sonde sont décrites dans les publications des sondes individuelles.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Utilisation des propriétés de sonde par rapport aux options de ligne de commande des sondes

Chaque propriété de sonde a une option de ligne de commande correspondante.

Par exemple, la propriété **Server** est définie dans le fichier de propriétés comme suit :

```
Server : "NCOMS"
```

Vous pouvez également définir cette propriété sur la ligne de commande en utilisant l'option de ligne de commande `-server` comme suit :

```
$OMNIHOME/probes/nco_p_nom_sonde -server NCOMS
```

L'option de ligne de commande prévaut sur la propriété quand les deux sont définies. Par exemple, si la propriété définit le serveur sur NCOMS et que l'option de ligne de commande le définit sur STWO, la valeur STWO est utilisée pour le nom du serveur ObjectServer.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Fichier de règles

Le fichier de règles définit comment la sonde traite les données d'événement pour créer une alerte significative. Pour chaque alerte, le fichier de règles crée également un identificateur qui identifie de manière unique la source du problème.

Les fichiers de règles fonctionnent comme suit :

1. La sonde acquiert des données brutes à partir de la source d'événement et les divise en jetons. Chaque jeton représente une partie des données d'événement.
2. La sonde fait une analyse syntaxique de ces jetons et les transforme en éléments.
3. La sonde traite ces éléments conformément aux règles du fichier de règles pour affecter des valeurs aux zones du serveur ObjectServer. Ces valeurs de zone sont les détails d'événement au format utilisé par le serveur ObjectServer.
4. Elle transmet les valeurs de zone au serveur ObjectServer sous forme d'alerte.

Dans le fichier de règles, les éléments sont identifiés par le symbole \$. Par exemple, \$Node est un élément contenant le nom du noeud de la source d'événement. Les valeurs de zones sont identifiées par le symbole @. Par exemple, @Node peut être une valeur de zone contenant le nom de poste de la source d'événement.

Lorsque la sonde traite les éléments, la zone @Identifier se voit attribuer une valeur unique. Les alertes dupliquées, c'est-à-dire les événements qui ont la même valeur dans la zone @Identifier sont corrélées, de sorte qu'elles ne sont affichées qu'une fois dans la liste d'événements.

Les fichiers de règles locaux sont stockés dans le répertoire \$NCHOME/omnibus/probes/arch, où arch est le répertoire du système d'exploitation.

Le fichier de règles est identifié par l'extension de fichier .rules. Par exemple, le fichier de règles de la sonde Ping exécutée sous Solaris est : \$NCHOME/omnibus/probes/solaris2/ping.rules

Une sonde lit son fichier de règles uniquement au démarrage ou si la sonde est forcée de relire son fichier de règles. Les fichiers de règles peuvent être mis en cache de sorte que lorsqu'une sonde ne parvient pas à lire son fichier de règles, elle tente de lire les règles du fichier en cache.

Vous pouvez utiliser une adresse Web pour spécifier un fichier de règles situé sur un serveur distant, accessible via HTTP. Cette méthode permet de définir la source de tous les fichiers de règles à partir d'un point central pour chaque sonde. Vous pouvez utiliser un outil de gestion de configuration approprié, comme CVS, sur le point central pour permettre la gestion des versions de tous les fichiers de règles.

Concepts associés:

«Fichier exécutable», à la page 5

Le fichier exécutable est le coeur d'une sonde. Il se connecte à la source d'événements, acquiert et traite les événements puis les transmet au serveur ObjectServer sous forme d'alerte.

«Architecture de sonde», à la page 10

Une sonde a pour fonction d'acquérir des informations d'une source d'événements et de les transmettre au serveur ObjectServer. Les sondes utilisent des *jetons* et des *éléments* et appliquent des règles pour convertir les données de la source d'événement dans un format que le serveur ObjectServer peut reconnaître.

Chapitre 4, «Exécution des sondes», à la page 97

Lorsque vous exécutez une sonde, vous pouvez spécifier des propriétés dans un fichier de propriétés ou des commandes sur une ligne de commande pour configurer les paramètres de la sonde.

Tâches associées:

«Relecture du fichier de règles», à la page 66

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

«Activation de la mise en cache des fichiers de règles de sonde», à la page 69

Pour vous assurer qu'une sonde est toujours en mesure de lire un ensemble valide de règles lorsqu'elle est démarrée, activez la mise en cache du fichier de règles. Par défaut, la mise en cache du fichier de règles est désactivée.

Référence associée:

Chapitre 2, «Fichiers de règles des sondes», à la page 21

Le fichier de règles définit comment la sonde traite les données d'événement pour créer des alertes Tivoli Netcool/OMNIBus significatives. Ce fichier crée également un identificateur pour chaque alerte, afin d'identifier de manière unique la source du problème et de permettre le dédoublement des événements répétés.

Conventions de dénomination des fichiers de composant de sonde

Chaque sonde a un nom abrégé utilisé pour identifier son fichier exécutable et les autres fichiers associés.

Les conventions de dénomination utilisées pour les noms de fichiers de sonde sont présentées dans le tableau suivant .

Tableau 3. Conventions de dénomination des noms de fichiers de sonde

Type de fichier de sonde	Nom et emplacement du fichier
Fichier de propriétés	\$NCHOME/omnibus/probes/arch/nom_sonde.props
Fichier de règles	\$NCHOME/omnibus/probes/arch/nom_sonde.rules

Dans ces chemins :

- *arch* représente le répertoire du système d'exploitation sur lequel la sonde est installée ; par exemple *solaris2* sur un système Solaris.
- *nom_sonde* est le nom abrégé de la sonde.

Par exemple, le nom abrégé de SunNet Manager est *snmlog* et le fichier de propriétés de sonde de SunNet Manager est nommé *snmlog.props*.

Le fichier de règles est nommé *snmlog.rules*.

Architecture de sonde

Une sonde a pour fonction d'acquérir des informations d'une source d'événements et de les transmettre au serveur ObjectServer. Les sondes utilisent des *jetons* et des *éléments* et appliquent des règles pour convertir les données de la source d'événement dans un format que le serveur ObjectServer peut reconnaître.

La figure suivante montre comment les sondes utilisent des règles pour traiter les données d'événement acquises de la source d'événements.

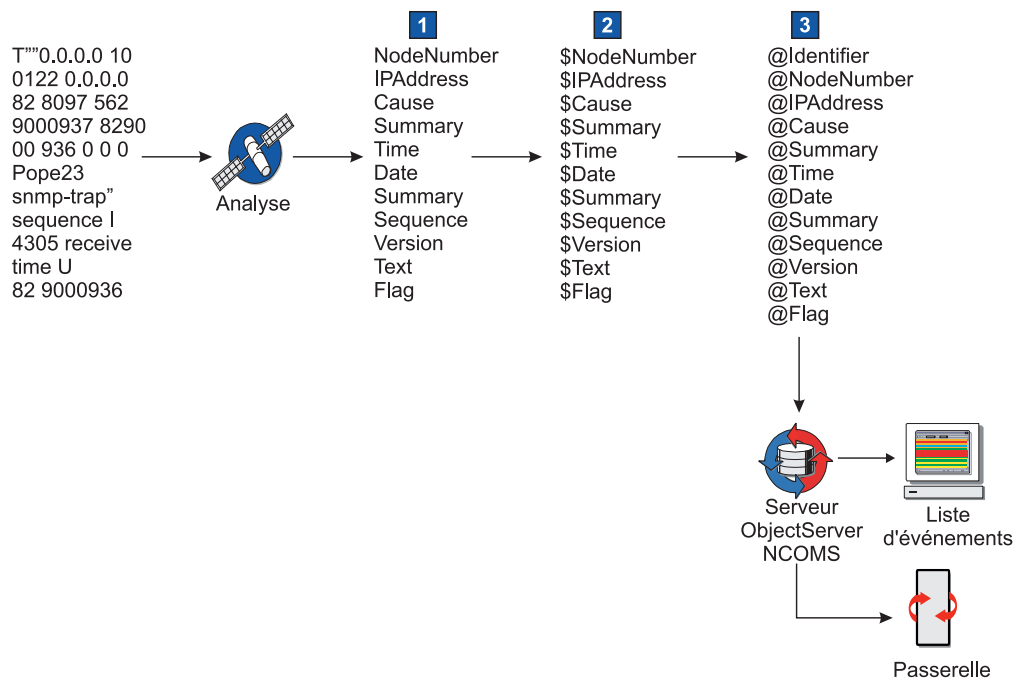


Figure 2. Mappage des événements à l'aide de règles

Les étapes de traitement sont les suivantes :

- 1** Les données d'événement brutes acquises par la sonde ne peuvent pas être envoyées directement au serveur ObjectServer. La sonde divise les données d'événement en jetons. Chaque jeton représente une partie des données d'événement.
- 2** La sonde fait ensuite une analyse syntaxique de ces jetons en éléments et traite ces derniers conformément aux règles du fichier de règles. Les éléments sont identifiés par le symbole \$ dans le fichier de règles. Par exemple, \$Node est un élément contenant le nom du poste de l'événement source.
- 3** Les éléments sont utilisés pour affecter des valeurs aux zones du serveur ObjectServer, indiquées par le symbole @. Les valeurs de zone contiennent les détails de l'événement dans un format que le serveur ObjectServer peut comprendre. Les zones constituent les alertes qui sont transmises au serveur ObjectServer, dans lequel elles sont stockées et gérées dans la table alerts.status et affichées dans la liste d'événements.

La zone Identificateur est également générée par le fichier de règles.

Concepts associés:

«Génération des identificateurs uniques pour les événements»

La zone Identificateur (@Identifieur) identifie de manière unique la source d'un incident. Comme d'autres zones du serveur ObjectServer, la zone Identificateur est générée à partir des jetons que la sonde acquiert dans le flux d'événements conformément aux règles du fichier de règles.

«Fichier de règles», à la page 8

Le fichier de règles définit comment la sonde traite les données d'événement pour créer une alerte significative. Pour chaque alerte, le fichier de règles crée également un identificateur qui identifie de manière unique la source du problème.

Génération des identificateurs uniques pour les événements

La zone Identificateur (@Identifieur) identifie de manière unique la source d'un incident. Comme d'autres zones du serveur ObjectServer, la zone Identificateur est générée à partir des jetons que la sonde acquiert dans le flux d'événements conformément aux règles du fichier de règles.

La zone Identificateur permet au serveur ObjectServer de corréliser des alertes pour n'afficher qu'une seule fois les alertes en double dans la liste d'événements. Au lieu d'insérer une nouvelle alerte, l'alerte est *réinsérée* ; c'est-à-dire que l'alerte existante est mise à jour. Ces mises à jour sont configurables. Par exemple, la zone Tally (@Tally) est généralement incrémentée pour suivre le nombre d'occurrences d'un événement.

Il est essentiel que l'identificateur identifie correctement les événements répétés. L'identificateur suivant n'est pas assez précis, car tous les événements ayant les mêmes gestionnaire et noeud sont traités comme des doubles :

```
@Identifieur=@Manager+@Node
```

Si l'identificateur est trop précis, le serveur ObjectServer ne peut pas corréliser et dédoubler les événements répétés. Par exemple, un identificateur qui contient une valeur temporelle empêche le dédoublage correct.

L'identificateur ci-dessous identifie correctement les événements répétés dans un environnement type :

```
@Identifieur=@Node+" "+@AlertKey+" "+@AlertGroup+" "+@Type+" "+@Agent+" "+@Manager
```

Dédoublage d'événements avec des sondes

Le dédoublage est géré par le serveur ObjectServer, mais peut être configuré dans le fichier de règles de sonde. Vous pouvez ainsi définir des règles de dédoublage par événement. Vous pouvez spécifier les zones d'une alerte qui doivent être mises à jour si l'alerte est dédoublée à l'aide de la fonction update.

Concepts associés:

«Architecture de sonde», à la page 10

Une sonde a pour fonction d'acquies des informations d'une source d'événements et de les transmettre au serveur ObjectServer. Les sondes utilisent des *jetons* et des *éléments* et appliquent des règles pour convertir les données de la source d'événement dans un format que le serveur ObjectServer peut reconnaître.

Référence associée:

Chapitre 2, «Fichiers de règles des sondes», à la page 21

Le fichier de règles définit comment la sonde traite les données d'événement pour créer des alertes Tivoli Netcool/OMNibus significatives. Ce fichier crée également un identificateur pour chaque alerte, afin d'identifier de manière unique la source

du problème et de permettre le dédoublement des événements répétés.

«Fonction update sur dédoublement», à la page 48

Le serveur ObjectServer gère le processus de dédoublement, mais vous pouvez également le configurer dans le fichier de règles de sonde. La fonction update vous permet de spécifier les zones d'une alerte qui sont mises à jour si l'alerte est dédoublement. Les règles de dédoublement peuvent ainsi être définies alerte par alerte.

Modes de fonctionnement des sondes

Vous pouvez configurer les sondes pour qu'elles fonctionnent dans différents modes, y compris le mode stocker-et-transmettre, le mode capture brute, le mode sécurisé et le mode de reprise en ligne d'égal à égal.

Mode stocker-et-transmettre pour les sondes

L'exécution des sondes peut continuer si le serveur ObjectServer cible est arrêté. Pendant cette période, la sonde passe en mode *stocker*. Elle repasse en mode *transmettre* lorsque le serveur ObjectServer fonctionne à nouveau.

Stockage et transmission automatiques

Par défaut, le mode stocker-et-transmettre est actif seulement lorsqu'une connexion au serveur ObjectServer a été établie, utilisée puis perdue. Si le serveur ObjectServer n'est pas en cours d'exécution au démarrage de la sonde, le mode stocker-et-transmettre n'est pas déclenché et la sonde s'arrête.

Vous pouvez configurer la sonde pour l'exécution en mode stocker-et-transmettre *automatique*. Dans ce mode, la sonde entre directement dans le mode stocker si le serveur ObjectServer n'est pas en cours d'exécution, à condition que la sonde ait été connectée au serveur ObjectServer au moins une fois avant cela. Activez le mode stocker-et-transmettre automatique à l'aide de l'option de ligne de commande `-autosaf` ou de la propriété **AutoSAF**.

Remarque : Si la sonde tente de se connecter à une paire virtuelle de serveurs ObjectServer et que les deux serveurs ObjectServer sont hors tension, la sonde vérifie le paramètre de la propriété **AutoSAF**. Si le stockage et le réacheminement automatiques sont activés, la sonde commence à stocker des événements dans le fichier stocker-et-transmettre. Dans le cas contraire, la sonde s'arrête.

Fonction stocker et transmettre héritée

Lorsque la sonde détecte que le serveur ObjectServer est absent (généralement car elle ne peut pas lui transmettre d'alerte), elle passe en mode stocker. Avec ce mode, la sonde écrit tous les messages qu'elle enverrait normalement au serveur ObjectServer dans un fichier stocker-et-transmettre. Ce nom de fichier est généré à l'aide de la valeur de la propriété **SAFFileName**. Une extension *.nom_serveur* est automatiquement ajoutée à la valeur **SAFFileName**, où *nom_serveur* est le nom du serveur ObjectServer auquel la sonde tente d'envoyer les alertes. Si la sonde est configurée pour envoyer des alertes à plusieurs serveurs ObjectServer, les fichiers stocker-et-transmettre individuels sont créés pour chaque ObjectServer.

Si des enregistrements corrompus sont identifiés dans un fichier stocker-et-transmettre, ils sont ignorés et la sonde transmet uniquement les enregistrements valides au serveur ObjectServer. La propriété **KeepLastBrokenSAF** permet d'indiquer si vous souhaitez sauvegarder automatiquement un fichier

contenant les enregistrements corrompus à des fins de diagnostic. Si vous définissez cette propriété sur 1, le fichier stocker-et-transmettre contenant des enregistrements corrompus est renommé *SAFFileName.nom_serveur.broken*. Dans ce nom de fichier, *SAFFileName* est la valeur de la propriété **SAFFileName** et *nom_serveur* est le nom du serveur ObjectServer auquel la sonde tente d'envoyer des alertes. Les éventuels fichiers *.broken* précédents sont écrasés.

Vous pouvez également utiliser la propriété **StoreSAFRejects** pour sauvegarder en permanence les différents enregistrements stocker-et-transmettre corrompus à des fins d'analyse. Si **StoreSAFRejects** est défini sur 1, les enregistrements corrompus sont sauvegardés en permanence dans un fichier *SAFFileName.nom_serveur.rejected*.

Conseil : La taille du fichier *SAFFileName.nom_serveur.rejected* est illimitée ; ce fichier doit être supprimé manuellement lorsque vous n'en avez plus besoin.

Le fichier stocker-et-transmettre existant peut être configuré à l'aide des propriétés de l'exemple suivant. **StoreAndForward** doit être défini sur 1 pour la fonction stocker-et-transmettre existante. Les autres propriétés affichent les valeurs par défaut qui peuvent être modifiées.

```
StoreAndForward:1
SAFFileName:'$OMNIHOME/var/SAF'
MaxSAFFileSize:1024
SAFPoolSize:3
```

Mode stocker-et-transmettre circulaire

Exécutez la sonde en mode stocker-et-transmettre circulaire afin de minimiser la perte d'événements lors de la reprise en ligne et de la reprise par restauration. Avec ce mode, la sonde stocke toutes les alertes qu'elle génère lorsqu'elle est connectée au serveur ObjectServer. Ces alertes sont stockées dans des fichiers stocker-et-transmettre tournants après un intervalle défini par la propriété **RollSAFInterval**. Définissez la propriété **RollSAFInterval** sur une valeur supérieure ou égale à la granularité du serveur ObjectServer.

Les fichiers stocker-et-transmettre circulaires sont appelés *SAFFileName.nom_serveur* et *SAFFileName.nom_serveur_1*.

Lorsque la sonde est déconnectée du serveur ObjectServer, elle enregistre l'horodatage du dernier événement ayant abouti ainsi que le nom de serveur ObjectServer dans un fichier. Ce fichier est nommé selon le format *SAFFilename.DisconnectionTime*. Ce fichier est stocké dans le même répertoire que les fichiers stocker-et-transmettre. Si un serveur ObjectServer de secours est disponible pour la reprise en ligne, la sonde utilise l'heure de déconnexion et la valeur de la propriété **RollSAFInterval** pour calculer les événements à envoyer au serveur ObjectServer de secours. Par exemple, si la propriété **RollSAFInterval** est définie sur 90 secondes, tous les événements à partir de 90 secondes avant l'heure de déconnexion sont réexécutés sur le serveur ObjectServer de secours. De ce fait, certains événements peuvent être dupliqués sur le serveur ObjectServer de secours. La sonde renvoie des événements qui peuvent avoir déjà été envoyés au serveur ObjectServer principal, mais qui ont pas été répliqués sur le serveur ObjectServer de sauvegarde avant l'échec du serveur principal.

Si la sonde ne peut pas se connecter à un serveur ObjectServer, sa fonction de gestion des fichiers stocker-et-transmettre tournants est automatiquement commutée sur la fonction stocker-et-transmettre héritée. La sonde commence à

stocker tous les événements dans un pool de fichiers stocker-et-transmettre ; la taille du pool est définie par la propriété **SAFPoolSize**, et la taille maximale de fichier est définie par la propriété **MaxSAFFileSize**. Dans le même temps, la propriété **RollSAFInterval** n'est pas utilisée pour faire tourner les fichiers stocker-et-transmettre ; le roulement des fichiers s'effectue lorsqu'un fichier atteint la taille maximale définie par la propriété **MaxSAFFileSize**.

Les fichiers stocker-et-transmettre circulaires peuvent être configurés à l'aide des propriétés suivantes. **StoreAndForward** doit être défini sur 2 pour la fonction stocker-et-transmettre circulaire ; les autres propriétés sont définies par défaut, et peuvent être modifiées.

```
StoreAndForward:2
SAFFileName:'$OMNIHOME/var/SAF'
MaxSAFFileSize:1024
SAFPoolSize:3
RollSAFInterval:90
```

Récapitulatif de la fonction stocker-et-transmettre

La table suivante récapitule la façon dont le statut du serveur ObjectServer et l'association des propriétés **AutoSAF** et **StoreAndForward** affectent le comportement des sondes.

Tableau 4. Récapitulatif de stocker-et-transmettre

Statut du serveur ObjectServer avant le démarrage de la sonde	Propriété AutoSAF	Propriété StoreAndForward	Résultat attendu
Serveur ObjectServer arrêté	0	0	La sonde ne démarre pas.
ObjectServer arrêté	0	1	La sonde ne démarre pas.
ObjectServer arrêté	1	0	La sonde commence à écrire des événements dans le fichier stocker-et-transmettre. Lorsque le serveur ObjectServer s'active, la sonde transmet les événements de fichier stocker-et-transmettre puis arrête d'écrire les événements dans ce fichier. Si le serveur ObjectServer se déconnecte, les événements ne sont pas stockés.
ObjectServer arrêté	1	1	La sonde démarre en écrivant les événements dans le fichier stocker-et-transmettre. Lorsque le serveur ObjectServer s'active, la sonde transmet les événements de fichier stocker-et-transmettre. Si le serveur ObjectServer se déconnecte ultérieurement, la sonde stocke les événements dans les fichiers stocker-et-transmettre. Ces événements seront transmis après reconnexion.

Tableau 4. Récapitulatif de stocker-et-transmettre (suite)

Statut du serveur ObjectServer avant le démarrage de la sonde	Propriété AutoSAF	Propriété StoreAndForward	Résultat attendu
ObjectServer actif	Pas d'effet de propriété	0	La sonde transmet les événements aux fichiers stocker-et-transmettre existants, mais elle ne stocke pas les nouveaux événements dans les fichiers stocker-et-transmettre. Si une sonde perd sa connexion au serveur ObjectServer, elle s'interrompt en mode contrôlé. Tous les événements que la sonde a reçus, mais n'a pas transmis à l'ObjectServer sont perdus. Une sonde qui possède plusieurs connexions au serveur ObjectServer ne peut pas s'exécuter en mode stocker-et-transmettre 0 et bascule automatiquement en mode stocker-et-transmettre 1.
ObjectServer actif	Pas d'effet de propriété	1	La sonde transmet les événements des fichiers stocker-et-transmettre au serveur ObjectServer connecté, et stocke les nouveaux événements dans les fichiers stocker-et-transmettre uniquement lorsqu'elle est déconnectée du serveur ObjectServer.
ObjectServer actif	Pas d'effet de propriété	2	La sonde transmet les événements des fichiers stocker-et-transmettre au serveur ObjectServer connecté, et stocke les nouveaux événements dans les fichiers stocker-et-transmettre tournants lorsqu'elle est connectée. La sonde stocke tous les événements dans un pool de fichiers lorsqu'elle est déconnectée.

Référence associée:

«Traitement multiprocessus des données d'alerte», à la page 55

Lorsqu'un fichier de règles de sonde est traité, le traitement multiprocessus est utilisé par défaut pour appliquer les règles de sonde aux données d'événements brutes obtenues sur la source d'événements, et pour envoyer les alertes générées aux serveurs ObjectServer enregistrés. Notez que ce traitement multiprocessus est différent de la capture d'événements mutliprocessus ou monoprocessus implémentée sur certaines classes de sondes.

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Mode de capture brute pour les sondes

Le mode capture brute vous permet de sauvegarder l'intégralité d'un flux de données d'événement acquis par une sonde dans un fichier, sans traitement par le fichier de règles. Cela peut être utile à des fins d'audit, d'enregistrement ou de débogage du fonctionnement d'une sonde.

Les données capturées sont à un format qui peut être relu par la sonde Standard Input File. Voir la documentation relative à la sonde Standard Input File pour plus d'informations. Vous pouvez accéder à cette publication de la manière suivante, à partir du centre de documentation IBM Tivoli Network Management (<http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp>) :

1. Développez le noeud *IBM Tivoli Netcool/OMNIBus* dans le panneau de navigation située sur la gauche.
2. Développez le noeud *Tivoli Netcool/OMNIBus probes and TSMs*.
3. Accédez au noeud *Universal*.

Pour activer le mode capture brute, utilisez l'option de ligne de commande `-raw` ou la propriété **RawCapture**. Vous pouvez également définir la propriété **RawCapture** dans le fichier de règles pour pouvoir envoyer les données d'événement brutes dans un fichier uniquement lorsque certaines conditions sont remplies.

Rediffusez les données capturées brutes à l'aide de la sonde Standard Input. La syntaxe peut être la suivante :

```
cat <nomdefichier_capture_brute> | $OMNIHOME/probes/nco_p_stdin -server <serveur>
```

Par exemple :

```
cat opt/Omnibus/var/mttrapd.cap | /opt/Omnibus/probes/nco_p_stdin -server NCOMS
```

Les propriétés **RawCaptureFile**, **RawCaptureFileAppend** et **MaxRawFileSize** contrôlent également le fonctionnement du mode capture brute.

Référence associée:

«Modification de la valeur de la propriété RawCapture dans le fichier de règles», à la page 25

La majorité des sondes lit les propriétés une fois au démarrage, la modification des propriétés de sonde après le démarrage n'affecte donc généralement pas le comportement de la sonde. Vous pouvez toutefois définir la propriété **RawCapture** dans le fichier de règles pour pouvoir envoyer les données d'événement brutes dans un fichier uniquement lorsque certaines conditions sont remplies.

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Mode sécurisé pour les sondes

Vous pouvez exécuter le serveur ObjectServer en mode sécurisé. Lorsque vous démarrez le serveur ObjectServer à l'aide de l'option de ligne de commande `-secure`, il authentifie les connexions de sonde, de passerelle et de serveur proxy en demandant un nom d'utilisateur et un mot de passe.

Lorsqu'une demande de connexion est envoyée, le serveur ObjectServer émet un message d'authentification. La sonde, la passerelle ou le serveur proxy doit répondre avec la combinaison nom d'utilisateur-mot de mot de passe correcte.

Si le serveur ObjectServer n'est pas exécuté en mode sécurisé, les demandes de connexion de la sonde, de la passerelle et du serveur proxy ne sont pas authentifiées.

Avant d'exécuter une sonde qui se connecte à un serveur ObjectServer sécurisé ou à un serveur proxy, assurez-vous que les propriétés **AuthUserName** et **AuthPassword** sont définies dans le fichier de propriétés de sonde avec des valeurs pour le nom d'utilisateur et le mot de passe. Si la combinaison nom d'utilisateur-mot de passe est incorrecte, le serveur ObjectServer émet un message d'erreur et refuse la connexion.

Lorsque le mode FIPS 140-2 est activé, le mot de passe peut être spécifié en texte normal ou chiffré à l'aide de l'utilitaire **nco_aes_crypt**. Si vous chiffrez des mots de passe à l'aide de l'utilitaire **nco_aes_crypt** en mode FIPS 140-2, vous devez spécifier `AES_FIPS` comme algorithme de chiffrement.

Lorsque le mode FIPS 140-2 est désactivé, le mot de passe peut être chiffré à l'aide des utilitaires **nco_g_crypt** ou **nco_aes_crypt**. Si vous chiffrez des mots de passe à l'aide de l'utilitaire **nco_aes_crypt** en mode FIPS 140-2 désactivé, vous pouvez spécifier `AES_FIPS` ou `AES` comme algorithme de chiffrement. Utilisez uniquement `AES` si vous devez conserver une compatibilité avec des mots de passe chiffrés à l'aide des outils fournis dans des versions antérieures à Tivoli Netcool/OMNIBus V7.2.1.

Pour obtenir plus d'informations sur l'utilisation de l'utilitaire **nco_aes_crypt**, voir le manuel *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIBus*.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Mode de reprise en ligne d'égal à égal pour les sondes

Deux instances d'une sonde peuvent être exécutées simultanément dans une relation de reprise en ligne d'égal à égal. Une instance est désignée comme maître. L'autre agit comme esclave et est en mode secours automatique. En cas d'échec de l'instance maître, l'instance esclave est activée.

Remarque : Le mode reprise en ligne d'égal à égal n'est pas pris en charge pour toutes les sondes. Les sondes qui répertorient les propriétés **Mode**, **PeerHost** et **PeerPort** lorsque vous exécutez la commande `$OMNIHOME/probes/nco_p_nom_sonde -dumpprops` prennent en charge le mode reprise en ligne d'égal à égal.

Pour configurer une relation de reprise en ligne d'égal à égal :

- Pour l'instance maître, définissez la propriété **Mode** sur master et la propriété **PeerHost** sur le nom de l'élément réseau de l'esclave.
- Pour l'instance esclave, définissez la propriété **Mode** sur slave et la propriété **PeerHost** sur le nom de l'élément réseau du maître.
- Pour les deux instances, définissez la propriété **PeerPort** sur le port via lequel le maître et l'esclave communiquent.

L'instance maître envoie un signal de présence à l'instance esclave à chaque intervalle défini par la propriété **BeatInterval**. L'instance esclave met en cache toutes les données d'alerte qu'elle reçoit et supprime ces données d'alerte à chaque fois qu'elle reçoit un signal de l'instance maître. Si l'instance esclave ne reçoit pas de signal dans le délai défini par la somme des valeurs des propriétés **BeatInterval** et **BeatThreshold** ($\text{BeatInterval} + \text{BeatThreshold}$), l'instance esclave suppose que l'instance maître n'est plus active, et transmet les alertes de la mémoire cache au serveur ObjectServer. L'instance esclave continue à transmettre toutes les alertes jusqu'à ce qu'elle reçoive un signal de présence de l'instance maître. Le délai de dépassement d'attente du signal de présence est de 1 seconde. Le délai d'attente maximal est de $(\text{BeatInterval} + \text{BeatThreshold} + 1)$ secondes avant que l'instance esclave transmette les alertes de sa mémoire cache. Toutes les alertes de la mémoire cache sont envoyées.

Le paramètre **BeatInterval** défini pour l'instance maître prévaut ; le paramètre local **BeatInterval** de l'instance esclave est ignoré.

Pour désactiver la relation de reprise en ligne d'égal à égal, exécutez une seule instance de la sonde avec la propriété **Mode** définie sur standard. Il s'agit du paramétrage par défaut.

Le mode de reprise en ligne des sondes exécutées en relation de reprise en ligne d'égal à égal est défini dans le fichier de propriétés.

Vous pouvez également commuter le mode d'une sonde entre maître et esclave dans le fichier de règles. Un délai d'environ une seconde s'écoule avant que la modification de mode entre en vigueur. Cela peut engendrer des événements en double si deux instances de sonde sont commutées du mode standard en master ou slave ; toutefois, aucune donnée n'est perdue.

Lorsque deux instances exécutées en mode stocker-et-transmettre sont connectées à une paire de reprise en ligne de serveurs ObjectServer, l'instance maître envoie des alertes au serveur ObjectServer principal. Si le serveur ObjectServer principal échoue, l'instance maître de la sonde effectue une reprise en ligne et commence à envoyer les alertes de son fichier stocker-et-transmettre au serveur ObjectServer de secours. Si l'instance maître de la sonde échoue, l'instance esclave prend le relais. Si l'instance esclave ne parvient pas à se connecter au serveur ObjectServer, l'esclave crée un fichier stocker-et-transmettre pour stocker les données d'alerte. Si l'instance maître est réactivée, tous les fichiers stocker-et-transmettre de l'instance maître sont supprimés pour éviter que les anciennes alertes soient renvoyées.

Exemple : configuration du mode de reprise en ligne d'égal à égal dans le fichier de propriétés

Les exemples de valeurs de fichier de propriété pour le maître sont les suivants :

```
PeerPort: 9999
PeerHost: "slavehost"
Mode: "master"
```

Les exemples de valeurs de fichier de propriété pour l'esclave sont les suivants :

```
PeerPort: 9999  
PeerHost: "masterhost"  
Mode: "slave"
```

Exemple : configuration du mode de reprise en ligne d'égal à égal dans le fichier de règles

Pour faire passer une instance en maître, utilisez la syntaxe de fichier de règles suivante :

```
%Mode = "master"
```

Chapitre 2. Fichiers de règles des sondes

Le fichier de règles définit comment la sonde traite les données d'événement pour créer des alertes Tivoli Netcool/OMNIbus significatives. Ce fichier crée également un identificateur pour chaque alerte, afin d'identifier de manière unique la source du problème et de permettre le dédoublement des événements répétés.

Concepts associés:

«Génération des identificateurs uniques pour les événements», à la page 11

La zone Identificateur (@Identifieur) identifie de manière unique la source d'un incident. Comme d'autres zones du serveur ObjectServer, la zone Identificateur est générée à partir des jetons que la sonde acquiert dans le flux d'événements conformément aux règles du fichier de règles.

«Fichier de règles», à la page 8

Le fichier de règles définit comment la sonde traite les données d'événement pour créer une alerte significative. Pour chaque alerte, le fichier de règles crée également un identificateur qui identifie de manière unique la source du problème.

Développement de fichiers de règles

Utilisez le guide suivant pour développer les fichiers de règles en respectant un format et une structure cohérents :

- Les fichiers de règles doivent être des produits de qualité et ne nécessiter aucune modification supplémentaire.
- Les fichiers de règles ne doivent entraîner aucune modifications supplémentaires au serveur ObjectServer. En d'autres termes, aucune zone d'événement (autre que celles fournies par Tivoli Netcool/OMNIbus) ne doit être ajoutée.
- La structure basique des fichiers de règles doit être à la fois facilement entretenue et facilement extensible pour permettre une la gestion d'un plus grand nombre d'événements pour de nouvelles unités, sans pour autant affecter les règles existantes.
- Les conventions textuelles basiques utilisées pour les fichiers de règles doivent être cohérentes, permettant ainsi que tous les fichiers de règles récemment créés soient au même format.
- Les fichiers de règles doivent être clairement documentés pour permettre à chaque événement d'être reconnu, sans aucune documentation supplémentaire.
- Les événements formatés par les fichiers de règles doivent être dupliqués convenablement par le serveur ObjectServer Tivoli Netcool/OMNIbus. La zone Identificateur (@Identifieur) doit être correctement définie afin que la granularité du dédoublement soit contrôlée directement.
- Les événements formatés par les fichiers de règles doivent être compatibles - autant que possible - avec l'automatisation GenericClear du serveur ObjectServer.
- Effectuez toujours une copie de sauvegarde du fichier de règles avant de le modifier. Enregistrez-le sous forme de fichier `.rules.date`. Par exemple, `snmp.rules.070131`. Vous aurez besoin de ce fichier en cas d'annulation.
- Tout changement effectué dans le fichier de règles doit être commenté à l'aide d'un signe dièse (#) en début de ligne.
- Utilisez la fonction `details ($*)` seulement pour le débogage ou l'écriture des fichiers de règles. Après avoir utilisé la commande `details ($*)` durant de

longues périodes, les tables du serveur ObjectServer deviennent très volumineuses et les performances du serveur ObjectServer s'en ressentent.

- Les constructions SWITCH et CASE sont traitées de manière plus efficace ; par conséquent, elles devraient avoir la préférence sur les instructions IF et ELSE.
- Utilisez les tables de recherche autant que possible. Lorsque des valeurs multiples sont liées à une clé unique, utilisez une table de consultation à plusieurs colonnes. Les tables de consultation doivent être définies à l'intérieur d'une table basée sur un fichier externe et doivent indiquer l'extension de fichier suivant : `.lookup`. Cela permet d'identifier clairement les tables de recherche. En outre, les tables de recherche doivent être les premiers éléments des fichiers de règles que la sonde lit. C'est à dire, dans le fichier de règles de base, placez l'instruction `include` du fichier de recherche en haut.
- Les paires de correspondance d'événements de problème et de résolution doivent avoir des valeurs identiques pour les zones `@AlertGroup` et `@AlertKey`, et des valeurs appropriées pour `@Type` et `@Severity`.
- Un événement de résolution doit avoir une alerte de gravité de niveau 1 (indéterminé) et un type de 2 (résolution). Ne définissez pas la gravité des événements de résolution sur 0 (supprimé). Cela les empêcherait d'être traités par l'automatisation `GenericClear` du serveur ObjectServer.
- Le fichier de règles doit être créé dans le codage de caractère utilisé par l'environnement local de l'environnement d'exécution de la sonde. Le codage de caractères doit être pris en charge par les bibliothèques ICU (International Components for Unicode).

Tâches associées:

«Test de fichiers de règles», à la page 64

Vous pouvez tester la syntaxe d'un fichier de règles à l'aide du vérificateur de syntaxe de règles de sonde, `nco_p_syntax`. Cela se révèle plus efficace que l'exécution de la sonde pour vérifier que la syntaxe du fichier de règles est correcte.

Éléments, zones, propriétés et tableaux dans les fichiers de règles

Une sonde prend un flux d'événements et l'analyse en éléments. Les éléments d'événement sont traités par la sonde en fonction de la logique du fichier de règles. Les éléments sont affectés aux zones et transmis au serveur ObjectServer, où ils sont insérés en tant qu'alertes dans la table `alerts.status`.

La zone Identificateur, utilisée par le serveur ObjectServer pour le dédoublement, est également créée en se basant sur la logique du fichier de règles.

Les éléments sont indiqués par le symbole `$` dans le fichier de règles. Par exemple, `$Node` est un élément contenant le nom du poste de l'événement source. Vous pouvez affecter des éléments aux zones du serveur ObjectServer, indiquées par le symbole `@` dans le fichier de règles.

Remarque : Le format normal pour faire référence aux éléments fonctionne uniquement si le nom de l'élément contient seulement des lettres, des chiffres et des caractères de soulignement. Si une sonde génère de manière dynamique les noms d'éléments, il est possible de générer des éléments qui contiennent d'autres caractères. Vous pouvez faire référence à de tels éléments en indiquant leur nom entre parenthèses ; par exemple `$(strange=name)`.

Concepts associés:

«Génération des identificateurs uniques pour les événements», à la page 11
La zone Identificateur (@Identifier) identifie de manière unique la source d'un incident. Comme d'autres zones du serveur ObjectServer, la zone Identificateur est générée à partir des jetons que la sonde acquiert dans le flux d'événements conformément aux règles du fichier de règles.

Affectation de valeurs aux zones du serveur ObjectServer

Vous pouvez affecter des valeurs aux zones du serveur ObjectServer par affectation directe, concaténation ou en ajoutant du texte.

Les exemples sont les suivants :

- Exemple d'affectation directe : @Node = \$Node
- Exemple de concaténation : @Summary = \$Summary + \$Group
- Exemple d'ajout de texte : @Summary = \$Node + "has problem" + \$Summary

Vous pouvez exprimer les valeurs numériques sous forme décimale ou hexadécimale. Les instructions suivantes, qui définissent la zone Class sur 100, sont équivalentes :

- @Class=100
- @Class=0x64

En plus d'affecter des éléments aux zones, vous pouvez utiliser des instructions, des opérateurs et des fonctions de traitement pour manipuler ces valeurs dans les fichiers de règles avant de les affecter.

Conseil : Les éléments sont stockés sous forme de chaîne, vous devez donc utiliser la fonction int pour les convertir en entiers avant d'effectuer des opérations numériques.

Référence associée:

«Fonctions mathématiques», à la page 42

Les fonctions mathématiques vous permettent d'effectuer des opérations numériques sur des éléments. Les éléments sont stockés sous forme de chaîne, vous devez donc utiliser ces fonctions pour les convertir en entiers avant d'effectuer des opérations numériques.

Affectation d'éléments temporaires dans des fichiers de règles

Vous pouvez créer un élément temporaire dans un fichier de règles en l'affectant à une expression.

Par exemple :

```
$tempement = "message"
```

Un élément \$tempement est créé et la valeur de chaîne message lui est affectée.

Si vous faites référence à un élément qui n'a pas été initialisé de cette façon, il est défini sur la chaîne nulle ("").

L'exemple suivant crée l'élément \$b et le définit sur setnow :

```
$b="setnow"
```

L'exemple suivant définit ensuite l'élément \$a sur setnow :

```
$a=$b
```

Dans l'exemple suivant, des éléments temporaires sont utilisés pour extraire des informations d'un élément Summary qui a la valeur de chaîne : The Port is down on Port 1 Board 2.

```
$temp1 = extract ($Summary, "Port ([0-9]+)")
$temp2 = extract ($Summary, "Board ([0-9]+)")
@AlertKey = $temp1 + "." + $temp2
```

La fonction extract est utilisée pour affecter des valeurs aux éléments temporaires temp1 et temp2. Ces éléments sont ensuite concaténés (à l'aide de l'opérateur de concaténation +) avec un . pour les séparer et affectés à la zone AlertKey. Une fois que ces instructions ont été exécutées, la zone AlertKey a la valeur 1.2.

Référence associée:

«Fonctions de chaîne», à la page 38

Les fonctions de chaîne vous permettent de manipuler des éléments de chaîne, généralement des noms de zones ou d'éléments.

«Opérateurs mathématiques et de chaîne», à la page 35

Les opérateurs mathématiques vous permettent d'ajouter, de soustraire, de diviser et de multiplier des opérandes numériques dans des expressions. Les opérateurs de chaîne vous permettent de manipuler des chaînes de caractères.

Affectation de valeurs de propriété aux zones

Vous pouvez affecter la valeur d'une propriété de sonde, définie dans le fichier de propriétés ou sur la ligne de commande, à une valeur de zone. Une propriété est représentée par le symbole % dans le fichier de règles.

Vous pouvez par exemple ajouter l'instruction suivante à votre fichier de règles :

```
@Summary = "Server = " + %Server
```

Dans cet exemple, lorsque le fichier de règles est traité, la sonde recherche une propriété **Server**. Si elle est trouvée, sa valeur est concaténée en chaîne de texte et affectée à la zone Summary. Si la propriété est introuvable, une chaîne nulle ("") est affectée.

Affectation de valeurs à des propriétés

Vous pouvez affecter des valeurs à une propriété dans le fichier de règles. Si la propriété n'existe pas, elle est créée.

Vous pouvez par exemple créer une propriété appelée **Counter** pour suivre le nombre d'événements traités comme suit :

```
if (match(%Counter,""))
  {%Counter = 1}
else {%Counter = int(%Counter) + 1}
```

Ces propriétés conservent leurs valeurs à travers les événements et à la relecture du fichier.

Modification de la valeur de la propriété RawCapture dans le fichier de règles :

La majorité des sondes lit les propriétés une fois au démarrage, la modification des propriétés de sonde après le démarrage n'affecte donc généralement pas le comportement de la sonde. Vous pouvez toutefois définir la propriété **RawCapture** dans le fichier de règles pour pouvoir envoyer les données d'événement brutes dans un fichier uniquement lorsque certaines conditions sont remplies.

La définition du mode capture brute entre en vigueur pour l'événement en cours.

Par exemple :

```
# Start rules processing
%RawCapture=0

if (condition) {
  # Send the current event to the raw capture file
  %RawCapture=1
}
```

Vous pouvez activer le mode capture brute de manière générale en définissant l'option de ligne de commande `-raw` ou la propriété **RawCapture** dans le fichier de propriétés de sonde.

Concepts associés:

«Mode de capture brute pour les sondes», à la page 16

Le mode capture brute vous permet de sauvegarder l'intégralité d'un flux de données d'événement acquis par une sonde dans un fichier, sans traitement par le fichier de règles. Cela peut être utile à des fins d'audit, d'enregistrement ou de débogage du fonctionnement d'une sonde.

Référence associée:

«Instructions de contrôle dans les fichiers de règles», à la page 26

Les instructions IF, SWITCH, FOREACH et BREAK permettent de contrôler le traitement des fichiers de règles.

Utilisation de tableaux

Vous devez définir des tableaux au début d'un fichier de règles avant d'indiquer des instructions de traitement.

Conseil : Vous devez également définir des tables et des serveurs ObjectServer cibles avant les instructions de traitement.

Pour définir un tableau, utilisez la syntaxe suivante :

```
array node_arr
```

Les tableaux ont une seule dimension. A chaque affectation pour une valeur de clé existante, la valeur précédente est écrasée. Par exemple :

```
node_arr["myhost"] = "a"
node_arr["yourhost"] = "b"
node_arr["myhost"] = "c"
```

Après l'exécution des instructions précédentes, il y a deux éléments dans le tableau `node_arr`. L'élément avec la clé `myhost` est défini sur `c` et l'élément avec la clé `yourhost` est défini sur `b`. Vous pouvez effectuer des affectations à l'aide des éléments de sonde, par exemple :

```
node_arr[$Node] = "d"
```

Remarque : Les valeurs de tableau sont conservées jusqu'au redémarrage de la sonde. Si vous obligez la sonde à relire le fichier de règles en émettant une commande `kill -HUP pid` sur l'ID processus de sonde, les valeurs du tableau sont conservées.

Référence associée:

«Opérations de table de consultation», à la page 45

Les tables de consultation permettent d'ajouter des informations supplémentaires à un événement. Une table de consultation est constituée d'une liste de clés et de valeurs.

«Envoi d'alertes à des serveurs ObjectServer et tables alternatifs», à la page 50

Les fonctions `registertarget`, `genevent`, `settarget` et `setdefaulttarget` permettent d'envoyer des alertes à un ou plusieurs serveurs ObjectServer et de définir la distribution de ces alertes entre ces serveurs.

Instructions de contrôle dans les fichiers de règles

Les instructions `IF`, `SWITCH`, `FOREACH` et `BREAK` permettent de contrôler le traitement des fichiers de règles.

Instruction `FOREACH`

Utilisez l'instruction `FOREACH` pour écrire des instructions dans le langage des fichiers de règles d'analyse qui se répandent à travers des listes d'éléments d'événements ou des entrées de tables.

Syntaxe

L'instruction `FOREACH` respecte la syntaxe suivante :

```
foreach (itérateur in liste)
{
  instructions
}
```

Dans cette syntaxe, *itérateur* représente l'élément qui doit être identifié par l'instruction. *itérateur* peut être composé de n'importe quelle combinaison de caractères ASCII imprimables et doit commencer par une lettre. Par exemple: *lettre {lettre|chiffre}*.

liste représente les éléments qui doivent être traités dans le fichier de règles. *liste* peut être une liste délimitée par des virgules répertoriant des éléments ou un tableau. Vous pouvez utiliser `$*` pour ordonner à la boucle de traiter tous les éléments. L'instruction traite les éléments de la *liste* dans un ordre non déterministe.

instructions représente les instructions des fichiers de règles d'analyse valides ou les fonctions qui doivent être appliquées aux éléments de la *liste*.

Vous pouvez imbriquer les instructions `FOREACH` les unes à l'intérieur des autres. L'instruction `FOREACH` prend en charge les instructions `IF` et les instructions de commutateur dans le corps d'une boucle.

Dans une instruction, *itérateur* représente la boucle en cours. Le référencement de sa valeur dépend du type de la liste en cours de traitement.

Lorsque vous exécutez des boucles à travers une liste d'éléments, notez les informations suivantes :

- Vous devez préfixer *itérateur* avec `$` pour faire référence à l'élément en cours.

- Lorsqu'il est utilisé seul, *l'itérateur* représente le nom de l'élément en cours.

Lorsque vous exécutez des boucles à travers un tableau, notez les informations suivantes :

- Substituez *l'itérateur* avec la clé pour faire référence à un élément du tableau.
- Lorsqu'il est utilisé seul, *l'itérateur* représente une chaîne qui est la clé de l'élément de tableau en cours.

Fonctions de fichiers de règles d'analyse prises en charge

Toutes les fonctions de fichiers de règles d'analyse sont prises en charge par l'instruction FOREACH.

Restrictions

Vous ne pouvez pas utiliser l'instruction FOREACH pour itérer à travers des propriétés, des zones ou des colonnes. Vous ne pouvez pas non plus utiliser l'instruction pour itérer à travers une combinaison de tableaux et d'éléments dans la même boucle.

Si vous utilisez la fonction `details()` dans une boucle FOREACH, seuls les résultats de la dernière fonction `details()` à avoir été exécutée sont stockés dans le serveur ObjectServer. Par ailleurs, puisque l'instruction FOREACH traite les éléments dans le fichier de règles dans un ordre non déterministe, elle ne peut pas prédire quel élément sera stocké.

Exemples de fonctions permettant d'exécuter des boucles :

Ces exemples sur l'instruction en boucle FOREACH vous aideront à déployer la fonction dans votre environnement Tivoli Netcool/OMNibus.

Exemple 1 : Exécuter des boucles à travers tous les éléments

L'exemple suivant indique comment utiliser le caractère générique `$*` pour traiter tous les éléments d'une boucle :

```
foreach ( e in $* )
{
  log( INFO, "The value of $" + e + " = " + $e)
}
```

Cette instruction renverrait les messages à tous les éléments. Chaque message est similaire à l'exemple suivant :

Information: I-UNK-000-000: The value of \$DateString = 12/04/10 16:39:50.

Exemple 2 : Exécuter des boucles à travers une liste délimitée par des virgules

L'exemple suivant indique comment convertir les jetons `$Node`, `$Agent`, et `$Group` en minuscule :

```
foreach ( e in $Node, $Agent, $Group )
{
  $e = lower( $e )
}
```

Exemple 3 : Exécuter des boucles à travers les entrées d'un tableau

Dans l'exemple suivant, un tableau appelé «names» (noms) est défini au sommet du fichier de règles. Au cours d'une itération de la boucle, key (la clé) contient la clé permettant d'accéder à l'entrée en cours dans «names». Grâce à cette boucle, chaque entrée de «names» est précédée de XX.

```
array names
....
foreach ( key in names )
{
  log( INFO, "Before: The value of names[" + key + "] = " + names[key])
  names[key] = "XX" + names[key]
}
```

Exemple 4 : Utiliser les instructions IF avec les instructions d'interruption

L'exemple suivant exécute des boucles à travers tous les éléments jusqu'à ce qu'il en trouve un dont la valeur est précédée de http://. La zone URL est définie sur cette valeur et l'exécution quitte la boucle.

```
foreach (e in $* )
{
  if ( nmatch( $e, "http://" ) )
  {
    @URL = $e
    break
  }
}
```

Pour plus d'informations, voir «Instruction d'interruption», à la page 32.

Exemple 5 : Utiliser une instruction IF à l'intérieur d'une boucle FOREACH EACH dans les zones ID objet respectant le protocole SNMP

L'exemple suivant propose une solution pour gérer les différences entre les protocoles SNMP V1 and V2 :

1. Dans la première boucle, x contient toujours le nom de l'élément en cours. Si ce nom commence par OID (ID objet), la valeur de l'élément est ajoutée au tableau de travail «oids.»
2. =la clé d'une entrée est préfixée de «OID», suivi d'un numéro qui correspond au numéro utilisé dans le nom de l'élément original moins 1.
3. L'élémet original est supprimé.

Par conséquent, les éléments OID sont à présent tous décalés vers la gauche de un élément, par exemple \$OID1 = \$OID2, \$OID2 = \$OID3, etc.

Remarque : L'exemple suivant n'apporte pas de solution complète pour gérer les interruptions SNMP V2 et V1.

```
# Déclare un tableau vide
array oids

# Exécute des boucles à travers tous les éléments.
foreach ( x in $* )
{
  # Trouve les éléments dont le nom commence par 'OID'
  if( nmatch( x, "OID" ) )
  {
    # Extrait le numéro des ID objets à partir du nom de l'élément
    # Enregistre la valeur d'élément dans le tableau 'oids'.
    # oids[ "OID1" ] = $OID2
  }
}
```

```

# oids[ "OID2" ] = $OID3 etc.
$n=extract( x, "OID([0-9]+)" )
if( int( $n ) > 1 )
{
  $n=int($n)-1
  oids["OID"+$n]=$x
}
# Supprime l'élément ID objet original
remove( $x )
}
}

# Crée de nouveaux éléments 'OID'
foreach ( x in oids )
{
  $x=oids[x]
}
clear (oids)

```

Exemple 6 : Utiliser l'instruction IF à l'intérieur d'une boucle FOREACH EACH pour gérer les éléments de fonction d'intégration d'événements

L'exemple suivant indique comment utiliser l'instruction FOREACH pour supprimer les guillemets simples (') qui encadrent tous les éléments des messages de fonction d'intégration d'événements :

```

foreach ( e in $* )
{
  if(regmatch($e, "^'.*'$"))
  {
    $e = extract($e, "^'(.*?)'$")
    log(DEBUG,"Colons removed from Token " + $e)
  }
}

```

Exemple 7 : Boucles imbriquées

L'exemple suivant indique comment traduire les éléments contenant des chaînes d'octets codées (nombres entiers séparés par des points) et comment traduire les chaînes en code ASCII :

```

array octets
table Ascii2Txt =
{
  {"0", ""},
  {"9", " "},
  {"32", " "},
  {"33", "!"},
  . . .
  {"125", ""},
  {"126", "~"}
}

foreach ( e in $* )
{
  $n = split( $e, octets, "." )
  $e = ""
  foreach ( n in octets )
  {
    $e = $e + lookup( octets[n], Ascii2Txt )
  }
  clear( octets )
}

```

Exemple 8 : Utiliser l'instruction FOREACH pour effectuer une analyse syntaxique des éléments nom-valeur

Dans l'exemple suivant, les contenus de \$input représentent un ensemble de paires nom-valeur séparée par des points-virgules (;). L'exemple crée de nouveaux éléments à partir des paires nom-valeur.

```
array pairs
array values
$input="foo=blah;wibble=wobble"
$num = split( $input, pairs, ";" )
foreach ( t in pairs )
{
  $n = extract( pairs[t], "(.*)=" )
  $v = extract( pairs[t], ".*=(.*)" )
  values[ $n ] = $v
}
remove( $n )
remove( $v )
foreach ( t in values )
{
  $t = values[t]
}
```

Exemple 9 : Utiliser l'instruction FOREACH pour charger les paires nom-valeur dans la zone @ExtendedAttr

Pour créer les paires nom-valeur de tous les éléments en cours et les charger dans la zone @ExtendedAttr, utilisez l'instruction suivante :

```
@ExtendedAttr = nvp_add($*)
```

S'il s'agit seulement d'un sous-ensemble d'éléments, utilisez une instruction comme suit :

```
foreach ( e in $interface, $network, $ipaddr, $netmask, $gateway )
{
  @ExtendedAttr = nvp_add( @ExtendedAttr, e, $e )
}
```

Dans cet exemple, l'instruction exploite le fait que e représente le nom de l'élément et que \$e représente la valeur de l'élément. Cet exemple permet de remplir la zone @ExtendedAttr avec des données similaires à celles de l'échantillon suivant :

```
interface="eth0";network="178.268.2.0";ipaddr="178.268.2.64";
netmask="233.233.233.0";gateway="178.268.2.1"
```

Exemple 10 : Utiliser l'instruction FOREACH pour supprimer les paires nom-valeur de manière sélective dans la zone @ExtendedAttr

Tout comme «Exemple 9 : Utiliser l'instruction FOREACH pour charger les paires nom-valeur dans la zone @ExtendedAttr», l'instruction FOREACH peut servir à supprimer les paires nom-valeur sélectionnées dans la zone @ExtendedAttr, comme l'indique l'exemple suivant :

```
@ExtendedAttr
= nvp_add($*)
foreach ( e in $network, $netmask )
{
  @ExtendedAttr = nvp_remove( @ExtendedAttr, e )
}
```

Exemple 11 : Utiliser les instructions de commutateur et d'interruption dans une boucle FOREACH

L'exemple suivant indique comment utiliser une instruction d'interruption dans une instruction de commutateur pour interrompre le traitement d'une boucle FOREACH :

```
foreach ( x in $*)
{
  switch( $x ):
  {
    case "1":
      instructions
    case "2":
      instructions
    case "3":
      instructions
    default:
      log (ERROR, "Unexpected element $" + x + " = " + $x)
      break
  }
}
```

Pour plus d'informations, voir «Instruction de commutateur», à la page 32 et «Instruction d'interruption», à la page 32.

Référence associée:

«Exemples de fichier de règles», à la page 62

Les exemples suivants illustrent des segments types du fichier de règles.

Instruction IF

Une condition est une combinaison d'expressions et d'opérations résolues sur TRUE ou FALSE. L'instruction IF permet une exécution conditionnelle d'un ensemble d'une ou de plusieurs instructions d'affectation en exécutant uniquement les règles pour la condition TRUE.

L'instruction IF a la syntaxe suivante :

```
if (condition) {
  rules
} [ else if (condition) {
  rules
} ... ]
[ else (condition) {
  rules
} ]
```

Vous pouvez combiner des conditions en conditions de complexité croissante à l'aide de l'opérateur logique AND (&&), qui est uniquement vrai si *toutes* ses entrées sont vraies et de l'opérateur OR (|), vrai si *une* des entrées est vraie. Par exemple :

```
if match ($Enterprise, "Acme") && match ($trap-type, "Link-Up")
{ @Summary = "Acme Link Up on " + @Node }
```

Référence associée:

«Opérateurs logiques», à la page 37

Vous pouvez utiliser des opérateurs logiques sur des valeurs booléennes pour former des expressions qui sont résolues sur TRUE ou FALSE.

«Fonctions de chaîne», à la page 38

Les fonctions de chaîne vous permettent de manipuler des éléments de chaîne, généralement des noms de zones ou d'éléments.

Instruction de commutateur

Une instruction SWITCH transfère la commande à un ensemble d'une ou plusieurs instructions d'affectation de règles en fonction de la valeur d'une expression.

L'instruction de commutateur a la syntaxe suivante :

```
switch (expression) {  
  case "chaîne_littérale":  
    règles  
  case "chaîne_littérale":  
    règles  
  ...  
  default:  
    [règles]  
}
```

La variable *expression* peut être toute expression valide. Par exemple :

```
switch($node)
```

La variable *chaîne_littérale* peut être une valeur littérale quelconque. Par exemple :

```
case "jupiter":
```

Vous pouvez avoir plusieurs variables *chaîne_littérale* séparées par le symbole de barre verticale (|). Par exemple :

```
case "jupiter" | "mars" | "venus":
```

Ce cas est exécuté si l'expression correspond à l'une des chaînes spécifiées.

L'instruction de commutateur recherche uniquement des correspondances exactes. Utilisez cette instruction dès que possible à la place d'une instruction IF, car les instructions de commutateur sont traitées plus efficacement et donc plus rapidement.

Toute règle dans le chemin par défaut s'exécute si aucun autre chemin ne correspond. Chaque instruction de commutateur doit contenir un chemin par défaut, même si aucune règle ne lui est associée. Il n'est pas possible de passer d'un chemin à l'autre.

Le comportement d'une instruction BREAK dans une instruction SWITCH est identique au comportement d'une instruction BREAK à l'intérieur d'une instruction IF. Si l'instruction SWITCH est dans le corps d'une instruction en boucle, le processus quittera la boucle. Si l'instruction SWITCH ne fait pas partie du corps d'une boucle, les règles traitant l'événement sont interrompues à cet endroit et l'événement est transféré au serveur ObjectServer.

Instruction d'interruption

L'instruction d'interruption utilisée avec l'instruction FOREACH permet d'arrêter le traitement d'une boucle avant la fin de son traitement.

Une instruction d'interruption se comporte comme suit :

- Si l'instruction d'interruption est contenue dans une instruction FOREACH, alors lorsque l'instruction d'interruption est traitée, le traitement de la boucle FOREACH est immédiatement interrompu. Le traitement reprendra lors de la prochaine instruction, après l'instruction FOREACH. Si l'instruction contient des instructions FOREACH imbriquées, seule la boucle la plus au centre contenant l'instruction d'interruption est quittée.

- Si l'instruction d'interruption est à l'extérieur d'une instruction FOREACH, l'instruction d'interruption interrompt le traitement des règles de l'événement en cours. Plus aucune règle n'est traitée après l'interruption, mais l'événement est néanmoins envoyé au serveur ObjectServer (contrairement à la fonction discard).

Pour plus d'informations, voir «Exemples de fonctions permettant d'exécuter des boucles», à la page 27.

Fonctions et opérateurs de fichier de règles

Les opérateurs et fonctions vous permettent de manipuler des éléments dans les fichiers de règles avant de les affecter aux zones du serveur ObjectServer.

Le tableau suivant répertorie les opérateurs du fichier de règles.

Tableau 5. Opérateurs du fichier de règles

Opérateurs	Description	Détails supplémentaires
<code>*</code> , <code>/</code> , <code>-</code> , <code>+</code>	Effectue des opérations mathématiques et sur les chaînes.	«Opérateurs mathématiques et de chaîne», à la page 35
<code>&</code> , <code> </code> , <code>^</code> , <code>>></code> , <code><<</code>	Effectue des opérations bit à bit.	«Opérateurs de manipulation de bit», à la page 36
<code>==</code> , <code>!=</code> , <code><></code> , <code><</code> , <code>></code> , <code><=</code> , <code>>=</code>	Effectue des opérations de comparaison.	«Opérateurs de comparaison», à la page 37
NOT (NE PAS) (aussi <code>!</code>), AND (ET) (aussi <code>&&</code>), OR (OU) (aussi <code> </code>), XOR (aussi <code>^</code>)	Effectue des opérations logiques (booléennes).	«Opérateurs logiques», à la page 37

Le tableau suivant répertorie les fonctions du fichier de règles.

Tableau 6. Fonctions du fichier de règles

Nom de fonction	Description	Détails supplémentaires
<code>charcount</code>	Renvoie le nombre de caractères d'une chaîne.	«Fonctions de chaîne», à la page 38
<code>clear</code>	Supprime les éléments d'un tableau.	«Fonctions de chaîne», à la page 38
<code>datetotime</code>	Convertit une chaîne en type de données d'heure.	«Fonctions de date et d'heure», à la page 43
<code>details</code>	Ajoute des informations à la table <code>alerts.details</code> .	«Fonction détails», à la page 48
<code>discard</code>	Supprime l'intégralité d'un événement.	«Fonctions d'éléments et d'événement», à la page 38
<code>exists</code>	Test l'existence d'un élément.	«Fonction d'existence», à la page 37
<code>expand</code>	Cette fonction est obsolète et ne doit pas être utilisée comme argument d'expression régulière dans les fonctions <code>regmatch</code> ou <code>extract</code> . Plutôt que d'utiliser <code>expand</code> , placez l'expression régulière entre guillemets simples. Par exemple : <code>'[\n\r]'</code>	«Fonctions de chaîne», à la page 38
<code>extract</code>	Renvoie la partie d'une chaîne (qui peut être une zone, un élément, ou une expression de chaîne) qui correspond à la section entre parenthèses de l'expression régulière.	«Fonctions de chaîne», à la page 38

Tableau 6. Fonctions du fichier de règles (suite)

Nom de fonction	Description	Détails supplémentaires
genevent	Vous permet de : <ul style="list-style-type: none"> Créer et envoyer une alerte d'un fichier de règles à un serveur ObjectServer cible. Envoyer la même alerte à plusieurs serveurs ObjectServer ou tables. 	«Envoi d'alertes à des serveurs ObjectServer et tables alternatifs», à la page 50
getdate	Renvoie la date actuelle en tant que type de données date.	«Fonctions de date et d'heure», à la page 43
getenv	Renvoie la valeur d'une variable d'environnement.	«Fonctions de l'utilitaire hôte et processus», à la page 44
geteventcount	Renvoie le nombre d'événements dans la fenêtre d'événements.	«Surveillance des charges de la sonde», à la page 59
gethostaddr	Renvoie l'adresse IP de l'hôte en utilisant un service de dénomination.	«Fonctions de l'utilitaire hôte et processus», à la page 44
gethostname	Renvoie le nom de l'hôte en utilisant un service de dénomination.	«Fonctions de l'utilitaire hôte et processus», à la page 44
getload	Mesure la charge sur le serveur ObjectServer.	«Surveillance des charges de la sonde», à la page 59
getpid	Renvoie l'ID processus d'une sonde en cours d'exécution.	«Fonctions de l'utilitaire hôte et processus», à la page 44
getplatform	Renvoie la plateforme du système d'exploitation sur laquelle la sonde est en cours d'exécution.	«Fonctions de l'utilitaire hôte et processus», à la page 44
hostname	Renvoie le nom de l'hôte sur lequel la sonde est en cours d'exécution.	«Fonctions de l'utilitaire hôte et processus», à la page 44
int	Convertit une valeur numérique en entier.	«Fonctions mathématiques», à la page 42
length	Renvoie le nombre d'octets d'une chaîne.	«Fonctions de chaîne», à la page 38
log	Vous permet de consigner des messages.	«Fonctions de consignation de message», à la page 49
lookup	Utilise une table de consultation pour mapper des informations supplémentaires sur une alerte.	«Opérations de table de consultation», à la page 45
lower	Convertit une expression en minuscules.	«Fonctions de chaîne», à la page 38
ltrim	Supprime les espaces à gauche d'une expression.	«Fonctions de chaîne», à la page 38
match	Teste la correspondance exacte d'une chaîne.	«Fonctions de chaîne», à la page 38
nmatch	Teste une correspondance de chaîne au début d'une chaîne spécifiée.	«Fonctions de chaîne», à la page 38
nvp_add	Permet aux sondes de générer des événements qui contiennent des attributs étendus, fournis sous forme de paires nom-valeur.	«Fonctions de chaîne», à la page 38
nvp_remove	Utilisé avec des attributs étendus. Supprime les clés spécifiées d'une chaîne de paire nom-valeur et renvoie la nouvelle chaîne de paire nom-valeur.	«Fonctions de chaîne», à la page 38
printable	Convertit les caractères non imprimables d'une expression en espaces.	«Fonctions de chaîne», à la page 38
real	Convertit une valeur numérique en nombre réel.	«Fonctions mathématiques», à la page 42
recover	Récupère un événement supprimé.	«Fonctions d'éléments et d'événement», à la page 38
regmatch	Effectue une correspondance d'expression régulière complète d'une valeur dans une expression régulière d'une chaîne.	«Fonctions de chaîne», à la page 38
regreplace	Utilise les expressions régulières pour effectuer des opérations de recherche et de remplacement dans les chaînes.	«Fonction Rechercher/Remplacer», à la page 57
remove	Supprime un élément d'un événement.	«Fonctions d'éléments et d'événement», à la page 38

Tableau 6. Fonctions du fichier de règles (suite)

Nom de fonction	Description	Détails supplémentaires
registertarget	Enregistre un serveur ObjectServer pour que des alertes puissent être envoyées à plusieurs serveurs ObjectServer.	«Envoi d'alertes à des serveurs ObjectServer et tables alternatifs», à la page 50
rtrim	Supprime les espaces à droite d'une expression.	«Fonctions de chaîne», à la page 38
scanformat	Convertit une expression conformément aux formats disponibles, de manière similaire à la famille de routines scanf dans C.	«Fonctions de chaîne», à la page 38
setlog	Vous permet de définir le niveau du journal des messages.	«Fonctions de consignation de message», à la page 49
settarget, setdefaulttarget	Définit le serveur ObjectServer auquel les alertes sont envoyées.	«Envoi d'alertes à des serveurs ObjectServer et tables alternatifs», à la page 50
service	Définit le statut d'un service.	«Fonction service», à la page 58
split	Divise une chaîne en éléments d'un tableau.	«Fonctions de chaîne», à la page 38
substr	Extrait une sous-chaîne d'une expression.	«Fonctions de chaîne», à la page 38
table	Définit une table de consultation.	«Opérations de table de consultation», à la page 45
timetodate	Convertit une valeur horaire en type de données chaîne.	«Fonctions de date et d'heure», à la page 43
toBase (<i>valeur_numérique,valeur_numérique</i>)	Convertit une valeur numérique décimale en une base différente.	«Fonctions mathématiques», à la page 42
update	Indique les zones mises à jour lorsqu'une alerte est dédoublonnée.	«Fonction update sur dédoublonnage», à la page 48
updateload	Met à jour les statistiques de charge pour le serveur ObjectServer.	«Surveillance des charges de la sonde», à la page 59
upper	Convertit une expression en majuscules.	«Fonctions de chaîne», à la page 38

Opérateurs mathématiques et de chaîne

Les opérateurs mathématiques vous permettent d'ajouter, de soustraire, de diviser et de multiplier des opérands numériques dans des expressions. Les opérateurs de chaîne vous permettent de manipuler des chaînes de caractères.

Le tableau suivant décrit les opérateurs mathématiques pris en charge dans les fichiers de règles.

Tableau 7. Opérateurs mathématiques

Opérateur	Description	Exemple
* /	Opérateurs utilisés pour multiplier (*) ou diviser (/) deux opérands.	\$eventid=int(\$eventid)*2
+ -	Opérateurs utilisés pour ajouter (+) ou soustraire (-) deux opérands.	\$eventid=int(\$eventid)+1

Le tableau suivant décrit l'opérateur de chaîne pris en charge dans les fichiers de règles.

Tableau 8. Opérateur de chaîne

Opérateur	Description	Exemple
+	Concatène deux chaînes ou plus.	@field = \$element1 + "message" + \$element2

Opérateurs de manipulation de bit

Les opérateurs bit à bit vous permettent de manipuler des opérandes d'entier dans les expressions.

Le tableau suivant décrit les opérateurs bit à bit pris en charge dans les fichiers de règles.

Tableau 9. Opérateur bit à bit

Opérateur	Description	Exemple
& ^	AND (&), OR () et XOR (^) bit à bit. Les résultats sont déterminés bit par bit.	\$result1 = int(\$number1) & int(\$number2)
>> <<	Décale les bits à droite (>>) ou à gauche (<<).	\$result2 = int(\$number3) >> 1

Ces opérateurs manipulent les bits dans des expressions de type entier. Par exemple, dans l'instruction :

```
$result2 = int($number3) >> 1
```

Si number3 a la valeur 17, result2 est résolu sur 8, comme indiqué :

```

 16 8 4 2 1
>> 1 0 0 0 1
    0 1 0 0 0

```

Remarque : Les bits ne reviennent pas automatiquement à la ligne. Lorsqu'ils dépassent une extrémité, ils sont remplacés à l'autre extrémité par un 0.

Les opérateurs bit à bit fonctionnent uniquement avec des expressions de type entier. Les éléments sont stockés sous forme de chaîne, vous devez donc utiliser la fonction mathématique `int` pour les convertir en entiers avant d'effectuer ces opérations.

Pour plus d'informations sur les opérateurs bit à bit pris en charge dans la syntaxe SQL du serveur ObjectServer, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Référence associée:

«Fonctions mathématiques», à la page 42

Les fonctions mathématiques vous permettent d'effectuer des opérations numériques sur des éléments. Les éléments sont stockés sous forme de chaîne, vous devez donc utiliser ces fonctions pour les convertir en entiers avant d'effectuer des opérations numériques.

Opérateurs de comparaison

Les opérateurs de comparaison vous permettent de tester des valeurs numériques sur l'égalité et l'inégalité.

Le tableau suivant décrit les opérateurs de comparaison pris en charge dans les fichiers de règles.

Tableau 10. Opérateurs de comparaison

Opérateur	Description	Exemple
==	Tests d'égalité.	<code>int(\$eventid) == 5</code>
!=	Tests d'inégalité.	<code>int(\$eventid) != 0</code>
<>		
< > <= >=	Tests de supériorité (>), d'infériorité (<), de supériorité ou d'égalité (>=) ou d'infériorité et d'égalité (<=).	<code>int(\$eventid) <=30</code>

Opérateurs logiques

Vous pouvez utiliser des opérateurs logiques sur des valeurs booléennes pour former des expressions qui sont résolues sur TRUE ou FALSE.

Le tableau suivant décrit les opérateurs logiques pris en charge dans les fichiers de règles.

Tableau 11. Opérateurs logiques

Opérateur	Description	Exemple
NOT (aussi !)	Une expression NOT nie la valeur d'entrée et n'est définie sur TRUE que si son entrée est FALSE.	<code>if NOT(Severity=0)</code>
AND (aussi &&)	Une expression AND n'est true que si toutes ses entrées sont TRUE.	<code>if match(\$Enterprise,"Acme") && match(\$trap-type,"Link-Up")</code>
OR (aussi)	Une expression OR est TRUE si une de ses entrées est TRUE.	<code>if match(\$Enterprise,"Acme") match(\$Enterprise,"Bo")</code>
XOR (aussi ^)	Une expression XOR est TRUE si une de ses entrées, mais pas les deux, est TRUE.	<code>if match(\$Enterprise,"Acme") XOR match(\$Enterprise,"Bo")</code>

Fonction d'existence

La fonction `exists` vous permet de tester si un élément existe.

Utilisez la syntaxe suivante :

```
exists ($element)
```

La fonction renvoie TRUE si l'élément a été créé pour cet événement précis, sinon elle renvoie FALSE.

Fonctions d'éléments et d'événement

Vous pouvez utiliser des fonctions pour supprimer des éléments d'un événement, pour supprimer un événement entier et pour récupérer un événement supprimé.

Le tableau suivant décrit ces fonctions.

Tableau 12. Suppression d'éléments ou d'événements

Fonction	Description	Exemple
discard	Supprime un événement entier. Remarque : Cette fonction doit faire partie d'une instruction conditionnelle, sinon tous les événements sont supprimés.	<pre>if match(@Node,"testnode") { discard }</pre>
discarded	Teste si l'alerte en cours est marquée pour suppression. Cette condition renvoie TRUE si l'alerte est marquée pour suppression. Sinon, FALSE est renvoyé.	<pre>if(discarded) { log(DEBUG, "Alert from Node=["+@Node+"] has been marked for discard") }</pre>
recover	Récupère un événement supprimé.	<pre>if match(@Node,"testnode") { recover }</pre>
remove(<i>nom_élément</i>)	Supprime l'élément de l'événement.	<pre>remove(<i>élément_test</i>)</pre>

Fonctions de chaîne

Les fonctions de chaîne vous permettent de manipuler des éléments de chaîne, généralement des noms de zones ou d'éléments.

Le tableau suivant décrit les fonctions de chaîne prises en charge dans les fichiers de règles.

Tableau 13. Fonctions de chaîne

Fonction	Description	Exemple
charcount(<i>expression</i>)	Renvoie le nombre de caractères d'une chaîne. Remarque : Lorsque vous utilisez des caractères codés sur un octet, ce nombre est identique à celui renvoyé par la fonction length(). Si vous utilisez des caractères codés sur plusieurs octets, ce nombre peut différer de celui renvoyé par la fonction length().	<pre>\$NumChar = charcount(\$Node)</pre>
clear	Supprime les éléments d'un tableau.	<pre>clear(<i>nom_tableau</i>)</pre>

Tableau 13. Fonctions de chaîne (suite)

Fonction	Description	Exemple
<code>expand("chaîne")</code>	<p>Renvoie la chaîne (qui doit être littérale) avec des séquences d'échappement étendues. Les extensions possibles sont :</p> <ul style="list-style-type: none"> <code>\"</code> - guillemet <code>\NNN</code> - valeur octale de NNN <code>\\</code> - barre oblique inversée <code>\a</code> - alerte (BEL) <code>\b</code> - retour arrière <code>\e</code> - échappement (octal 033) <code>\f</code> - saut de page <code>\n</code> - nouvelle ligne <code>\r</code> - retour chariot <code>\t</code> - tabulation horizontale <code>\v</code> - tabulation verticale <p>Cette fonction ne peut pas être utilisée comme argument d'expression régulière dans les fonctions <code>regmatch</code> ou <code>extract</code>.</p> <p>Remarque : Cette fonction est obsolète et ne doit pas être utilisée comme argument d'expression régulière dans les fonctions <code>regmatch</code> ou <code>extract</code>. Plutôt que d'utiliser <code>expand</code>, placez l'expression régulière entre guillemets simples. Par exemple :</p> <p><code>'[\n\r]'</code></p>	<pre>log(debug, expand("Rules file with embedded \\\""))</pre> <p>envoie la sortie suivante au journal :</p> <pre>Sun Oct 21 19:56:15 2001 Debug: Rules file with embedded \"</pre>
<code>extract(chaîne, "exp_rég")</code>	<p>Renvoie la partie de la chaîne (qui peut être une zone, un élément, ou une expression de chaîne) qui correspond à la section entre parenthèses de l'expression régulière.</p>	<pre>extract (\$expr, "ab([0-9]+)cd")</pre> <p>Si <code>\$expr</code> est "ab123cd", la valeur renvoyée est 123.</p>
<code>length(expression)</code>	<p>Renvoie le nombre d'octets d'une chaîne.</p>	<code>\$NodeLength = length(\$Node)</code>
<code>lower(expression)</code>	<p>Convertit une expression en minuscules.</p>	<code>\$Node = lower(\$Node)</code>
<code>ltrim(expression)</code>	<p>Supprime les espaces à gauche d'une expression.</p>	<code>\$TrimNode = ltrim(\$Node)</code>
<code>match(expression, "chaîne")</code>	<p>TRUE si la valeur de l'expression correspond exactement à la chaîne.</p>	<code>if match(\$Node, "New")</code>
<code>nmatch(expression, "chaîne")</code>	<p>TRUE si l'expression commence par la chaîne spécifiée.</p>	<code>if nmatch(\$Node, "New")</code>

Tableau 13. Fonctions de chaîne (suite)

Fonction	Description	Exemple
<p><code>nvp_add(chaîne_pvn, \$nom, \$valeur [, \$nom2, \$valeur2,]*)</code></p> <p><code>nvp_add(\$*)</code></p>	<p>Crée ou met à jour une chaîne de paires nom-valeur d'attributs étendus. Plusieurs paires nom-valeur peuvent être fournies pour la chaîne. Des variables et leurs valeurs peuvent être ajoutées à, ou remplacées dans la chaîne de paires nom-valeur.</p> <p>Crée une chaîne de paires nom-valeur de toutes les variables et leurs valeurs si elle est appelée en tant que <code>nvp_add(\$*)</code>.</p>	<pre>if (int(\$PercentFull) > 95) { @Severity = 5 @ExtendedAttr = nvp_add(@ExtendedAttr, "PercentFull", \$PercentFull, "Disk", \$Disk) }</pre> <p>Si \$PercentFull est 97 et \$Disk est /dev/sfa1, @ExtendedAttr sera (en supposant qu'il était initialement vide) : PercentFull="97";Disk="/dev/sfa1"</p>
<p><code>nvp_remove(chaîne_pvn, chaîne_clé1 [, chaîne_clé2, [...]])</code></p>	<p>Utilisé avec des attributs étendus. Supprime les clés spécifiées d'une chaîne de paire nom-valeur et renvoie la nouvelle chaîne de paire nom-valeur.</p> <p>Utile lorsque la liste d'attributs étendus à inclure est plus longue que celle des attributs à exclure. Vous pouvez utiliser <code>nvp_add(\$*)</code> pour inclure toutes les variables et leurs valeurs puis utiliser <code>nvp_remove</code> pour supprimer des variables et des valeurs spécifiques.</p>	<pre>\$interface = "eth0" \$network = "178.268.2.0" \$ipaddr = "178.268.2.64" \$netmask = "233.233.233.0" \$gateway = "178.268.2.1"</pre> <pre>@ExtendedAttr = nvp_add(\$*) @ExtendedAttr = nvp_remove(@ExtendedAttr, "network", "netmask")</pre> <p>Par conséquent, @ExtendedAttr est : interface="eth0";ipaddr="178.268.2.64"; gateway="178.268.2.1"</p>
<p><code>printable(expression)</code></p>	<p>Convertit les caractères non imprimables d'une expression donnée en espaces.</p>	<pre>\$Print = printable(\$Node)</pre>
<p><code>regmatch(expression, "exp_rég")</code></p>	<p>Mise en correspondance complète d'une expression régulière.</p>	<pre>if (regmatch(\$enterprise, "^Acme Config:[0-9]"))</pre>
<p><code>regreplace(expression, "regexp", chaîne [, nombre])</code></p>	<p>Utilise une expression régulière et un remplacement permettent pour effectuer des opérations de recherche et de remplacement sur une expression de chaîne en entrée.</p>	<pre>\$result = regreplace(\$input, "(%)", "")</pre>
<p><code>rtrim(expression)</code></p>	<p>Supprime les espaces à droite d'une expression.</p>	<pre>\$TrimNode = rtrim(\$Node)</pre>

Tableau 13. Fonctions de chaîne (suite)

Fonction	Description	Exemple
<code>scanformat(expression, "chaîne")</code>	<p>Convertit l'expression conformément aux formats suivants, de manière similaire à la famille de routines scanf dans C. Les spécifications de conversion sont :</p> <p><code>%%</code> - littéral <code>%</code>; ne pas interpréter</p> <p><code>%d</code> - correspond à un entier décimal signé</p> <p><code>%u</code> - identique à <code>%d</code> ; aucune vérification n'est effectuée pour la signature</p> <p><code>%o</code> - correspond à un numéro octal à signature facultative</p> <p><code>%x</code> - correspond à un nombre hexadécimal à signature facultative</p> <p><code>%i</code> - correspond à un entier à signature facultative</p> <p><code>%e, %f, %g</code> - correspond à un nombre en virgule flottante à signature facultative</p> <p><code>%s</code> - correspond à une chaîne terminée par un espace ou une fin de chaîne</p>	<p><code>\$element = "Lou is up in 15 seconds"</code></p> <p><code>[\$node, \$state, \$time] = scanformat(\$element, "%s is %s in %d seconds")</code></p> <p>Ces exemples définissent <code>\$node</code>, <code>\$state</code>, et <code>\$time</code> sur Lou, up, et 15.</p>
<code>nb_zones_renvoyées = split("chaîne", tableau_destination, "séparateur_zone")</code>	<p>Divise la chaîne spécifiée en éléments du tableau de destination.</p> <p>Le séparateur de zones sépare les éléments. Le séparateur lui-même n'est pas renvoyé. Si vous spécifiez plusieurs caractères dans un séparateur de zones, une séparation sera effectuée si une combinaison d'un ou de plusieurs caractères est détectée dans la chaîne.</p> <p>Les expressions régulières sont interdites dans la chaîne ou dans le séparateur de zones.</p>	<p><code>\$num_elements=split("bilbo: frodo:gandalf",names,":")</code></p> <p>crée un tableau avec trois entrées :</p> <p><code>names[1] = bilbo</code></p> <p><code>names[2] = frodo</code></p> <p><code>names[3] = gandalf</code></p> <p><code>num_elements</code> est défini sur 3.</p> <p>Vous devez définir le tableau <code>names</code> au début du fichier de règles avant les instructions de traitement.</p>
<code>substr(expression,n, len)</code>	<p>Extrait une sous-chaîne en démarrant à la position spécifiée par le deuxième paramètre et pour le nombre de caractères spécifié par le troisième paramètre.</p>	<p><code>\$Substring = substr(\$Node,2,10)</code></p> <p>extrait 10 caractères à partir de la deuxième position de l'élément <code>\$Node</code>.</p>
<code>upper(expression)</code>	<p>Convertit une expression en majuscules.</p>	<p><code>\$Node = upper(\$Node)</code></p>

Concepts associés:

Annexe C, «Expressions régulières», à la page 345

Tivoli Netcool/OMNIBus prend en charge l'utilisation des expressions régulières dans les demandes de recherche que vous réalisez sur le serveur ObjectServer. Les expressions régulières sont des séquences d'*atomes* composées de caractères normaux et de métacaractères.

Référence associée:

«Utilisation de tableaux», à la page 25

Vous devez définir des tableaux au début d'un fichier de règles avant d'indiquer des instructions de traitement.

«Fonction Rechercher/Remplacer», à la page 57

Utilisez la fonction `regrepl` pour effectuer des opérations de recherche et de remplacement sur des chaînes à l'aide d'expressions régulières.

Fonctions mathématiques

Les fonctions mathématiques vous permettent d'effectuer des opérations numériques sur des éléments. Les éléments sont stockés sous forme de chaîne, vous devez donc utiliser ces fonctions pour les convertir en entiers avant d'effectuer des opérations numériques.

Le tableau suivant décrit les fonctions mathématiques prises en charge dans les fichiers de règles.

Tableau 14. Fonctions mathématiques

Fonction	Description	Exemple
<code>int(<i>numérique</i>)</code>	Convertit une valeur numérique en entier.	<code>if int(\$PercentFull) > 80</code>
<code>real(<i>valeur_numérique</i>)</code>	Convertit une valeur numérique en nombre réel.	<code>@DiskSpace= (real(\$diskspace)/real(\$total))*100</code>
<code>toBase(<i>base,valeur</i>)</code>	Convertit une valeur numérique décimale en une base différente.	<code>toBase(2,16)</code> renvoie 10000 <code>toBase(16,14)</code> renvoie E <code>toBase(16,\$a)</code> renvoie la valeur de l'élément \$a converti en base 16

Exemple : définition de la gravité d'une alerte en fonction de l'espace disque disponible

Dans l'exemple suivant, la gravité d'une alerte qui surveille l'utilisation de l'espace disque est définie en fonction de la quantité d'espace disque disponible.

```
if (int($PercentFull) > 80 && int($PercentFull) <=85)
{
  @Severity=2
}
else if (int($PercentFull) > 85 && int($PercentFull) <=90)
{
  @Severity=3
}
else if (int($PercentFull) > 90 && int($PercentFull) <=95)
{
  @Severity=4
}
else if (int($PercentFull) > 95)
{
  @Severity=5
}
```

Exemple : calcul de la quantité d'espace disque

Le pourcentage d'espace disque n'est pas toujours fourni dans le flux d'événements. Vous pouvez calculer ce pourcentage dans le fichier de règles comme suit :

```
if (int($total) > 0)
{
  @DiskSpace=(100*int($diskspace))/int($total)
}
```

Il peut également être calculé avec la fonction `real` :

```
if (int($total) > 0)
{
  @DiskSpace=(real($diskspace)/real($total))*100
}
```

Vous pouvez ensuite définir la gravité d'une alerte, comme indiqué dans l'exemple précédent.

Fonctions de date et d'heure

Vous pouvez utiliser des fonctions de date et d'heure pour obtenir l'heure actuelle ou pour effectuer des conversions de date et d'heure.

Les heures sont exprimées en temps UNIX (nombre de secondes écoulées depuis le 1 janvier 1970 UTC à minuit). Le tableau suivant décrit les fonctions de date et d'heure prises en charge dans les fichiers de règles.

Tableau 15. Fonctions de date et d'heure

Fonction	Description	Exemple de fichier de règles
<code>datetotime(<i>chaîne</i>, <i>spécification_conversion</i>)</code>	Convertit une représentation textuelle d'un horodatage en heure d'époque UNIX (le nombre de secondes depuis 00:00:00 1 Jan 1970 UTC).	<code>\$Date = datetotime("Tue Dec 19 18:33:11 GMT+00:00 2000", "EEE MMM dd HH:mm:ss ZZZ yyyy")</code>
<code>getdate</code>	N'a aucun argument et renvoie la date actuelle sous forme de date.	<code>\$tempdate = getdate</code>
<code>timetodate(UTC, <i>spécification_conversion</i>)</code>	Convertit une valeur temporelle en chaîne.	<code>@Summary = "Occurred at " + timetodate (\$StateChange, "HH:mm:ss, MM/dd/yy")</code> <code>\$Time2 = timetodate (@EventTime, 'EEE MMM dd HH:mm:ss yyyy')</code>

Le paramètre *spécification_conversion* des fonctions `datetotime` et `timetodate` est le format de date et d'heure dans lequel la conversion doit être exprimée. Le tableau suivant présente des exemples de données d'entrée et de sortie correspondant à certains formats de date et d'heure possibles.

Remarque : Les formats de date et d'heure POSIX sont obsolètes dans Tivoli Netcool/OMNIbus V7.3 et remplacés par des modèles de date et d'heure LDML (Locale Data Markup Language). Les modèles de date et d'heure LDML sont définis sur le site <http://userguide.icu-project.org/formatparse/datetime>.

Tableau 16. Quelques formats de date et d'heure LDML et données d'entrée et de sortie correspondantes

Format	Exemple de données d'entrée et de sortie
MM/dd/yy	02/29/12
MMMM	Décembre
d/M/yyyy H:m:s	19/12/2000 18:33:11
EEE MMM dd HH:mm:ss ZZZZ yyyy	Mardi 19 déc 17:33:11 GMT 2000
yyyy-MM-dd:hh:mma v	2009-03-28:02:00PM United Kingdom Time
EEE MMM dd HH:mm:ss yyyy ZZZ	Sun Jan 15 08:30:00 2006 +0500

Pour plus d'informations sur les formats de date et d'heure LDML utilisés dans Tivoli Netcool/OMNIBus, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIBus*.

Fonctions de l'utilitaire hôte et processus

Les fonctions d'utilitaire vous permettent d'obtenir des informations sur l'environnement dans lequel la sonde est exécutée.

Le tableau suivant décrit les fonctions d'hôte et de processus prises en charge dans les fichiers de règles.

Tableau 17. Fonctions de l'utilitaire hôte et processus

Fonction	Description	Exemple
getenv(<i>chaîne</i>)	Renvoie la valeur d'une variable d'environnement spécifiée.	\$My_OMNIHOME = getenv("OMNIHOME")
gethostaddr(<i>chaîne</i>)	Renvoie l'adresse IP de l'hôte à l'aide d'un service de dénomination (par exemple, DNS ou /etc/hosts). L'argument peut être une chaîne contenant un nom d'hôte ou une adresse IP. Si l'hôte ne peut pas être recherché, la valeur initiale est renvoyée. Remarque : La recherche DNS (et d'autres services similaires) peut durer assez longtemps, ce qui peut avoir une incidence importante sur la performance de la sonde. Vous devriez envisager d'utiliser une table de consultation dans le fichier de règles à la place et de n'utiliser gethostaddr que si l'hôte n'est pas dans la table.	@Summary = \$Summary + " Node: " + \$Node + " Address: " + gethostaddr(\$Node)

Tableau 17. Fonctions de l'utilitaire hôte et processus (suite)

Fonction	Description	Exemple
gethostname(<i>chaîne</i>)	<p>Renvoie le nom de l'hôte à l'aide d'un service de dénomination (par exemple, DNS ou /etc/hosts). L'argument peut être une chaîne contenant un nom d'hôte ou une adresse IP. Si l'hôte ne peut pas être recherché, la valeur initiale est renvoyée.</p> <p>Remarque : La recherche DNS (et d'autres services similaires) peut durer assez longtemps, ce qui peut avoir une incidence importante sur la performance de la sonde. Vous devriez envisager d'utiliser une table de consultation dans le fichier de règles à la place et de n'utiliser gethostname que si l'hôte n'est pas dans la table.</p>	<pre>@Summary = \$Summary + " Node: " + \$Node + " Name: " + gethostname(\$Node)</pre>
getpid()	Renvoie l'ID processus de la sonde en cours d'exécution.	<pre>\$My_PID = getpid()</pre>
getplatform()	Renvoie la plateforme du système d'exploitation sur laquelle la sonde est en cours d'exécution. Une des valeurs suivantes est renvoyée : linux2x86, solaris2, hpux11, aix5, ou win32.	<pre>log(INFO, "Netcool Platform = " + getplatform())</pre>
hostname()	Renvoie le nom de l'hôte sur lequel la sonde est en cours d'exécution.	<pre>\$My_Hostname = hostname()</pre>

Opérations de table de consultation

Les tables de consultation permettent d'ajouter des informations supplémentaires à un événement. Une table de consultation est constituée d'une liste de clés et de valeurs.

Vous définissez une table de consultation à l'aide de la fonction `table` et vous y accédez à l'aide de la fonction `lookup`.

La fonction `lookup` évalue l'expression dans les clés de la table désignée et renvoie la valeur associée. Si la clé est introuvable, une chaîne vide est renvoyée. La fonction `lookup` a la syntaxe suivant :

```
lookup(expression,nom_table)
```

Vous pouvez créer une table de consultation dans le fichier de règles ou dans un fichier distinct.

Remarque : Si un fichier de table de consultation a plusieurs colonnes, chaque ligne doit avoir le même nombre de colonnes. Toutes les lignes qui ont un nombre de colonnes incorrect sont supprimées. En mode colonne unique, seule la première tabulation est significative, toutes les autres sont lues comme partie de la valeur unique définie sur cette ligne.

Définition de tables de recherche dans le fichier de règles :

Vous pouvez créer une table de consultation directement dans le fichier de règles.

Pourquoi et quand exécuter cette tâche

Les définitions de tables de recherches doivent se trouver au début d'un fichier de règles, *après* toutes les instructions `registertarget`, mais *avant* toute instruction de traitement. Une table de consultation peut disposer de plusieurs colonnes. Vous pouvez également définir plusieurs tables de recherche dans un fichier de règles. Pour que les modifications apportées à la table de consultation soient prises en compte, la sonde doit être forcée à relire le fichier de règles.

Pour créer une table de consultation :

Procédure

1. Ouvrez le fichier de règles de la sonde.
2. A la suite des instructions `registertarget`, ajoutez l'entrée de définition de table pertinente pour une table de consultation portant le nom `nom_table` :
 - a. Pour créer la table de consultation avec une liste de clés et de valeurs, utilisez le format suivant :
 - b. Pour créer la table de consultation avec plusieurs colonnes, utilisez le format suivant :
 - c. Pour créer la table de consultation et indiquer une option par défaut pour gérer un événement ne correspondant à aucune des valeurs de clés de la table, utilisez le format suivant :

```
table tablename={{"clé","valeur"},"clé","valeur"}...}
```

```
table tablename={{"clé1", "valeur1", "valeur2", "valeur3"}, {"clé2", "val1", "val2", "val3"}}
```

```
table tablename={{"clé1", "valeur1", "valeur2", "valeur3"}, {"clé2", "val1", "val2", "val3"}}
default = {"defval1", "defval2", "defval3"}
```

```
table tablename=
{"clé1", "valeur1", "valeur2", "valeur3"},
{"clé2", "val1", "val2", "val3"}}
default = {"defval1", "defval2", "defval3"}
```

Remarque : L'instruction par défaut doit suivre la définition de table spécifique.

Exemple

Par exemple, pour créer une table de consultation nommée `dept`, qui met en correspondance un nom de noeud et le service auquel appartient ce noeud, ajoutez la ligne suivante dans le fichier de règles :

```
table dept={{"node1","Technical"},"node2","Finance"}}
```

Pour accéder à cette table de consultation dans le fichier de règles, procédez comme suit :

```
@ExtraChar=lookup(@Node,dept)
```

Cet exemple utilise la zone `@Node` comme la clé. Si la valeur de la zone `@Node` correspond à une clé de la table, `@ExtraChar` est défini sur la valeur correspondante.

Pour obtenir des valeurs à partir d'une table de consultation contenant plusieurs valeurs, procédez comme suit :

```
[@Summary, @AlertKey, $error_code] = lookup("key1", tablename)
```

Tâches associées:

«Relecture du fichier de règles», à la page 66

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Définition de tables de consultation dans un fichier distinct :

Vous pouvez créer une table dans un fichier distinct, comme alternative à la création directe de la table dans le fichier de règles.

Remarque : Toutes les entrées des fichiers de recherche externes doivent se terminer par un caractère de retour à la ligne.

Si vous spécifiez une seule valeur, le fichier doit être au format :

```
key[TAB]valeur  
key[TAB]valeur
```

Pour des valeurs multiples, le format est :

```
clé1[TAB]valeur1[TAB]valeur2[TAB]valeur3  
clé2[TAB]val1[TAB]val2[TAB]val3
```

Vous pouvez spécifier une option par défaut pour traiter un événement qui ne correspond à aucune valeur de clé de la table. L'instruction par défaut doit suivre la définition de table spécifique. L'exemple suivant correspond à une table dans un fichier distinct :

```
table dept="$OMNIHOME/probes/solaris2/Dept"  
default = {"valdéf1", "valdéf2", "valdéf3"}
```

Par exemple, pour créer une table dans laquelle le nom de poste correspond au service dans lequel il se trouve, utilisez le format suivant :

```
node1[TAB]"Technical"  
node2[TAB]"Finance"
```

Spécifiez le chemin d'accès à la table de consultation comme chemin d'accès absolu ou relatif. Les chemins relatifs commencent à partir du répertoire du fichier de règles actuel. Vous pouvez utiliser des variables d'environnement dans le chemin. Par exemple :

```
table dept="$OMNIHOME/probes/solaris2/Dept"
```

Vous pouvez ensuite utiliser cette table de consultation dans le fichier de règles de la manière suivante :

```
@ExtraChar=lookup(@Node,dept)
```

Vous pouvez également contrôler la manière dont la sonde contrôle les tables de consultation externes à l'aide de la propriété **LookupTableMode**. Cette propriété détermine comme les erreurs sont gérées lorsque des tables de consultation externes n'ont pas la même valeur sur chaque ligne.

Fonction update sur dédoublement

Le serveur ObjectServer gère le processus de dédoublement, mais vous pouvez également le configurer dans le fichier de règles de sonde. La fonction update vous permet de spécifier les zones d'une alerte qui sont mises à jour si l'alerte est dédoublement. Les règles de dédoublement peuvent ainsi être définies alerte par alerte.

La fonction update peut activer la mise à jour du dédoublement dans les zones qui n'ont pas été définies pour être mises à jour dans le déclencheur du dédoublement. Vous ne pouvez pas utiliser la fonction update pour redéfinir le déclencheur du dédoublement afin d'éviter que les zones soient mises à jour.

La syntaxe de la fonction update est la suivante :

```
update(nom_zone [, TRUE | FALSE ] )
```

Si elle est définie sur TRUE, la mise à jour sur dédoublement est activée. En cas de définition sur FALSE, la mise à jour sur dédoublement est désactivée. La valeur par défaut est FALSE.

Par exemple, pour vous assurer que la zone Severity (Gravité) est mise à jour lors du dédoublement, ajoutez l'entrée suivante au fichier de règles :

```
update(@Severity)
```

L'exemple suivant indique comment désactiver les mises à jour du dédoublement dans le fichier de règles pour une zone précédemment activée :

```
update(@Severity, FALSE)
```

Si, dans le déclencheur du dédoublement, la zone est définie pour être mise à jour, définir la fonction update sur FALSE n'aura aucun effet.

Fonction détails

Les détails sont des éléments supplémentaires créés par une sonde pour afficher des informations d'alerte qui ne sont pas stockées dans une zone de la table alerts.status. Alerts ne contient pas d'informations détaillées, sauf si ces informations sont ajoutées.

Les éléments de détails sont stockés dans la table de détails alerts.details du serveur ObjectServer. Pour afficher les détails, cliquez deux fois sur une alerte et sélectionnez **Détails (Détails)**.

Vous pouvez ajouter des informations à la table de détails à l'aide de la fonction details. Les informations détaillées sont ajoutées lors de l'insertion d'une alerte, mais pas si elle est dédoublement.

L'exemple suivant ajoute les éléments \$a et \$b à la table alerts.details :

```
details($a,$b)
```

L'exemple suivant ajoute toutes les informations d'alertes dans la table alerts.details :

```
details($*)
```

Avertissement : Vous devez uniquement utiliser \$* lors du débogage ou de l'écriture des fichiers de règles. Après l'utilisation de \$* pendant de longues périodes, les tables du serveur ObjectServer deviennent très volumineuses et les performances du serveur en sont affectées.

Exemple : utilisation de la fonction details

Dans cet exemple, l'élément \$Summary est comparé aux chaînes Incoming et Backup. S'il n'y a aucune correspondance, la zone @Summary est définie sur la chaîne Please see details et toutes les informations de l'alerte sont ajoutées à la table des détails :

```
if (match($Summary, "Incoming"))
{
  @Summary = "Received a call"
}
else if(match($Summary, "Backup"))
{
  @Summary = "Attempting to back up"
}
else
{
  @Summary = "Please see details"
  details($*)
}
```

Fonctions de consignation de message

La fonction log vous permet de consigner des messages pendant le traitement des règles. Vous pouvez également définir un niveau de consignation en utilisant la fonction setlog et seuls les messages de niveau équivalent ou supérieur seront consignés.

Il existe cinq niveaux de consignation : DEBUG, INFO, WARNING, ERROR, et FATAL, par ordre de gravité croissant. Par exemple, si vous définissez le niveau de consignation sur WARNING, seuls les messages WARNING, ERROR et FATAL sont consignés, mais si vous définissez cette consignation sur ERROR, seuls les messages ERROR et FATAL sont consignés.

Fonction log :

La fonction log envoie un message au fichier journal.

La syntaxe est :

```
log([ DEBUG | INFO | WARNING | ERROR | FATAL ],"string")
```

Remarque : Lorsqu'un message FATAL est consigné, la sonde s'arrête.

Fonction setlog :

La fonction setlog définit le niveau minimum auquel les messages sont consignés pendant le traitement des règles. Le niveau de consignation par défaut est WARNING et supérieur.

La syntaxe est :

```
setlog([ DEBUG | INFO | WARNING | ERROR | FATAL ])
```

Exemple : consignation de messages :

Les lignes suivantes présentent une séquence de fonctions de consignation du fichier de règles :

```
setlog(WARNING)
log(DEBUG,"A debug message")
log(WARNING,"A warning message")
setlog(ERROR)
log(WARNING,"Another warning message")
log(ERROR,"An error message")
```

Cet exemple produit la sortie de journal suivante :

```
A warning message
An error message
```

Le message de niveau DEBUG n'est pas consigné car le paramètre de consignation est défini sur un niveau supérieur à DEBUG. Le deuxième message de niveau WARNING n'est pas consigné car la fonction `setlog` précédente a défini le niveau de consignation sur un niveau supérieur à WARNING.

Envoi d'alertes à des serveurs ObjectServer et tables alternatifs

Les fonctions `registertarget`, `genevent`, `settarget` et `setdefaulttarget` permettent d'envoyer des alertes à un ou plusieurs serveurs ObjectServer et de définir la distribution de ces alertes entre ces serveurs.

Enregistrement de serveurs ObjectServer cible et définition de cibles pour les alertes :

Pour enregistrer un serveur ObjectServer et une table des alertes sur le serveur ObjectServer auquel vous souhaitez envoyer des événements (désigné par *serveur ObjectServer cible*), utilisez la fonction `registertarget`. Si nécessaire, vous pouvez utiliser cette fonction pour spécifier plusieurs serveurs ObjectServer cible et les tables d'alertes correspondantes. Utilisez la fonction `setdefaulttarget` pour spécifier un autre serveur ObjectServer cible et une autre table par défaut. Pour spécifier un serveur ObjectServer cible alternatif autre que le serveur par défaut, utilisez la fonction `settarget`.

Enregistrez tous les serveurs ObjectServer cible au début du fichier de règles, avant toute instruction de traitement ou table de recherche.

Sur chaque serveur ObjectServer cible, chaque alerte n'est généralement envoyée qu'à une seule table d'alertes. Facultativement, l'alerte peut aussi être envoyée à une table de détails correspondante. Des exceptions se produisent lorsque la fonction `genevent` est utilisée.

Format de la fonction `registertarget`

Le format de la fonction `registertarget` est le suivant :

```
cible = registertarget(nom_serveur, nom_serveur_sauvegarde, "table_base_de_données"
[, "table_détails" ] )
```

Dans cette instruction :

- *cible* est un libellé qui identifie le serveur ObjectServer cible. Ce libellé doit être unique parmi tous les serveurs ObjectServer cible enregistrés. Un libellé peut indiquer le type ou la distribution des alertes à envoyer au serveur ObjectServer.

Exemples : NCOMSalerts, FloodProtectionActiveAlert, HighAlerts, StatsInfoAlert. La cible est utilisée comme argument dans les fonctions `genevent`, `settarget` et `setdefaulttarget`.

- *nom_serveur* est le nom du serveur ObjectServer cible.
- *nom_serveur_sauvegarde* est le nom du serveur ObjectServer cible de sauvegarde dans la paire de reprise en ligne (si une paire de reprise en ligne est configurée).
- *table_base_de_données* est le nom d'une table de base de données valide dans laquelle vous souhaitez que les données d'alerte soient insérées. Indiquez cette valeur entre guillemets (" ").
- *table_détails* est le nom d'une table de détails de base de données valide dans laquelle vous souhaitez que les données d'alerte soient insérées. Indiquez cette valeur entre guillemets (" ").

Les valeurs *nom_serveur* et *nom_serveur_sauvegarde* peuvent avoir l'un des formats suivants :

- Chaînes placées entre guillemets (" "), par exemple "Server" ou "ServerBackup".
- Valeur de la propriété précédée du signe pourcentage (%), par exemple %Server ou %ServerBackup.

Pour omettre le serveur ObjectServer de sauvegarde, spécifiez *nom_serveur_sauvegarde* en tant que chaîne vide : "".

Instructions d'utilisation

La fonction `registertarget` requiert la même autorisation utilisateur pour tous les serveurs ObjectServer référencés. Lorsque vous vous connectez à plusieurs serveurs ObjectServer sécurisés, une sonde utilise les données d'identification définies par ses propriétés **AuthUserName** et **AuthPassword** pour tous les serveurs ObjectServer cible. Il est également nécessaire que le compte utilisateur dispose des droits d'autorisation adéquats pour toutes les tables cible auxquelles la sonde envoie des données d'événement.

La première instruction `registertarget` d'un fichier de règles de sonde définit le serveur ObjectServer cible par défaut. Ce serveur ObjectServer remplace tout serveur ObjectServer cible spécifié en exécutant la sonde avec l'option de ligne de commande `-server`. Par exemple, si un fichier de règles de sonde possède une seule instruction `registertarget` qui enregistre TEST1 comme serveur ObjectServer cible par défaut, si vous exécutez ensuite la sonde avec l'option `-server` définie sur TEST2, les alertes sont envoyées sur TEST1.

Si vous souhaitez insérer des informations détaillées supplémentaires pour l'alerte, utilisez la fonction `details` pour spécifier ces informations dans le fichier de règles.

Si un fichier de règles est mis en cache, les noms des serveurs ObjectServer cible sont toujours écrits dans le fichier de cache sous forme de chaînes. Si vous avez enregistré les serveurs ObjectServer cible par les noms de zone %Server et %ServerBackup, les valeurs de ces zones sont également résolues dans le fichier de cache sous forme de chaînes. Par exemple, si le serveur ObjectServer cible s'appelle NCOMS, l'instruction `registertarget alerts = registertarget(%Server,"", "alerts.status","alerts.details")` est copiée dans le fichier de cache sous la forme `alerts = registertarget("NCOMS","", "alerts.status","alerts.details")`.

Exemple 1 : Enregistrement de plusieurs serveurs ObjectServer cible

L'exemple suivant présente la fonction `registertarget` utilisée pour enregistrer plusieurs serveurs ObjectServer cible. Dans cet exemple, les alertes sont insérées dans la table `alerts.status` du serveur ObjectServer TEST1, sauf si les paramètres sont remplacés par un appel ultérieur aux fonctions `setdefaulttarget` ou `settarget`.

```
DefaultAlerts = registertarget( "TEST1", "", "alerts.status" )
HighAlerts = registertarget( "TEST2", "", "alerts.status" )
ClearAlerts = registertarget( "TEST3", "", "alerts.status" )
London = registertarget( "NCOMS", "NCOMSBACK", "alerts.london" )
MasterStats = registertarget( %Server, %ServerBackup, "master.stats" )
```

Exemple 2 : Spécification d'un serveur ObjectServer cible par défaut différent

L'exemple suivant présente la commande `setdefaulttarget` utilisée pour modifier le serveur ObjectServer cible par défaut pour les alertes qui possèdent des niveaux de gravité spécifiques.

```
# When an event of Major severity or higher comes in,
# set the default ObjectServer to TEST2
if(int(@Severity) > 3)
{ setdefaulttarget(HighAlerts) }
```

Exemple 3 : Spécification d'un serveur ObjectServer cible alternatif

L'exemple suivant présente la commande `settarget` utilisée pour modifier le serveur ObjectServer cible pour les alertes qui possèdent un niveau de gravité spécifique.

```
# Send all clear events to TEST3
if (int(@Severity) = 0)
{ settarget(ClearAlerts) }
```

Tâches associées:

«Activation de la mise en cache des fichiers de règles de sonde», à la page 69
Pour vous assurer qu'une sonde est toujours en mesure de lire un ensemble valide de règles lorsqu'elle est démarrée, activez la mise en cache du fichier de règles. Par défaut, la mise en cache du fichier de règles est désactivée.

Référence associée:

«Fonction détails», à la page 48

Les détails sont des éléments supplémentaires créés par une sonde pour afficher des informations d'alerte qui ne sont pas stockées dans une zone de la table `alerts.status`. `Alerts` ne contient pas d'informations détaillées, sauf si ces informations sont ajoutées.

«Opérations de table de consultation», à la page 45

Les tables de consultation permettent d'ajouter des informations supplémentaires à un événement. Une table de consultation est constituée d'une liste de clés et de valeurs.

«Envoi d'alertes à plusieurs serveurs ObjectServer et tables», à la page 53

Si vous souhaitez envoyer la même alerte à plusieurs serveurs ObjectServer enregistrés ou à plusieurs tables, vous devez utiliser la fonction `genevent`.

Envoi d'alertes à plusieurs serveurs ObjectServer et tables :

Si vous souhaitez envoyer la même alerte à plusieurs serveurs ObjectServer enregistrés ou à plusieurs tables, vous devez utiliser la fonction `genevent`.

Voici quelques scénarios d'utilisation de la fonction `genevent` :

- Vous souhaitez configurer le fichier de règles de sonde pour détecter une condition de débordement d'événements et supprimer temporairement les données d'alertes envoyées au serveur ObjectServer. Vous pouvez utiliser la fonction `genevent` pour envoyer au serveur ObjectServer une alerte d'information au début du débordement d'événements et lorsque le débordement d'événements est terminé.
- Vous devez séparer les alertes de traitement de priorité élevée et les alertes d'information de priorité faible à la source et les traiter différemment. Vous pouvez utiliser la fonction `genevent` pour envoyer des alertes de priorité élevée à un serveur ObjectServer prioritaire et à un serveur ObjectServer qui met en corrélation et archive toutes les alertes. Vous pouvez également envoyer les alertes de priorité faible uniquement au serveur ObjectServer qui met en corrélation et archive toutes les alertes.
- Vous avez besoin d'une analyse statistique des données d'alertes entrantes mais ne souhaitez pas augmenter la charge sur le serveur ObjectServer qui reçoit les événements. La fonction `genevent` vous permet d'envoyer des informations statistiques dérivées des données d'alerte entrantes à un autre ObjectServer pour effectuer une analyse ultérieure.
- Vous souhaitez dupliquer toutes les données d'alerte sur deux serveurs ObjectServer ou plus pour qu'ils puissent effectuer différentes opérations sur les données. Vous voulez également supprimer le temps système de l'exécution d'une passerelle unidirectionnelle entre les serveurs ObjectServer. Vous pouvez utiliser la fonction `genevent` pour envoyer les données d'alerte à tous les serveurs ObjectServer. Notez toutefois que ce type d'utilisation n'est *pas* prévu comme remplacement de l'utilisation d'une passerelle dans une paire de reprise en ligne car les alertes dupliquées ne seront pas correctement associées.

Le format de la fonction `genevent` est le suivant :

```
genevent(cible[, identificateur_colonne, valeur_colonne, ...])
```

Dans cette instruction :

- *cible* est la valeur que vous avez indiquée pour *cible* dans l'instruction `registertarget` concernée.
- *identificateur_colonne* et *valeur_colonne* représentent des paires *nom_valeur*, où *identificateur_colonne* est une zone ObjectServer valide dans la table dans laquelle l'alerte doit être insérée et *valeur_colonne* est la valeur de données que vous souhaitez insérer. La valeur *identificateur_colonne* doit avoir comme préfixe le symbole `@` pour indiquer une zone ObjectServer ; par exemple `@Summary`. La *valeur_colonne* peut être une valeur statique ou une expression résolue lors du traitement du fichier de règles ; par exemple `$Summary + $Group`. Si *valeur_colonne* est une valeur de chaîne, elle doit être incluse entre guillemets.

Conseil : Lors de la spécification de *valeur_colonne*, utilisez un type de données approprié à la zone du serveur ObjectServer. Dans Netcool/OMNIBus Administrator, vous pouvez utiliser le panneau Databases, Tables and Columns (Bases de données, tables et colonnes) pour vérifier les types de données qui sont affectés aux zones. Vous pouvez également utiliser la commande SQL DESCRIBE du serveur ObjectServer.

Remarque : Lorsque le fichier de règles est traité, une conversion du type de données est tentée sur une valeur de colonne en cas de non concordance entre l'identificateur de la colonne et le type de données spécifié. Si la conversion échoue, une erreur non fatale est consignée et l'événement n'est pas généré.

Si vous souhaitez envoyer les données d'alerte actuelles à un serveur ObjectServer cible et insérer automatiquement toutes les données disponibles dans les zones correspondantes, omettez les identificateurs et valeurs de colonne de l'instruction `genevent` comme dans l'exemple suivant. Vous pouvez trouver ce format utile si vous souhaitez envoyer un double des données d'alerte actuelles à plusieurs serveurs ObjectServer. Avec ce format, notez que les données d'alerte incluent uniquement les zones qui ont été définies au-dessus de l'instruction `genevent` dans le fichier de règles.

```
genevent(cible)
```

Incluez les identificateurs et valeurs de colonne dans l'instruction `genevent(cible)` si vous souhaitez renseigner un sous-ensemble de zones spécifique dans le serveur ObjectServer cible. Par exemple :

```
genevent(StatusAlerts, @Node, $Node, @Summary, "Condition X has occurred")
```

Pour obtenir des exemples de fonction `genevent`, voir l'exemple de fichier de règles secondaire fourni pour prendre en charge la détection des débordements d'événements et des taux d'événements anormaux. Ce fichier, appelé `flood.rules`, est disponible dans le répertoire `$NCHOME/omnibus/extensions/eventflood`. (Le fichier `flood.rules` doit être utilisé avec le fichier de règles de configuration qui l'accompagne, appelé `flood.config.rules`.)

Envoi d'informations détaillées et de statuts de service aux cibles

Vous pouvez utiliser les instructions `genevent` pour envoyer des informations détaillées et des statuts de service dans les cas suivants :

- Si une instruction `registertarget` spécifie une table de détails à laquelle les informations détaillées doivent être envoyées, une instruction `genevent` qui envoie des alertes à la même cible envoie aussi les informations détaillées à la table de détails spécifiée dans l'instruction `registertarget`. Cette condition est uniquement valable si l'instruction `details` précède l'instruction `genevent`.
- Si vous utilisez la fonction de service pour définir le statut d'un service et que l'instruction de service précède l'instruction `genevent`, l'instruction `genevent` enverra les informations de statut au serveur ObjectServer cible.
- Si plusieurs instructions `details` ou de service précèdent ou suivent une instruction `genevent`, seules les informations de la dernière instruction `details` ou de service directement au-dessus de l'instruction `genevent` sont envoyées à la cible. Les informations générées par les autres instructions `details` ou de service sont uniquement associées à l'alerte principale et ne sont envoyées qu'aux cibles concernées définies dans les instructions `registertarget`.

Dans l'exemple suivant, l'instruction `genevent` ajoute les éléments `$c` et `$d` à la table `alerts.details` du serveur ObjectServer TEST2. Pour l'hôte surveillé, un statut de service marginal est également affecté à chaque alerte, lors de l'affichage depuis la fenêtre Services, disponible depuis le Conductor ou la liste d'événements.

```
DefaultAlerts = registertarget( "TEST1", "", "alerts.status" )
HighAlerts = registertarget( "TEST2", "", "alerts.status" "alerts.details" )
...
details ($a,$b)
...
details ($c,$d)
```

```
...
service($host, bad)
...
service($host, marginal)
...
genevent(HighAlerts)
...
details ($y,$z)
...
service($host, good)
```

Concepts associés:

«Détection de débordements d'événements et de taux d'événements anormaux», à la page 77

Les débordements d'événements peuvent provoquer une indisponibilité du serveur ObjectServer et entraîner de longues périodes pendant lesquelles il n'y a aucune visibilité des événements réseau. Un taux de réception d'événements anormalement faible ou élevé peut également indiquer un incident ou une modification de la source, qui doit être traité.

Référence associée:

«Enregistrement de serveurs ObjectServer cible et définition de cibles pour les alertes», à la page 50

Pour enregistrer un serveur ObjectServer et une table des alertes sur le serveur ObjectServer auquel vous souhaitez envoyer des événements (désigné par *serveur ObjectServer cible*), utilisez la fonction `registertarget`. Si nécessaire, vous pouvez utiliser cette fonction pour spécifier plusieurs serveurs ObjectServer cible et les tables d'alertes correspondantes. Utilisez la fonction `setdefaulttarget` pour spécifier un autre serveur ObjectServer cible et une autre table par défaut. Pour spécifier un serveur ObjectServer cible alternatif autre que le serveur par défaut, utilisez la fonction `settarget`.

«Fonction détails», à la page 48

Les détails sont des éléments supplémentaires créés par une sonde pour afficher des informations d'alerte qui ne sont pas stockées dans une zone de la table `alerts.status`. `Alerts` ne contient pas d'informations détaillées, sauf si ces informations sont ajoutées.

«Fonction service», à la page 58

La fonction `service` vous permet de définir le statut d'un service avant la transmission des alertes au serveur ObjectServer. Ce statut modifie la couleur d'une alerte lorsqu'elle est affichée dans les fenêtres de liste d'événements et `Service`.

Traitement multiprocessus des données d'alerte :

Lorsqu'un fichier de règles de sonde est traité, le traitement multiprocessus est utilisé par défaut pour appliquer les règles de sonde aux données d'événements brutes obtenues sur la source d'événements, et pour envoyer les alertes générées aux serveurs ObjectServer enregistrés. Notez que ce traitement multiprocessus est différent de la capture d'événements `mutliprocessus` ou `monoprocessus` implémentée sur certaines classes de sondes.

En mode multiprocessus, un seul processus est utilisé pour le traitement du fichier de règles, et des processus individuels sont utilisés pour communiquer avec chaque ObjectServer enregistré. L'unité d'exécution de traitement de fichier de règles applique les règles aux données entrantes, établit des connexions avec les serveurs ObjectServer concernés et envoie les résultats traités à l'unité d'exécution de communication appropriée. Cette unité d'exécution transforme les données traitées en instructions SQL `INSERT` et les envoie au serveur ObjectServer.

Le cas échéant, vous pouvez passer du traitement multiprocessus au traitement à une seule unité d'exécution en définissant la propriété **SingleThreadedComms** sur TRUE. En mode à une seule unité d'exécution, une seule unité d'exécution est utilisée pour le traitement du fichier de règles et pour la communication.

Avec le traitement multiprocessus, les alertes sont envoyées simultanément aux différents serveurs ObjectServer. Si nécessaire, vous pouvez utiliser le mode à un seul processus pour appliquer l'ordre d'envoi des alertes aux serveurs ObjectServer, défini par l'ordre dans lequel les instructions `registerTarget` sont répertoriées dans le fichier de règles. Vous pouvez également utiliser le mode monoprocessus pour le débogage ; cela permet de mieux comprendre l'ordre de traitement et d'envoi des événements.

En mode multiprocessus, si la mise en mémoire tampon est activée à l'aide de la propriété **Buffering**, une mémoire tampon de texte distincte est gérée pour chaque ObjectServer afin de conserver temporairement des données qui ne peuvent pas être traitées immédiatement par l'unité d'exécution de communication. Si la mise en mémoire tampon est désactivée, les instructions SQL INSERT sont envoyées aux serveurs ObjectServer dès leur création.

Si le mode stocker-et-transmettre est activé à l'aide de la propriété **StoreAndForward** et que le traitement multiprocessus est exécuté, des fichiers stocker-et-transmettre distincts sont créés pour conserver les données qui ne peuvent pas être envoyées à chaque ObjectServer. Les fichiers stocker-et-transmettre sont stockés dans le répertoire `$OMNIHOME/var` et sont nommés avec le format par défaut **SAFFileName.nom_serveur**, où **SAFFileName** représente le paramètre de la propriété **SAFFileName** et `.servername` est ajouté pour indiquer le nom du serveur ObjectServer.

Remarque : Lors de l'exécution du mode multiprocessus, une sonde démarre en mode monoprocessus (en raison de sa conception) avant de passer en mode multiprocessus. Ce comportement est également observé lorsqu'une sonde lit à nouveau son fichier de règles, et est enregistré dans le fichier journal de sonde en mode débogage.

Concepts associés:

«Mode stocker-et-transmettre pour les sondes», à la page 12

L'exécution des sondes peut continuer si le serveur ObjectServer cible est arrêté. Pendant cette période, la sonde passe en mode *stocker*. Elle repasse en mode *transmettre* lorsque le serveur ObjectServer fonctionne à nouveau.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Fonction Rechercher/Remplacer

Utilisez la fonction `regrepl` pour effectuer des opérations de recherche et de remplacement sur des chaînes à l'aide d'expressions régulières.

La syntaxe est la suivante :

```
regrepl(entrée, "expressionrégulière", "substitution" [,nombre])
```

Où :

- L'*entrée* est une expression de chaîne. La fonction `regrepl` lit la chaîne en entrée de gauche à droite.
- L'*expressionrégulière* est une chaîne. Cela ne peut pas être une expression de chaîne. Vous pouvez utiliser des parenthèses () dans la chaîne pour indiquer quelles sous-chaînes nécessitent une correspondance spécifique. Vous pouvez utiliser plusieurs ensembles de parenthèses dans une seule chaîne.
- La *substitution* est une expression de chaîne qui indique comment les chaînes correspondant à *entrée* doivent être écrites dans le résultat. Pour faire référence aux sous-chaînes correspondantes (entre parenthèses), vous pouvez utiliser des métacaractères ou une chaîne complète qui y correspond (ou plusieurs chaînes). Par exemple, `\1` correspond au premier groupe de l'expression régulière, `\2` au deuxième, etc., alors que `&` ou `\0` correspond à la chaîne complète. Les caractères et les chaînes ne correspondant pas à l'expression régulière sont copiés dans la chaîne de résultat.
- Le *nombre* est l'expression facultative d'un nombre entier positif ; il représente le nombre de substitutions à effectuer dans les chaînes correspondantes. Si vous n'indiquez pas de valeur pour *nombre*, les substitutions se poursuivent jusqu'à ce qu'il n'y ait plus aucune chaîne correspondante. Si *nombre* représente un nombre non entier, il est interprété comme un 0 et l'*entrée* reste inchangée. Si *nombre* est un nombre entier négatif, un message d'avertissement est ajouté au journal de la sonde et l'*entrée* reste inchangée.

Exemple : Utilisation de la fonction Rechercher/Remplacer pour supprimer les caractères indésirables d'une chaîne

L'exemple suivant montre comment utiliser la fonction `regrepl` pour remplacer les traits de soulignement (`_`), les symboles de pourcentage (`%`) et les guillemets simples (`'`) par une chaîne vide :

```
$result = regrepl("%Node__='foobar27'%" , "([\_%']*)", "" )
```

Le résultat de cette expression est le suivant :

```
$result="Node=foobar27"
```

L'exemple suivant montre comment utiliser la fonction `regrepl` pour remplacer les caractères de contrôle CR (retour chariot) ou LF (saut de ligne) par une chaîne vide :

```
@Summary = regrepl($Summary, '[\n\r]', "")
```

Exemple : Réorganisation des groupes de caractères dans une chaîne

L'exemple suivant indique comment faire correspondre plusieurs sous-chaînes à l'intérieur une même chaîne et, à la sortie, comment réorganiser les sous-chaînes. L'ordre des sous-chaînes dans une chaîne en entrée diffère de celui de la chaîne en sortie.

```
regrepl("aba argle aca", "(a.a) (.*) (a.a)", "\3 \2 \1")
```

L'expression régulière correspond aux sous-chaînes dans l'ordre suivant :

```
\1="aba"  
\2="argle"  
\3="aca"
```

Le remplacement indique que les chaînes déjà mises en correspondance doivent être écrites dans le sens inverse de celui dans lequel l'entrée est lue. Par conséquent, le résultat de cette expression est le suivant :

```
$result = "aca argle aba"
```

Exemple : Utilisation de métacaractères pour mettre une chaîne complète en correspondance

L'exemple suivant indique comment utiliser les métacaractères &, aussi exprimés sous forme de \0, pour mettre en correspondance la chaîne complète représentée par l'expression régulière :

```
regreplace("aaabbbaaa", "a(b+)a" "_&_")
```

Les métacaractères & ou \0 mettent en correspondance tout ce qui est mappé à l'expression régulière, et pas uniquement la sous-chaîne entre parenthèses. Dans cet exemple, l'expression régulière correspond à la sous-chaîne suivante dans l'entrée : abba. Les sous-chaînes non mises en correspondance sont copiées vers la sortie.

Le résultat de cette expression est le suivant :

```
$result="aa_abba_aa"
```

Concepts associés:

Annexe C, «Expressions régulières», à la page 345

Tivoli Netcool/OMNIBus prend en charge l'utilisation des expressions régulières dans les demandes de recherche que vous réalisez sur le serveur ObjectServer. Les expressions régulières sont des séquences d'*atomes* composées de caractères normaux et de métacaractères.

Fonction service

La fonction `service` vous permet de définir le statut d'un service avant la transmission des alertes au serveur ObjectServer. Ce statut modifie la couleur d'une alerte lorsqu'elle est affichée dans les fenêtres de liste d'événements et Service.

La syntaxe est :

```
service(identificateur_service, statut_service)
```

La variable `identificateur_service` identifie le service surveillé, par exemple `$host`.

Le tableau suivant répertorie les niveaux de statut de service.

Tableau 18. Niveaux de statut de la fonction `service`

Niveau de statut de service	Définition
BAD	Le contrat de service n'est pas respecté.
MARGINAL	Des problèmes existent avec le service.
GOOD	Il n'y a aucun incident avec le service.
<i>No Level Defined</i>	Le statut du service est inconnu.

Exemple : fonction service

Si vous souhaitez qu'une sonde Ping renvoie un statut de service pour chaque hôte surveillé, vous pouvez utiliser la fonction `service` du fichier de règles pour affecter un statut de service à chaque alerte. Dans l'exemple suivant, un statut de service est affecté à chaque alerte en fonction de la valeur de l'élément `status`.

```
switch ($status)
{
  case "unreachable":
    @Severity = "5"
    @Summary = @Node + " is not reachable"
    @Type = 1
    service($host, bad) # Entrée de service
  case "alive":
    @Severity = "3"
    @Summary = @Node + " is now alive"
    @Type = 2
    service($host, good) # Entrée de service
  case "noaddress":
    @Severity = "2"
    @Summary = @Node + " has no address"
    service($host, marginal) # Entrée de service
  case "removed":
    @Severity = "5"
    @Summary = @Node + " has been removed"
    service($host, marginal) # Entrée de service
  case "slow":
    @Severity = "2"
    @Summary = @Node + " has not responded within
    trip time"
    service($host, marginal) # Entrée de service
  case "newhost":
    @Severity = "1"
    @Summary = @Node + " is a new host"
    service($host, good) # Entrée de service
  case "responded":
    @Severity = "0"
    @Summary = @Node + " has responded"
    service($host, good) # Entrée de service
  default:
    @Summary = "Ping Probe error details: " + $*
    @Severity = "3"
    service($host, marginal) # Entrée de service
}
```

Surveillance des charges de la sonde

Pour surveiller la charge, il est nécessaire d'obtenir des mesures temporelles et de calculer le nombre d'événements traités pendant une période. La fonction `updateload` effectue une mesure temporelle chaque fois qu'elle est appelée et la fonction `getload` renvoie la charge en tant qu'événements par seconde.

A chaque exécution de la fonction `updateload`, l'horodatage actuel, enregistré en secondes et en microsecondes, est ajouté au début d'une série d'horodatages. Les horodatages restants enregistrent la différence d'heure avec l'horodatage précédent. Par exemple, pour prendre une mesure temporelle et mettre à jour une propriété appelée **load** avec un nouvel horodatage :

```
%load = updateload(%load)
```

Conseil : En fonction du système d'exploitation, différents niveaux de granularité peuvent être rapportés dans les horodatages.

Vous pouvez spécifier une fenêtre temporelle maximale pendant laquelle des échantillons sont conservés ainsi qu'un nombre maximum d'échantillons. Par défaut, la fenêtre temporelle est d'une seconde et le nombre d'échantillons maximum est de 50. Vous pouvez spécifier le nombre de secondes pendant lesquelles des échantillons de chargement sont conservés ainsi que le nombre maximum d'échantillons au format :

```
fenêtre_temporelle_en_secondes.nombre_échantillons_max
```

Par exemple, pour définir ou réinitialiser ces valeurs pour la propriété load :

```
%load = "2.40"
```

Lorsque le nombre de secondes d'une fenêtre temporelle est dépassé, tous les échantillons en-dehors de cette fenêtre sont supprimés. Lorsque le nombre d'échantillons atteint la limite, la mesure la plus ancienne est supprimée.

La fonction getload calcule la charge actuelle, renvoyée en tant qu'événements par seconde. Par exemple, pour calculer la charge actuelle et l'affecter à l'élément temporaire current_load :

```
$current_load = getload(%load)
```

La fonction geteventcount complète la fonction getload en renvoyant le nombre total d'événements de la fenêtre d'événements.

Référence associée:

«Exemples de fichier de règles», à la page 62

Les exemples suivants illustrent des segments types du fichier de règles.

Mots réservés dans le langage de règles de sonde

Dans le langage de règles de sonde, certains mots sont réservés en tant que mots clés et ne doivent pas être utilisés en tant que noms de variable ou de propriété dans les fichiers de règles de sonde.

La liste ci-dessous indique les mots réservés :

- and
- array
- bad
- break
- case
- char
- character
- charcount
- clear
- datetime
- datetotime
- debug
- decode
- default
- details
- discard
- discarded
- double

- else
- error
- exists
- exit
- expand
- extract
- false
- fatal
- foreach
- genevent
- getdate
- good
- if
- in
- include
- info
- information
- int
- entier
- len
- length
- log
- lookup
- lower
- ltrim
- marginal
- match
- nmatch
- no
- not
- nvp_add
- nvp_remove
- off
- on
- or
- printable
- real
- recover
- registertarget
- regmatch
- regreplace
- remove
- rtrim
- scanformat
- service

- setdefaultobjectserver
- setdefaulttarget
- setlog
- setobjectserver
- settarget
- split
- string
- substr
- switch
- table
- timetodate
- true
- update
- upper
- warn
- warning
- xor
- yes

Exemples de fichier de règles

Les exemples suivants illustrent des segments types du fichier de règles.

«Exemple : amélioration de la zone Summary»

«Exemple : renseignement de plusieurs zones»

«Exemple : instructions IF imbriquées», à la page 63

«Exemple : correspondance d'expressions régulières», à la page 63

«Exemple : extraction d'une expression régulière», à la page 63

«Exemple : comparaisons numériques», à la page 63

«Exemple : expressions numériques simples», à la page 63

«Exemple : chaînes et valeurs numériques dans une expression», à la page 64

«Exemple : utilisation des fonctions de chargement pour surveiller des noeuds», à la page 64

Exemple : amélioration de la zone Summary

Cet exemple de règle teste si l'élément \$trap-type est Link-Up. Si tel est les cas, la zone @Summary est renseignée avec une chaîne constituée de Link up on, du nom du noeud de l'enregistrement qui est généré, de Port et de la valeur de l'élément \$ifIndex :

```
if( match($trap-type,"Link-Up") )
{
  @Summary = "Link up on " + @Node + " Port " + $ifIndex
}
```

Exemple : renseignement de plusieurs zones

Cet exemple de règle est similaire à la règle précédente, mais les zones @AlertKey et @Severity sont également renseignées :

```

if( match($trap-type, "Link-Up") )
{
  @Summary = "Link up on " + @Node + " Port " + $ifIndex
  @AlertKey = $ifIndex
  @Severity = 4
}

```

Exemple : instructions IF imbriquées

Cet exemple de règle teste d'abord si l'interruption provient d'un gestionnaire Acme puis s'il s'agit d'un Link-Up. Si les deux conditions sont remplies, la zone @Summary est renseignée avec les valeurs de la zone @Node et des éléments \$ifIndex et \$ifLocReason :

```

if( match($enterprise,"Acme") )
{
  if( match($trap-type, "Link-Up") )
  {
    @Summary= "Acme Link Up on " + @Node + " Port " + $ifIndex +
    " Reason: "+$ifLocReason
  } }

```

Exemple : correspondance d'expressions régulières

Cet exemple de règle recherche la présence d'une ligne commençant par Acme Configuration: suivie d'un seul numéro :

```

if (regmatch($enterprise,"^Acme Configuration:[0-9]"))
{
  @Summary="Generic configuration change for " + @Node
}

```

Exemple : extraction d'une expression régulière

Cet exemple de règle recherche la présence d'une ligne commençant par Acme Configuration: suivie d'un seul numéro. Si la condition est remplie, il extrait ce numéro et le place dans la zone @Summary :

```

if (regmatch($enterprise,"^Acme Configuration:[0-9]"))
{
  @Summary="Acme error "+extract($enterprise,"^Acme Configuration:
  ([0-9])")+" on" + @Node
}

```

Exemple : comparaisons numériques

Cet exemple de règle teste la valeur d'un élément appelé \$freespace en tant que valeur numérique en la convertissant en entier et en effectuant une comparaison numérique :

```

if (int($freespace) < 1024)
{
  @Summary="Less than 1024K free on drive array"
}

```

Exemple : expressions numériques simples

Cet exemple de règle crée un élément appelé \$tmpval. La valeur de \$tmpval est dérivée de l'élément \$temperature, converti en entier auquel 20 est soustrait. L'élément de type chaîne \$tmpval contient le résultat de ce calcul :

```

$tmpval=int($temperature)-20

```

Exemple : chaînes et valeurs numériques dans une expression

Cet exemple de règle crée un élément appelé \$Kilobytes. La valeur de \$Kilobytes est dérivée de l'élément \$DiskSize, divisé par 1024 avant d'être converti en type de chaîne avec ajout de la lettre K :

```
$Kilobytes = string(int($DiskSize)/1024) + "K"
```

Exemple : utilisation des fonctions de chargement pour surveiller des noeuds

Cet exemple illustre comment mesurer le chargement de chaque noeud qui génère des événements. Si un noeud crée plus de cinq événements par seconde, un avertissement est écrit dans le fichier journal de la sonde. Si plus de 80 événements sont générés par seconde pour tous les noeuds surveillés par la sonde, ils sont envoyés à un serveur ObjectServer alternatif et un avertissement est écrit dans le fichier journal de la sonde.

```
# declare the ObjectServers HIGHLOAD and LOWLOAD
# declare the loads array
LOWLOAD = registertarget( "NCOMS_LOW", "", "alerts.status")
HIGHLOAD = registertarget( "NCOMS_HIGH", "", "alerts.status")
array loads;

# initialize array items with the number of seconds samples may span and
# number of samples to maintain.

if ( match("", loads[@Node] ) ){
    loads[@Node] = "2.50"
}
if ( match("", %general_load) ){
    %general_load="2.50"
}
loads[@Node] = updateload(loads[@Node])
%general_load=updateload(%general_load)
if ( int(getload(loads[@Node]) ) > 5 ){
    log(WARN, $Node + " is creating more than 5 events per second")
}
if ( int(getload(%general_load)) > 80){
    log(WARN, "Probe is creating more than 80 events per second - switching to HIGHLOAD")
    settarget(HIGHLOAD)
}
}
```

Référence associée:

«Fonction Rechercher/Remplacer», à la page 57

Utilisez la fonction `regreplace` pour effectuer des opérations de recherche et de remplacement sur des chaînes à l'aide d'expressions régulières.

«Exemples de fonctions permettant d'exécuter des boucles», à la page 27

Ces exemples sur l'instruction en boucle `FOREACH` vous aideront à déployer la fonction dans votre environnement Tivoli Netcool/OMNIBus.

Test de fichiers de règles

Vous pouvez tester la syntaxe d'un fichier de règles à l'aide du vérificateur de syntaxe de règles de sonde, `nco_p_syntax`. Cela se révèle plus efficace que l'exécution de la sonde pour vérifier que la syntaxe du fichier de règles est correcte.

Pourquoi et quand exécuter cette tâche

Le vérificateur de syntaxe de règles de sonde est installé dans la fonction Support de sonde de Tivoli Netcool/OMNIBus, dans le répertoire suivant :

- `UNIX` `Linux` `$NCHOME/omnibus/probes`

- `Windows %NCHOME%\omnibus\probes\win32`

Procédure

Pour exécuter le vérificateur de syntaxe de règles de sonde, entrez la commande suivante :

```
nco_p_syntax -rulesfile /rules_file_path/fichier_règles.rules
```

Lors de l'exécution de cette commande, utilisez l'option de ligne de commande `-rulesfile` pour indiquer le chemin d'accès complet et le nom du fichier de règles.

Résultats

Le vérificateur de syntaxe de règles de sonde s'exécute en mode débogage par défaut. Vous pouvez modifier ce paramètre à l'aide de l'option de ligne de commande `-messagelevel`; par exemple, `-messagelevel info`.

La sonde se connecte au serveur ObjectServer, teste le fichier de règles, affiche les erreurs à l'écran, puis se ferme. Si aucune erreur n'est affichée, cela signifie que la syntaxe du fichier de règles est correcte. Pour plus de détails sur le vérificateur de syntaxe de règles de sonde, voir la publication concernant cette sonde. Vous pouvez accéder à cette publication de la manière suivante, à partir du centre de documentation IBM Tivoli Network Management (<http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp>) :

1. Développez le noeud *IBM Tivoli Netcool/OMNIBus* dans le panneau de navigation située sur la gauche.
2. Développez le noeud *Tivoli Netcool/OMNIBus probes and TSMs*.
3. Accédez au noeud *Universal*.

Débogage de fichiers de règles

Lorsque vous modifiez le fichier de règles, ajoutez de nouvelles règles ou encore créez des tables de consultation, il se révèle particulièrement utile de tester la sonde en l'exécutant en mode débogage. Le mode débogage montre la manière dont un événement fait l'objet d'une analyse syntaxique effectuée par la sonde et peut détecter les problèmes éventuels relatifs au fichier de règles.

Pourquoi et quand exécuter cette tâche

Vous pouvez activer le mode débogage à partir de l'interface de ligne de commande ou en modifiant le fichier de propriétés de sonde. Si vous avez besoin de modifier le niveau de message d'une sonde en cours d'exécution sans arrêter la sonde, vous pouvez utiliser la commande `kill` sur l'identificateur de processus de sonde (PID).

Procédure

- Pour activer le mode débogage à partir de l'interface de ligne de commande, entrez la commande suivante :

```
$OMNIHOME/probes/nco_p_nom_sonde -messagelevel DEBUG -messagelog STDOUT
```

Si vous omettez l'option de ligne de commande `-messagelog`, les informations de débogage sont envoyées au fichier journal de la sonde situé dans le répertoire `$OMNIHOME/log`, au lieu d'être affichées.

- Pour activer le mode débogage en utilisant le fichier de propriétés de sonde, ajoutez les entrées suivantes au fichier :

```
MessageLevel : "DEBUG"  
MessageLog : "STDOUT"
```

Si vous omettez la propriété **MessageLog**, les informations de débogage sont envoyées au fichier journal de la sonde dans le répertoire \$OMNIHOME/log, au lieu d'être affichées.

- Pour modifier le niveau de message d'une sonde en cours d'exécution afin qu'elle s'exécute en mode débogage, utilisez la commande `kill -USR2 pid` sur l'identificateur de processus de sonde (PID).

Lors de chaque exécution de la commande `kill -USR2 pid`, le niveau de message est cyclé.

Pour plus d'informations, voir les pages d'aide concernant l'utilisation des commandes **ps** et **kill**.

Conseil : Pour les sondes JAVA, exécutez la commande **kill** sur l'identificateur de processus **nco_p_nonnative**.

Que faire ensuite

Pour permettre la prise en compte des modifications apportées au fichier de règles, forcez la sonde à relire le fichier de règles.

Tâches associées:

«Relecture du fichier de règles»

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Relecture du fichier de règles

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Procédure

Vous pouvez forcer une sonde à relire le fichier de règles de l'une des manières suivantes :

- Exécutez la commande suivante sur l'identificateur de processus de sonde (PID) :
`kill -HUP pid`

où *pid* est l'identificateur de processus.

Si le fichier de règles mis à jour contient des erreurs de syntaxe ou des références à des zones qui n'existent pas, la sonde consigne un message d'erreur sur réception du signal HUP et continue d'utiliser la version antérieure du fichier de règles.

Pour plus d'informations, voir les pages d'aide concernant l'utilisation des commandes **ps** et **kill**.

Conseil : Pour les sondes JAVA, exécutez la commande `kill -HUP` sur le processus `nco_p_nonnative`.

- Émettez une commande HTTP `nco_probereloadrules` sur la sonde pour recharger le fichier de règles.
- Redémarrez la sonde.

Important : Si vous redémarrez la sonde, les événements reçus pendant que la sonde était arrêtée peuvent être perdus.

Concepts associés:

«Fichier de règles», à la page 8

Le fichier de règles définit comment la sonde traite les données d'événement pour créer une alerte significative. Pour chaque alerte, le fichier de règles crée également un identificateur qui identifie de manière unique la source du problème.

Tâches associées:

«Débogage de fichiers de règles», à la page 65

Lorsque vous modifiez le fichier de règles, ajoutez de nouvelles règles ou encore créez des tables de consultation, il se révèle particulièrement utile de tester la sonde en l'exécutant en mode débogage. Le mode débogage montre la manière dont un événement fait l'objet d'une analyse syntaxique effectuée par la sonde et peut détecter les problèmes éventuels relatifs au fichier de règles.

«Définition de tables de recherche dans le fichier de règles», à la page 46

Vous pouvez créer une table de consultation directement dans le fichier de règles.

«Rechargement de fichiers de règles (`nco_probereloadrules`)», à la page 114

Vous pouvez utiliser l'utilitaire `nco_probereloadrules` pour recharger à distance un fichier de règles de sonde sans avoir à redémarrer la sonde.

«Activation de la mise en cache des fichiers de règles de sonde», à la page 69

Pour vous assurer qu'une sonde est toujours en mesure de lire un ensemble valide de règles lorsqu'elle est démarrée, activez la mise en cache du fichier de règles. Par défaut, la mise en cache du fichier de règles est désactivée.

«Configurer une sonde pour le téléchargement de fichiers de règles éloignés», à la page 68

Vous pouvez configurer la propriété `RulesFile` d'un fichier de règles de sonde pour permettre le téléchargement via HTTP de fichiers de règles à partir d'un référentiel central ou d'un système de fichiers.

Référence associée:

«Recharger le fichier de règles», à la page 119

Cette demande accède à l'attribut `reloadrulesflag` de la sonde, contenu dans la bibliothèque C de sondes standard (`libOpl`). En définissant la valeur de cet attribut, la demande force la sonde à recharger son fichier de règles avant le traitement de l'événement suivant. Un signal `SIGHUP` sur UNIX déclenche la même action.

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Configurer une sonde pour le téléchargement de fichiers de règles éloignés

Vous pouvez configurer la propriété **RulesFile** d'un fichier de règles de sonde pour permettre le téléchargement via HTTP de fichiers de règles à partir d'un référentiel central ou d'un système de fichiers.

Avant de commencer

Remarque : Pour des raisons de sécurité, il est conseillé de n'exécuter les sondes qui utilisent l'interface HTTP que dans un environnement de réseau privé sécurisé.

Procédure

1. Dans le fichier de propriétés de la sonde, définissez la propriété **RulesFile** en lui affectant comme valeur l'URL du fichier de règles. Par exemple :
`http://testhost.ibm.com:8080/ibm/console/omnibus/repository/probes/rules/simnet.rules`
Pour spécifier un fichier de règles stocké sur un serveur Web IPv6, utilisez des crochets ([]) pour délimiter l'adresse du serveur dans l'URL. Par exemple :
`http://[fed0::7887:234:5edf:fe65:348]:8080/ibm/console/omnibus/repository/probes/rules/simnet.rules`
2. Redémarrez la sonde ou forcez-la à relire le fichier de règles.

Résultats

Le fichier de règles est chargé à partir du système de fichiers distant. Si la sonde ne réussit pas à lire le fichier de règles, un message similaire au message suivant est écrit dans le fichier journal de la sonde :

```
Error: E-UNK-000-000: Failed to open Rules file:  
http://testhost:9080/ibm/console/omnibus/repository/probe/rules/simnet.rules
```

Si la mise en cache des fichiers de règles est activée, la sonde revient à la copie du fichier de règles qui se trouve dans le cache. Sinon, la sonde s'arrête.

Vous pouvez utiliser l'utilitaire **nco_http** fourni avec Tivoli Netcool/OMNIBus pour déterminer l'état HTTP du serveur du référentiel ou du système de fichiers. Utilisez la commande suivante pour obtenir le code d'état HTTP qui indique la cause de l'échec :

```
$OMNIHOME/bin/nco_http -uri chemin
```

Où *chemin* représente l'URL ou le chemin de répertoire que la sonde tente d'utiliser pour télécharger le fichier Include.

Tâches associées:

«Relecture du fichier de règles», à la page 66

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Référence associée:

«Intégration de plusieurs fichiers de règles dans un fichier de règles», à la page 70
Vous pouvez utiliser des instructions `include` pour appeler un ou plusieurs fichiers de règles secondaires à partir d'un fichier de règles principal.

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Activation de la mise en cache des fichiers de règles de sonde

Pour vous assurer qu'une sonde est toujours en mesure de lire un ensemble valide de règles lorsqu'elle est démarrée, activez la mise en cache du fichier de règles. Par défaut, la mise en cache du fichier de règles est désactivée.

Lorsque vous activez la mise en cache des fichiers de règles et que vous démarrez la sonde, celle-ci écrit une copie du fichier de règles dans un fichier de cache unique. Si la taille du fichier dépasse 1 Go, le fichier de règles est écrit dans plusieurs fichiers de cache. Chaque fois que la sonde lit avec succès le fichier de règles, elle l'écrit en mémoire cache. Lorsque la sonde est ensuite redémarrée, elle lit le fichier de cache si elle ne peut pas lire le fichier de règles.

Les fichiers d'inclusion et les tables de consultation référencés dans le fichier de règles sont écrits en ligne dans le fichier de cache. Si le fichier de règles contient des instructions `registertarget`, utilisez les zones `%Server` et `%ServerBackup` pour enregistrer les serveurs ObjectServer cibles, les valeurs de propriété sont résolues en tant que chaînes entre guillemets dans le fichier de cache. Si l'analyseur syntaxique des règles de sonde identifie, dans une table de consultation, des lignes ayant des clés en double, un avertissement est généré. Le fichier de cache ne conserve pas l'ordre des entrées dans le fichier de règles. Par conséquent, si vous recevez un avertissement, vérifiez que l'ordre des entrées dans le fichier de cache ne modifie pas le comportement de la sonde.

Remarque : Lorsqu'un fichier de règles est mis en cache, un total de contrôle identifiant la version du fichier est préservée dans des métadonnées de commentaire dans le fichier mis en cache. Ne modifiez pas cette valeur dans le fichier mis en cache. Voici un exemple de métadonnées de commentaire contenant un total de contrôle :

```
#METADATA CheckSum=f967b187bf66e08d4277ab688f87f696bea3bb22
```

Procédure

Pour activer la mise en cache :

1. Dans le fichier de propriétés de sonde, définissez la propriété **CacheRules** sur 1.
2. Si vous souhaitez utiliser un fichier de mise en cache différent de la valeur par défaut, définissez la propriété **CacheRulesFile** pour indiquer son chemin d'accès et son nom. Placez le chemin entre guillemets (" ").

Résultats

La sonde relit le fichier de règles et l'écrit dans le fichier mis en cache. Chaque fois que la sonde relit le fichier de règles, elle vérifie la valeur de la propriété **CacheRules** et de la propriété **CacheRulesFile**. Si les valeurs de ces propriétés sont définies en conséquence, le fichier de cache est mis à jour.

Si la taille du fichier de cache dépasse 1 Go, un nouveau fichier de cache est créé, et le suffixe 1 est ajouté au nom de fichier.

Exemple

Si la valeur de la propriété **CacheRulesFile** est "\$OMNIHOME/var/rulescaché", le fichier de règles est écrit dans ce fichier de cache jusqu'à ce qu'il atteigne 1 Go. Ensuite, un nouveau fichier de cache est créé, appelée \$OMNIHOME/var/rulescaché1. Si ce fichier dépasse 1 Go, \$OMNIHOME/var/rulescaché2 est créé, et ainsi de suite..

Concepts associés:

«Fichier de règles», à la page 8

Le fichier de règles définit comment la sonde traite les données d'événement pour créer une alerte significative. Pour chaque alerte, le fichier de règles crée également un identificateur qui identifie de manière unique la source du problème.

Tâches associées:

«Relecture du fichier de règles», à la page 66

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Référence associée:

«Enregistrement de serveurs ObjectServer cible et définition de cibles pour les alertes», à la page 50

Pour enregistrer un serveur ObjectServer et une table des alertes sur le serveur ObjectServer auquel vous souhaitez envoyer des événements (désigné par *serveur ObjectServer cible*), utilisez la fonction `registertarget`. Si nécessaire, vous pouvez utiliser cette fonction pour spécifier plusieurs serveurs ObjectServer cible et les tables d'alertes correspondantes. Utilisez la fonction `setdefaulttarget` pour spécifier un autre serveur ObjectServer cible et une autre table par défaut. Pour spécifier un serveur ObjectServer cible alternatif autre que le serveur par défaut, utilisez la fonction `settarget`.

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Intégration de plusieurs fichiers de règles dans un fichier de règles

Vous pouvez utiliser des instructions `include` pour appeler un ou plusieurs fichiers de règles secondaires à partir d'un fichier de règles principal.

Le format d'une instruction `include` est le suivant : `include "fichiers-de-règles"`. Pour spécifier un fichier de règles secondaire, vous pouvez utiliser une URL de base ou un chemin de répertoire relatif ou absolu.

La meilleure manière d'appeler des fichiers de règles secondaires consiste à utiliser une variable d'environnement pour indiquer l'URL de base. Dans les exemples ci-dessous, c'est la variable d'environnement `NC_RULES_HOME` qui est utilisée.

Remarque : Les caractères de traversée de répertoires (`../` ou `..\`) ne sont pas pris en charge dans les URL de base ou les inclusions relatives. La commande suivante, par exemple, n'est pas prise en charge :

```
include "subdir1/subdir2/./include.rules"
```

Utilisation d'URL de base

Pour utiliser une URL de base pour spécifier un fichier Include, procédez comme suit :

1. Créez une variable d'environnement `NC_RULES_HOME` sur l'ordinateur hôte de la sonde et affectez-lui comme valeur l'URL de base du système de fichiers ou du référentiel de règles. Par exemple :

```
http://testhost.ibm.com:8080/ibm/console/omnibus/repository/probes/
rules/
```

2. Dans le fichier de règles principal de la sonde, ajoutez la commande suivante pour inclure un fichier de règles :

```
include "$NC_RULES_HOME/chemin-des-inclusions"
```

Où *chemin-des-inclusions* représente le chemin de répertoire menant à vos fichiers de règles secondaires.

Lorsque vous utilisez une variable d'environnement, la sonde appelle toujours le fichier Include correct lors de l'analyse des règles. En outre, vous avez ensuite la possibilité de changer l'hôte et le numéro de port sans modifier le fichier de règles.

3. Utilisez une variable d'environnement lorsque vous spécifiez la propriété **RulesFile** dans le fichier de propriétés de la sonde. Par exemple :

```
RulesFile : '$NC_RULES_HOME/myrulesfile.rules'
```

Utilisation d'inclusions relatives

Une inclusion relative doit être exprimée relativement au fichier de règles qui appelle le fichier Include. Par exemple, soit la structure de répertoire suivante :

```
répertoire-de-base/simnet.rules
répertoire-de-base/include-common/probewatch.include
```

Pour inclure `probewatch.include` dans `simnet.rules`, ajoutez l'instruction `include` suivante dans le fichier `simnet.rules` :

```
include "include-common/probewatch.include"
```

Identification et résolution des problèmes d'inclusion

Si une sonde ne démarre pas, regardez si le fichier journal de la sonde ne contient pas un message d'erreur similaire au message suivant :

```
Error: E-UNK-000-000: Failed to open Rules file: chemin
```

Où *chemin* représente l'URL ou le chemin de répertoire que la sonde tente d'utiliser pour télécharger le fichier Include.

Si aucun message ne contient la chaîne `Failed to open Rules file`, recherchez des messages contenant une ou plusieurs des chaînes suivantes :

- Failed to open include file
- Failed to open lookup file

Si une sonde démarre mais n'utilise pas le fichier Include attendu, redémarrez-la après avoir défini le niveau de messages adapté au débogage. Les entrées de journal similaires aux entrées suivantes indiquent à partir de quel emplacement la sonde charge le fichier Include :

```
Debug: D-UNK-000-000: Reading chemin
Debug: D-UNK-000-000: Plain text rules file detected.
Debug: D-UNK-000-000: Including path
```

Dans les exemples de message suivants, la sonde charge le fichier de règles via HTTP à partir d'un référentiel distant :

```
2013-09-23T16:33:57: Debug: D-UNK-000-000: Reading http://testserver:9080/ibm/
console/probe/rules/simnet.rules
2013-09-23T16:33:57: Debug: D-UNK-000-000: Plain text rules file detected
2013-09-23T16:33:57: Debug: D-UNK-000-000: Including http://testserver:9080/ibm/
console/probe/rules/extensions/roi/probewatch.include
```

Tâches associées:

«Envoi de demandes à distance à des sondes (nco_http)», à la page 109
Utilisez l'utilitaire **nco_http** pour vous connecter à l'interface HTTP ou HTTPS d'une sonde et envoyer des demandes.

«Configurer une sonde pour le téléchargement de fichiers de règles éloignés», à la page 68

Vous pouvez configurer la propriété **RulesFile** d'un fichier de règles de sonde pour permettre le téléchargement via HTTP de fichiers de règles à partir d'un référentiel central ou d'un système de fichiers.

Référence associée:

Chapitre 2, «Fichiers de règles des sondes», à la page 21

Le fichier de règles définit comment la sonde traite les données d'événement pour créer des alertes Tivoli Netcool/OMNIBus significatives. Ce fichier crée également un identificateur pour chaque alerte, afin d'identifier de manière unique la source du problème et de permettre le dédoublement des événements répétés.

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Probe Rules Utility

Vous pouvez utiliser l'outil de ligne de commande de Probe Rules Utility (**nco_rules_util**) pour gérer plus facilement les fichiers de règles de sonde qui sont stockés dans un référentiel central ou un système de fichiers.

Vous pouvez configurer Probe Rules Utility à l'aide d'un fichier de propriétés qui est lu lorsque l'outil est exécuté. Les options de ligne de commande indiquées lors de l'exécution se substituent aux valeurs du fichier de propriétés. Des exemples de fichier de propriétés (**nco_rules_util.props**) sont fournis dans les répertoires *NCHOME/omnibus/etc/* et *NCHOME/omnibus/etc/defaults*.

Remarque : Probe Rules Utility s'exécute directement sur le système de fichiers du référentiel et doit donc pouvoir y accéder.

Syntaxe

La syntaxe d'utilisation de l'outil est la suivante :

```
nco_rules_util [action] [option] [chemin]
```

Où `action` représente l'une des actions disponibles et `option` représente l'une des options pour une action ou l'une des options de ligne de commande. Le paramètre `chemin` représente le chemin d'accès d'un fichier de règles ou d'un répertoire. Seuls les fichiers possédant l'extension `.rules` sont répertoriés.

Les actions disponibles sont les suivantes :

- `list` : Affiche la liste des fichiers de règles contenus dans le répertoire indiqué.
 - `lstree` : Affiche la liste des fichiers `Lookup` et `Include` auxquels le fichier de règles indiqué fait référence. Si vous indiquez un répertoire, les informations concernant tous les fichiers de règles qu'il contient sont affichées.
- Cette action vous offre une meilleure visibilité sur les dépendances contenues dans un fichier de règles et est utile pour déboguer les problèmes.

Le tableau ci-dessous répertorie les options disponibles pour les actions `list` et `lstree`.

Tableau 19. Options des actions `Probe Rules Utility`.

Option	Description
<code>-checksum</code>	<p>Affiche le total de contrôle SHA1 de chaque fichier de règles répertorié.</p> <p>L'option <code>-checksum</code> génère pour un fichier de règles une valeur de total de contrôle qui peut être comparée à la valeur <code>RulesChecksum</code> de la table <code>registry.probes</code>. Si un fichier de règles contient des fichiers <code>Include</code> ou <code>Lookup</code>, le total de contrôle qui est généré est une agrégation des totaux de contrôle de tous les fichiers référencés.</p> <p>Le total de contrôle identifie de façon unique le fichier de règles ainsi que les éventuels fichiers <code>Include</code> et <code>Lookup</code> référencés. Un algorithme personnalisé est utilisé pour générer le total de contrôle d'après le "digest" SHA1 de chaque fichier.</p>
<code>-details</code>	Affiche la date et l'heure de dernière modification d'un fichier ainsi que sa taille en octets.
<code>-short</code>	<p>Affiche au format suivant les informations relatives à un fichier de règles :</p> <pre> <i>nom-fichier</i> Checksum : <i>total-de-contrôle</i> Last Modified : <i>heure date</i> Size : <i>n octets</i> </pre> <p>Exemple :</p> <pre> syslog.rules Checksum : b66759ddc35e4b6503791ba5603548bd38179502 Last Modified : 8:54:06 AM Oct 15, 2012 Size : 7,758 Bytes </pre> <p>L'option <code>-short</code> réduit la largeur de la sortie écran pour un meilleur affichage dans les petites fenêtres de terminal. Elle donne les meilleurs résultats si vous affichez les informations d'un petit nombre de fichiers. L'action <code>lstree</code> peut générer beaucoup de texte, qui peut être difficile à lire si vous utilisez l'option <code>-short</code>.</p> <p>Si vous n'indiquez pas l'option <code>-details</code> ou <code>-checksum</code>, l'option <code>-short</code> est ignorée.</p>

Le tableau suivant répertorie les propriétés et les options de ligne de commande de `Probe Rules Utility` :

Tableau 20. Propriétés et options de ligne de commande de Probe Rules Utility.

Propriété	Option de ligne de commande	Description
N/D	-dumpprops	Affiche la liste des propriétés configurées pour l'outil, y compris celles qui ont leur valeur par défaut.
N/D	-help	Affiche l'aide de l'utilitaire.
log.filename	-messagelog	Spécifie le nom du fichier journal. Si aucun fichier journal n'est spécifié, les messages sont dirigés vers la sortie d'erreur standard (stderr).
log.maxcount	N/D	<p>Spécifie le nombre maximum de fichiers journaux pouvant être écrits.</p> <p>Lorsque cette propriété est activée, le fichier journal est remplacé quand il atteint la taille spécifiée dans la propriété log.maxsize.</p> <p>Un suffixe est ajouté au fichier journal. Chaque nouveau fichier journal reçoit ce suffixe incrémenté lorsqu'il atteint sa taille maximum.</p> <p>Par exemple, si la valeur de log.maxcount est 2 et que le nom du fichier journal est test.log, les fichiers journaux suivants sont créés :</p> <ul style="list-style-type: none"> • test.log0 • test.log1 <p>La valeur de cette propriété est ignorée si la propriété log.filename n'est pas activée.</p> <p>La valeur par défaut est '2'.</p>
log.maxsize	N/D	<p>Spécifie la taille maximale du fichier journal en ko.</p> <p>Lorsque le fichier atteint sa taille maximale, il est remplacé ou écrasé.</p> <p>La valeur de cette propriété est ignorée si la propriété log.filename n'est pas activée.</p> <p>La valeur par défaut est '100' ko.</p>
log.level	-messagelevel	<p>Spécifie l'un des niveaux de message suivants :</p> <ul style="list-style-type: none"> • ERREUR • AVERTISSEMENT • INFO • DEBOGAGE <p>DEBOGAGE est le niveau le plus prolix. Il inclut tous les niveaux de message.</p> <p>La valeur par défaut est 'AVERTISSEMENT'.</p>

Tableau 20. Propriétés et options de ligne de commande de Probe Rules Utility. (suite)

Propriété	Option de ligne de commande	Description
N/D	-propsfile	Spécifie un fichier de propriétés. Si vous n'indiquez pas de fichier de propriétés, le fichier suivant est utilisé par défaut : 'NCHOME/omnibus/etc/nco_rules_util.props'
repository.root	N/D	Spécifie le niveau supérieur du référentiel de fichiers de règles. Les références de chemin de répertoire ou de fichier relatives sont exprimées relativement à cette valeur. La racine du référentiel doit être un chemin d'accès absolu. Si vous n'indiquez aucune valeur pour cette propriété, elle prend par défaut la valeur de la variable d'environnement NC_RULES_HOME. Si la variable d'environnement NC_RULES_HOME n'est pas définie, cette propriété prend par défaut la valeur du répertoire en cours.
targetpath	N/D	Spécifie le chemin du paramètre de répertoire ou de fichier pour l'action list. La valeur de cette propriété est toujours remplacée par celle du paramètre de ligne de commande chemin. La valeur par défaut est '.'.
N/D	-version	Affiche les informations de version de l'outil.

Tâches associées:

«Vérifier la synchronisation d'un fichier de règles», à la page 332

L'utilitaire Probe Rules Utility vous permet de vérifier qu'un fichier de règles de sonde est bien synchronisé avec la même version de ce fichier stockée dans un référentiel central ou dans un système de fichiers.

Référence associée:

«Table registry.probes», à la page 366

La table registry.probes est utilisée pour effectuer le suivi des informations d'exécution dynamiques relatives aux sondes. Lorsqu'une sonde se connecte au serveur ObjectServer, elle enregistre des informations la concernant dans la table registry.probes. La sonde contrôle les données entrées dans la table.

Variables d'environnement et échappement de caractères

L'outil Probe Rules Utility ne reconnaît que certains formats pour les variables d'environnement. En outre, l'échappement de certains caractères n'est pas pris en charge.

Variables d'environnement

Probe Rules Utility reconnaît les formats de variable d'environnement suivants :

- `$NOM_VAR_ENV`
- `%NOM_VAR_ENV%`
- `${NOM_VAR_ENV}`

Ces formats s'appliquent aux instructions `include` et `lookup` contenues dans les fichiers de règles, ainsi qu'aux propriétés spécifiées dans le fichier `nco_rules_util.props`.

L'outil recherche la valeur indiquée par la variable d'environnement et la remplace par le contenu de la variable. Si la variable d'environnement n'existe pas, elle est remplacée par une chaîne vide.

Par exemple, supposons que `$NC_RULES_HOME` ait la valeur `/var/rulesfiles` et qu'un fichier de règles contienne l'instruction suivante :

```
include $NC_RULES_NOME/subdir/some.include
```

L'outil analyse l'instruction comme étant `include /var/rulesfiles/subdir/some.include`.

Remarque : Si vos fichiers de règles contiennent des variables d'environnement, vous devez définir ces variables avant d'exécuter Probe Rules Utility afin que ce dernier puisse les résoudre.

Échappement de caractères

Probe Rules Utility ne permet pas l'échappement des caractères suivants : `$`, `%`, `{` et `}`.

Par exemple, dans l'instruction `include` suivante, l'outil n'échappe pas le caractère `$` mais analyse `$name` comme une variable d'environnement :

```
include /random/file\$name/the.rules
```

Ces restrictions d'échappement s'appliquent aux instructions `include` et `lookup` contenues dans les fichiers de règles, ainsi qu'aux propriétés spécifiées dans le fichier `nco_rules_util.props`.

Chapitre 3. Personnalisations du fichier de règles de sonde

Vous pouvez étendre les fonctionnalités des sondes à l'aide de certaines ressources fournies dans le répertoire \$NCHOME/omnibus/extensions de votre installation Tivoli Netcool/OMNIbus. Des modèles de fichiers de règle de sonde et SQL permettent de personnaliser les sondes pour détecter les débordements d'événements ou les taux d'événements anormaux, et pour appliquer l'auto-surveillance à l'aide de données statistiques capturées et traitées par la sonde.

Détection de débordements d'événements et de taux d'événements anormaux

Les débordements d'événements peuvent provoquer une indisponibilité du serveur ObjectServer et entraîner de longues périodes pendant lesquelles il n'y a aucune visibilité des événements réseau. Un taux de réception d'événements anormalement faible ou élevé peut également indiquer un incident ou une modification de la source, qui doit être traité.

Vous pouvez configurer une sonde pour détecter une condition de débordement d'événement ou des taux d'événements anormaux et effectuer des actions correctives. Certains scénarios courants sont les suivants :

- Lorsqu'un débordement d'événements est détecté, vous souhaitez supprimer toutes les alertes ultérieures jusqu'à ce que le taux d'événements revienne sous un seuil prédéfini, qui indique que le débordement d'événements est terminé.
- Lorsqu'un débordement d'événements est détecté, vous voulez rediriger toutes les alertes ultérieures vers un serveur ObjectServer différent jusqu'à ce que le taux d'événements revienne sous un seuil prédéfini, qui indique que le débordement d'événements est terminé.
- Lorsqu'un débordement d'événements est détecté, vous voulez envoyer une alerte d'information au serveur ObjectServer au début du débordement d'événements et une autre à la fin de ce débordement.
- Lorsqu'un débordement d'événements est détecté, vous voulez uniquement transmettre les alertes importantes et critiques au serveur ObjectServer principal et supprimer toutes les autres alertes ou les rediriger vers un autre ObjectServer jusqu'à ce que le débordement soit terminé.
- Lorsqu'un taux de réception d'événements anormal est détecté, vous voulez envoyer une alerte informationnelle au serveur ObjectServer, qui décrit la nature du taux d'événements anormal.

Lorsque le contrôle de débordement est activé et qu'un taux d'événements inhabituellement élevé ou faible est détecté, la liste des événements peut être remplie avec plusieurs événements pour le même problème. Ces événements ne sont pas supprimés de la liste des événements lorsque le problème est résolu. Vous pouvez supprimer manuellement entrées indésirables.

Deux fichiers de règle secondaires sont fournis pour configurer une sonde de façon à ce qu'elle détecte les débordements d'événements et les taux d'événements anormaux. Ces fichiers de règles figurent dans le répertoire \$NCHOME/omnibus/extensions/eventflood. Voici les détails de ces fichiers :

- flood.rules : ce fichier de règles de débordement contient les calculs de taux d'événements et la logique de détection des débordements d'événements et des

taux d'événements anormaux. Il calcule un taux moyen de réception des événements pour la sonde puis définit les seuils de taux d'événement inférieur et supérieur en tant que pourcentage configurable de ce taux d'événements moyen. Le taux d'événements actuel est comparé à ces seuils de taux d'événements pour déterminer si la sonde est soumise à un taux de réception d'événement anormal. Le fichier de règles de débordement utilise également des seuils prédéfinis pour un taux d'événement normal et un taux de débordement d'événements afin de déterminer si la sonde est soumise à un débordement d'événements. Des actions correctives facultatives sont incluses pour générer des alertes informationnelles, supprimer des alertes ou les rediriger.

De plus, le fichier de règles de débordement écrit un flux de messages dans le journal de la sonde pour détailler les résultats du traitement. Pour que ce fichier journal puisse contenir les messages, il peut s'avérer nécessaire d'augmenter sa taille maximale actuelle.

- `flood.config.rules` : ce fichier de règles définit les éléments de configuration et leurs valeurs, utilisées dans le fichier `flood.rules`. Ces éléments incluent des multiplicateurs et limites de seuils définis, la durée d'échantillonnage, les fenêtres temporelles et le nombre maximum d'événements autorisés pour calculer les charges moyenne, de débordement et anormale ainsi que les variables associées aux actions correctives.

Pour configurer la sonde, vous devez intégrer des copies mises à jour de ces deux fichiers dans le fichier de règles de sonde principal (obligatoire pour `flood.config.rules`), ou l'un des fichiers de règle secondaires. Lorsque le fichier de règles de sonde est traité, les actions correctives sont effectuées conformément à vos spécifications.

Référence associée:

«Fichier de règles de débordement», à la page 85

Le fichier `flood.rules` vous permet de calculer des taux d'événements afin de détecter des débordements d'événements ou la réception anormale d'événements et de spécifier des actions correctives. Ce fichier doit être utilisé avec le fichier de règles de configuration du débordement `flood.config.rules`.

«Fichier de règles de configuration du débordement», à la page 82

Le fichier `flood.config.rules` vous permet de définir les variables de configuration utilisées pour détecter un débordement d'événements ou un taux d'événements anormal. Ce fichier doit être utilisé avec le fichier de règles de débordement `flood.rules`.

Configuration des sondes pour détecter des débordements d'événements et des débits d'événements anormaux

Vous pouvez configurer des sondes pour détecter des débordements d'événements et des débits d'événements anormaux à l'aide des fichiers de règles secondaires installés dans le répertoire `$NCHOME/omnibus/extensions/eventflood`.

Procédure

Pour activer ces fonctions pour une sonde :

1. Accédez au répertoire `$NCHOME/omnibus/extensions/eventflood`.
2. Copiez les fichiers `flood.config.rules` et `flood.rules` dans un répertoire local ou distant préféré, dans lequel le fichier de règles principal de la sonde ou tout autre fichier de règles secondaire sont stockés. Supprimez les droits d'accès en lecture seule par défaut des fichiers `flood.config.rules` et `flood.rules`.

3. Editez les fichiers `flood.config.rules` et `flood.rules` en fonction de votre configuration. Vous pouvez mettre en commentaire toutes les sections non requises. Par exemple, si vous ne souhaitez pas supprimer d'alertes au cours d'un débordement d'événements, vous pouvez mettre en commentaire l'instruction conditionnelle dans le fichier `flood.rules` . Dans l'exemple de fichier `flood.rules` , les débits d'événements sont calculés par rapport à toutes les sources d'événements qui envoient des données à la sonde.
4. Utilisez une instruction `include` pour inclure le fichier `flood.config.rules` au tout début de votre ensemble de règles, avant toute instruction de traitement. Ce fichier contient une déclaration de tableau et des instructions `registertarget` , lesquelles doivent être définies au début d'un fichier de règles.

Remarque : Si vous incluez d'autres fichiers de règles qui utilisent la commande `registertarget` (par exemple, `tivoli_eif_virtualization_pt1.rules`), vous devez organiser le fichier de règles de sonde principal pour placer au début de ce fichier toutes les instructions `registertarget` , suivies de toutes les instructions de tableau, puis de toutes les déclarations de variable. Si vous n'incluez pas ces éléments dans cet ordre, le fichier de règles provoque des erreurs de syntaxe lors de l'exécution de la sonde.

5. Utilisez une instruction `include` pour inclure le fichier `flood.rules` (dans le fichier de règles principal ou dans un autre fichier de règles secondaire) dans la section qui définit votre ensemble de règles. Si vous souhaitez supprimer ou dérouter de manière conditionnelle des alertes avec des niveaux de gravité particuliers au cours du débordement d'événements, vous devez inclure le fichier `flood.rules` vers la fin de l'ensemble de règles, à un endroit où la gravité de l'alerte a déjà été déterminée.
6. Après avoir mis à jour le fichier de règles avec les instructions `include` , relisez le fichier de règles avec la sonde.
7. Dans le fichier de propriétés de sonde, définissez la propriété **MaxLogFileSize** sur une valeur qui permet de contenir les messages supplémentaires générés lors du traitement du fichier `flood.rules` .

Concepts associés:

«Fichier de propriétés», à la page 6

Les propriétés de sonde définissent l'environnement dans lequel la sonde est exécutée.

«Fichier de règles», à la page 8

Le fichier de règles définit comment la sonde traite les données d'événement pour créer une alerte significative. Pour chaque alerte, le fichier de règles crée également un identificateur qui identifie de manière unique la source du problème.

Tâches associées:

«Relecture du fichier de règles», à la page 66

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Référence associée:

«Fichier de règles de débordement», à la page 85

Le fichier `flood.rules` vous permet de calculer des taux d'événements afin de détecter des débordements d'événements ou la réception anormale d'événements et de spécifier des actions correctives. Ce fichier doit être utilisé avec le fichier de

règles de configuration du débordement `flood.config.rules`.

«Fichier de règles de configuration du débordement», à la page 82

Le fichier `flood.config.rules` vous permet de définir les variables de configuration utilisées pour détecter un débordement d'événements ou un taux d'événements anormal. Ce fichier doit être utilisé avec le fichier de règles de débordement `flood.rules`.

«Intégration de plusieurs fichiers de règles dans un fichier de règles», à la page 70

Vous pouvez utiliser des instructions `include` pour appeler un ou plusieurs fichiers de règles secondaires à partir d'un fichier de règles principal.

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Protection du serveur ObjectServer contre les afflux d'événements

Configurez le serveur ObjectServer de sorte qu'il mesure les seuils d'événement, qu'il détecte les afflux d'événements et qu'il contrôle le comportement de la sonde en conséquence. Cela vous donne davantage de contrôle sur les afflux d'événements que lorsque vous utilisez uniquement les fichiers de règles de la sonde (`flood.rules`) pour gérer les afflux.

Avant de commencer

Le serveur ObjectServer contrôle le comportement de la sonde en modifiant la valeur de la propriété **FloodControl** via le port de commande. Pour activer cette commande, vous devez effectuer les actions suivantes avant de configurer le serveur ObjectServer :

- Configurez les sondes pour de sorte qu'elles reçoivent des messages par le biais de leurs ports de commande en activant la gestion à distance des sondes.
- Configurez les sondes de sorte qu'elles interviennent lorsqu'un afflux d'événements est détecté.

Conseil : Si vous avez configuré un nom d'utilisateur et un mot de passe pour le port de contrôle, définissez-les dans le fichier `$OMNIHOME/etc/nco_http.props`.

Pourquoi et quand exécuter cette tâche

Deux métriques sont utilisées pour calculer la charge sur le serveur ObjectServer induite par les clients connectés. Une moyenne est calculée sur une période spécifiée et des seuils sont appliqués pour déterminer si le serveur ObjectServer doit activer le contrôle des débordements d'événements. Par défaut, pour chaque métrique, la moyenne est calculée toutes les 60 secondes, pendant une période de 300 secondes. Si la charge moyenne excède l'un des seuils, le contrôle des débordements d'événements est alors appelé et un événement est émis. Ces métriques, seuils et valeurs par défaut sont décrits dans le tableau ci-après.

Tableau 21. Métriques et seuils utilisés pour déterminer quand activer le contrôle des afflux d'événements

Mesure	Seuil	Description
Temps passé par le serveur ObjectServer pour le traitement des requêtes des clients	Temps passé pendant une période de 60 secondes. La valeur par défaut est de 40 secondes.	Pour calculer la valeur, le système lit la table catalog.profiles.
Temps passé par le serveur ObjectServer pour les déclencheurs.	Temps passé pendant une période de 60 secondes. La valeur par défaut est de 30 secondes.	Pour calculer la valeur, le système lit la table catalog.trigger_stats.

Par défaut, le mode de contrôle des débordements d'événements est à nouveau désactivé lorsque le temps de traitement devient inférieur aux deux seuils définis et que le contrôle des débordements d'événements a été activé pendant 300 secondes. Un événement de résolution est émis lorsque le contrôle des débordements d'événements est désactivé.

Toutes les valeurs par défaut et le comportement des déclencheurs sont configurables.

La configuration de la protection contre les afflux d'événements est fournie dans le fichier \$NCHOME/omnibus/extensions/eventflood/flood_control.sql. Pour configurer la protection contre les afflux d'événements, appliquez ce fichier au schéma ObjectServer. La configuration contient les valeurs par défaut.

Procédure

Pour configurer le contrôle des afflux d'événements, procédez comme suit :

1. Accédez au répertoire \$NCHOME/omnibus/extensions/eventflood et copiez le fichier flood_control.sql dans l'emplacement de votre choix.
2. Pour modifier les valeurs par défaut de la protection contre les afflux d'événements, ouvrez le fichier et modifiez le contenu aux points décrits dans le tableau ci-dessous.

Tableau 22. Valeurs par défaut du fichier flood_control.sql à éditer pour modifier les valeurs de la protection contre les afflux d'événements

Section de code	Description
set window_size = 5;	Période, en minutes, au cours de laquelle les valeurs moyennes des métriques sont calculées. La valeur par défaut est 5.
if(avg_trigger_per >= 0.5)	Seuil, exprimé sous forme de fraction de minute, du temps passé pour les déclencheurs. La valeur par défaut est de 0,5, soit 30 secondes dans une minute.
if(avg_client_per >= 0.66)	Seuil, exprimé sous forme de fraction de minute, du temps passé pour le traitement des requêtes SQL des clients. La valeur par défaut est de 0,66, soit 40 secondes dans une minute.

Tableau 22. Valeurs par défaut du fichier `flood_control.sql` à éditer pour modifier les valeurs de la protection contre les afflux d'événements (suite)

Section de code	Description
<code>if(elapsed >= 300)</code>	Temps, en secondes, qui doit s'écouler une fois que les valeurs de protection contre les afflux d'événements sont passées sous les seuils définis avant que cette protection ne soit désactivée.
<code>'ObjectServer ' + getservername() + ' is currently in flood',</code>	Texte de l'événement qui est déclenché lorsque le serveur ObjectServer passe en mode contrôle des afflux d'événements. Ne modifiez pas l'événement <code>' + getservername() + '</code> .
<code>'ObjectServer ' + getservername() + ' is ending flood control',</code>	Texte de l'événement de résolution qui est déclenché lorsque le contrôle des afflux d'événements est désactivé.

3. Appliquez la configuration de la protection contre les afflux d'événements au schéma ObjectServer en exécutant l'interface SQL interactive et en émettant la commande suivante :

```

UNIX      Linux      $NCHOME/omnibus/bin/nco_sql -user nom_utilisateur
-i mot_de_passe -server nom_serveur < chemin_répertoire/
flood_control.sql

```

```

Windows   "%NCHOME%\omnibus\bin\isql" -U nom_utilisateur -P mot_de_passe
-S nom_serveur -i chemin_répertoire/flood_control.sql

```

Dans ces commandes, *nom_utilisateur* correspond à un nom d'utilisateur valide, *mot_de_passe* au mot de passe correspondant, *nom_serveur* au nom du serveur ObjectServer et *chemin_répertoire* au chemin qualifié complet du répertoire du fichier `.sql`.

Fichier de règles de configuration du débordement

Le fichier `flood.config.rules` vous permet de définir les variables de configuration utilisées pour détecter un débordement d'événements ou un taux d'événements anormal. Ce fichier doit être utilisé avec le fichier de règles de débordement `flood.rules`.

Les entrées du fichier `flood.config.rules` ainsi que les actions que vous pouvez effectuer pour modifier les valeurs sont décrites dans le tableau suivant. Les entrées sont indiquées suivant leur ordre de définition dans le fichier, en commençant par le haut.

Tableau 23. Entrées du fichier `flood.config.rules`

Entrée	Description	Action
<code>DefaultOS = registertarget(%Server, %ServerBackup, "alerts.status")</code>	Cette instruction enregistre comme cible des alertes le serveur ObjectServer par défaut (ainsi que le serveur ObjectServer de sauvegarde, s'il est configuré). Dans le fichier de règles de débordement, il s'agit du serveur ObjectServer cible, auquel une alerte informationnelle est envoyée lorsque le taux d'événements actuel des sources de sonde est anormalement élevé ou faible ou au début et à la fin d'un débordement d'événements. La table par défaut à laquelle l'alerte est envoyée est <code>alerts.status</code> .	Remplacez le nom de table d'alertes par un nom valide de votre choix.

Tableau 23. Entrées du fichier `flood.config.rules` (suite)

Entrée	Description	Action
<code>#FloodEventOS = registertarget("NCOMS_BK", "", "alerts.status")</code>	Cette ligne mise en commentaire enregistre un serveur ObjectServer de sauvegarde, NCOMS_BK. Dans le fichier de règles de débordement, il s'agit d'un serveur ObjectServer cible alternatif, vers lequel les alertes de niveaux de gravité particuliers peuvent être redirigées pendant un débordement d'événements.	Supprimez la mise en commentaire de cette ligne si vous souhaitez rediriger les alertes vers cet ObjectServer en cas de détection d'un débordement d'événements. Remplacez le nom de serveur ObjectServer et le nom de table d'alertes par des noms valides de votre choix.
<code>array event_rate_array</code>	Ce tableau est défini pour contenir toutes les variables de calcul de taux d'événement. Ces variables sont utilisées tout au long du fichier de règles de débordement.	N/D
<code>\$average_event_rate_time_ window</code> <code>\$average_event_rate_max_ sample_size</code>	Ces éléments stockent des valeurs utilisées pour calculer le taux moyen (ou normal) de réception des événements : <ul style="list-style-type: none"> L'élément <code>\$average_event_rate_time_window</code> définit la fenêtre temporelle maximale (en secondes) pour laquelle les événements sont conservés. Cette valeur indique une fenêtre temporelle tournante, mise à jour en appelant la fonction <code>updateLoad</code>. L'élément <code>\$average_event_rate_time_window</code> définit également la <i>période de formation</i>, qui représente la durée nécessaire pour que la sonde détermine le taux d'événements moyen ou normal. L'élément <code>\$average_event_rate_max_sample_size</code> définit le nombre maximum d'événements à conserver pendant la fenêtre temporelle de taux d'événement moyen. <p>Dans le fichier de règles de débordement, ces éléments sont utilisés pour capturer le nombre d'événements des <i>n</i> dernières secondes avant l'heure actuelle et pour calculer le taux d'événements moyen pour cette période.</p>	Modifiez les valeurs par défaut en fonction de vos besoins.
<code>\$flood_detection_time_window</code> <code>\$flood_detection_max_sample_ size</code>	Ces éléments stockent des valeurs utilisées pour calculer le taux de détection des débordements d'événements afin de déterminer si un débordement d'événements est imminent : <ul style="list-style-type: none"> L'élément <code>\$flood_detection_time_window</code> définit la fenêtre temporelle maximale (en secondes) pour laquelle les événements sont conservés. Cette valeur indique une fenêtre temporelle tournante, mise à jour en appelant la fonction <code>updateLoad</code>. L'élément <code>\$flood_detection_max_sample_size</code> définit le nombre maximum d'événements à conserver pendant cette période. <p>Dans le fichier de règles de débordement, ces éléments sont utilisés pour capturer le nombre d'événements des <i>n</i> dernières secondes avant l'heure actuelle et pour calculer le taux de détection des débordements pour cette période.</p>	Modifiez les valeurs par défaut en fonction de vos besoins.
<code>\$flood_detection_startup_time</code>	Cet élément définit le nombre de secondes pendant lesquelles la sonde est exécutée avant que la détection des débordements d'événements puisse commencer.	Définissez une valeur.
<code>\$anomaly_detection_time_ window</code> <code>\$anomaly_detection_max_ sample_size</code>	Ces éléments stockent des valeurs utilisées pour calculer le taux de réception des événements pour la détection d'un flux anormal : <ul style="list-style-type: none"> L'élément <code>\$anomaly_detection_time_window</code> définit la fenêtre temporelle maximale (en secondes) pour laquelle les événements sont conservés. Cette valeur indique une fenêtre temporelle tournante, mise à jour en appelant la fonction <code>updateLoad</code>. L'élément <code>\$anomaly_detection_max_sample_size</code> définit le nombre maximum d'événements à conserver pendant cette période. <p>Dans le fichier de règles de débordement, ces éléments sont utilisés pour capturer le nombre d'événements des <i>n</i> dernières secondes avant l'heure actuelle et pour calculer le taux d'événements pour cette période.</p>	Modifiez les valeurs par défaut en fonction de vos besoins.

Tableau 23. Entrées du fichier `flood.config.rules` (suite)

Entrée	Description	Action
<p><code>\$flood_detection_event_rate_flood_threshold</code></p> <p><code>\$flood_detection_event_rate_normal_threshold</code></p>	<p>Ces éléments stockent des valeurs utilisées pour spécifier les seuils de taux d'événements pour la détection d'un débordement d'événements ou d'un taux d'événements normal.</p> <p>Si le nombre d'événements reçus par seconde dépasse la valeur spécifiée pour l'élément <code>\$flood_detection_event_rate_flood_threshold</code>, la détection du débordement d'événements est déclenchée.</p> <p>Si le nombre d'événements reçus par seconde est inférieur à la valeur spécifiée pour l'élément <code>\$flood_detection_event_rate_normal_threshold</code>, un taux d'événements normal est supposé.</p>	<p>Modifiez les valeurs par défaut en fonction de vos besoins.</p> <p>Assurez-vous que la valeur de <code>\$flood_detection_event_rate_normal_threshold</code> est inférieure à celle de <code>\$flood_detection_event_rate_flood_threshold</code>.</p>
<p><code>\$lower_event_rate_threshold_multiplieur</code></p> <p><code>\$upper_event_rate_threshold_multiplieur</code></p>	<p>L'élément <code>\$lower_event_rate_threshold_multiplieur</code> définit la valeur du facteur utilisé pour calculer le seuil de taux d'événements inférieur pour la détection d'un taux d'événements anormal.</p> <p>L'élément <code>\$upper_event_rate_threshold_multiplieur</code> définit la valeur du facteur utilisé pour calculer le seuil de taux d'événements supérieur pour la détection d'un taux d'événements anormal.</p> <p>Dans le fichier de règles de débordement, le taux d'événements moyen est multiplié par ces valeurs pour définir les seuils de détermination de taux d'événements anormalement faibles ou élevés.</p>	<p>Modifiez les valeurs par défaut en fonction de vos besoins.</p>
<p><code>\$discard_event_during_flood</code></p>	<p>Cet élément définit si une alerte est supprimée pendant un débordement d'événements. La valeur 1 équivaut à TRUE et la valeur 0 équivaut à FALSE.</p> <p>Dans le fichier de règles de débordement, si la valeur de <code>\$discard_event_during_flood</code> est 1 et que la gravité de l'alerte est inférieure à la valeur spécifiée pour <code>\$forward_event_minimum_severity</code>, l'alerte est supprimée.</p>	<p>Modifiez la valeur par défaut en fonction de vos besoins.</p>
<p><code>\$divert_event_during_flood</code></p>	<p>Cet élément définit si une alerte est redirigée vers un autre ObjectServer pendant un débordement d'événements. La valeur 1 équivaut à TRUE et la valeur 0 équivaut à FALSE.</p> <p>Dans le fichier de règles de débordement, si la valeur de <code>\$divert_event_during_flood</code> est 1 et que la gravité de l'alerte est inférieure à la valeur spécifiée pour <code>\$forward_event_minimum_severity</code>, l'alerte est redirigée.</p>	<p>Pour rediriger une alerte d'une gravité particulière, assurez-vous que la valeur de <code>\$divert_event_during_flood</code> est définie sur 1 dans le fichier <code> flood.config.rules </code>.</p> <p>Assurez-vous également que l'instruction <code>registertarget</code> avec la cible <code>FloodEventOS</code> (définie au début du fichier) ne soit pas mise en commentaire et qu'elle soit configurée avec le nom et la table du serveur ObjectServer appropriés.</p>
<p><code>\$forward_event_minimum_severity</code></p>	<p>Cet élément est défini sur la valeur 4 pour indiquer que les événements avec une gravité majeure ou critique doivent être transmis au serveur ObjectServer principal pendant un débordement d'événements.</p> <p>Dans le fichier de règles de débordement, cet élément est utilisé dans la condition IF qui définit si l'alerte est supprimée ou redirigée pendant un débordement d'événements.</p>	<p>Acceptez ou modifiez la valeur par défaut en fonction de vos besoins.</p>

Référence associée:

«Fichier de règles de débordement», à la page 85
 Le fichier `flood.rules` vous permet de calculer des taux d'événements afin de détecter des débordements d'événements ou la réception anormale d'événements et de spécifier des actions correctives. Ce fichier doit être utilisé avec le fichier de règles de configuration du débordement `flood.config.rules` .

Fichier de règles de débordement

Le fichier `flood.rules` vous permet de calculer des taux d'événements afin de détecter des débordements d'événements ou la réception anormale d'événements et de spécifier des actions correctives. Ce fichier doit être utilisé avec le fichier de règles de configuration du débordement `flood.config.rules`.

La logique du fichier `flood.rules` est décrite pour vous aider à comprendre l'exemple de configuration fourni.

Lors du premier traitement du fichier de règles par la sonde, le tableau (`event_rate_array`) est initialisé et les variables du tableau de taux d'événements sont utilisées pour :

- Définir la fenêtre temporelle tournante et le nombre maximum d'événements qui peuvent être utilisés pour calculer une charge moyenne, une charge de détection de débordement et une charge de détection d'anomalie. Les charges sont définies au format `fenêtre_temporelle_en_secondes.max_nombre_échantillons` en utilisant des éléments spécifiés dans le fichier `flood.config.rules`.
- Définir le mode de taux d'événements sur *normal*.
- Stocker l'horodatage actuel comme heure de démarrage pour la sonde.
- Indiquer que le taux d'événements moyen n'est pas encore calculé.

Calculs de taux d'événements anormaux

Lors des premières secondes de `$average_event_rate_time_window` (10 secondes par défaut) après le démarrage de la sonde, le comptage d'événements est géré afin de calculer un taux d'événements moyen pour la sonde.

A la fin de cette période, les seuils de taux d'événements supérieur et inférieur sont calculés en tant que pourcentages du taux d'événements moyen. Les éléments `$upper_event_rate_threshold_multiplier` et `$lower_event_rate_threshold_multiplier`, définis dans le fichier `flood.config.rules`, sont utilisés pour calculer ces seuils. Une fois le taux moyen déterminé, la sonde vérifie régulièrement le taux d'événements actuel et le compare aux seuils de taux d'événements supérieur et inférieur comme suit :

1. La fonction `update_load` est utilisée pour capturer la fenêtre temporelle (avant l'heure actuelle) et le comptage d'événements, utilisés pour déterminer le taux d'événements actuel pour les événements anormaux. Notez que la fenêtre temporelle par défaut d'une minute, définie par l'élément `$anomaly_detection_time_window`, est utilisée.
2. La fonction `get_load` est utilisée pour calculer le taux d'événements actuel en tant qu'événements par seconde.
3. Le taux d'événements actuel est comparé aux seuils de taux d'événements supérieur et inférieur afin de déterminer si la sonde est soumise à un taux de réception d'événements anormal.

Si un nombre d'événements anormalement bas ou élevé est détecté, la fonction `genevent` est utilisée pour générer et envoyer des alertes d'information au serveur ObjectServer cible enregistré dans le fichier `flood.config.rules` sous `DefaultOS`.

Par exemple, supposez que 200 événements sont reçus pendant la fenêtre temporelle de taux d'événements moyen, ce qui produit un taux d'événements moyen de 20 événements par seconde. Supposez également que l'élément

`$upper_event_rate_threshold_multiplier` est défini sur 5 et que l'élément `$lower_event_rate_threshold_multiplier` est défini sur 0.1 dans le fichier `flood.config.rules`.

Le seuil de taux d'événements supérieur peut être calculé comme suit :

taux d'événements moyen * 5 = 100 événements par seconde

Le seuil de taux d'événements inférieur peut être calculé comme suit :

taux d'événements moyen * 0.1 = 2 événements par seconde

Si le taux d'événements actuel est calculé comme étant de 120 événements par seconde, la sonde génère et envoie une alerte au serveur ObjectServer Default0S, avec des détails sur le taux d'événements élevé. Si le taux d'événements actuel est calculé comme étant 1 événement par seconde, la sonde génère et envoie une alerte au serveur ObjectServer Default0S, avec des détails sur le taux d'événements faible.

Calculs de détection de débordement

Lorsque la sonde démarre, une période d'exclusion est observée pour la détection du débordement. Cette période a une durée fixe à partir de l'heure de démarrage de la sonde, définie par l'élément `$flood_detection_startup_time` dans le fichier `flood.config.rules`.

A la fin de cette période, la fonction `updateLoad` est utilisée pour capturer la fenêtre temporelle (avant l'heure actuelle) et le comptage d'événements, utilisés pour déterminer le taux d'événements actuel pour la détection du débordement. La fonction `getLoad` est ensuite utilisée pour calculer le taux d'événements actuel par seconde. Le taux d'événements actuel est comparé aux seuils de taux d'événements, définis dans le fichier `flood.config.rules` pour un débordement d'événements et un taux d'événements normal. Le mode d'événement est ensuite défini sur `flood` ou `normal`, le cas échéant.

Si le mode d'événement actuel est `flood`, la sonde détermine si le débordement d'événements vient de commencer, s'il est en cours ou s'il s'est terminé et effectue l'action appropriée :

- La fonction `genevent` est utilisée pour générer et envoyer une alerte informationnelle au serveur ObjectServer cible enregistré dans le fichier `flood.config.rules` comme `Default0S`. Cette alerte informationnelle indique qu'un débordement d'événements vient de commencer ou s'il est terminé et inclut des détails sur ce débordement.
- Pendant le débordement d'événements, des alertes peuvent être supprimées si leur gravité est inférieure à un niveau minimum défini. La configuration par défaut supprime les alertes avec une valeur de gravité inférieure à 4 (majeur).
- Pendant le débordement d'événements, des alertes peuvent également être redirigées vers un serveur ObjectServer si leur gravité est inférieure à un niveau minimum défini. Cet ObjectServer est enregistré dans le fichier `flood.config.rules` comme cible (`FloodEvent0S`) des événements pendant un débordement. La configuration par défaut redirige les événements avec une valeur de gravité inférieure à 4 (majeur).

Consignation des messages

Différents messages sont écrits dans le fichier journal pendant le traitement du fichier de règles de débordement. Les détails enregistrés incluent :

- L'horodatage de démarrage de la sonde
- Le taux d'événements moyen
- Les charges d'événement
- Les taux d'événements anormalement faibles ou élevés
- Les taux d'événements de détection de débordement, l'état du débordement, les actions correctives, le nombre d'événements et la durée du débordement

Référence associée:

«Fichier de règles de configuration du débordement», à la page 82

Le fichier `flood.config.rules` vous permet de définir les variables de configuration utilisées pour détecter un débordement d'événements ou un taux d'événements anormal. Ce fichier doit être utilisé avec le fichier de règles de débordement `flood.rules`.

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

«Envoi d'alertes à des serveurs ObjectServer et tables alternatifs», à la page 50

Les fonctions `registertarget`, `genevent`, `settarget` et `setdefaulttarget` permettent d'envoyer des alertes à un ou plusieurs serveurs ObjectServer et de définir la distribution de ces alertes entre ces serveurs.

Activation de l'auto-surveillance des sondes

Vous pouvez configurer les sondes de telle façon qu'elles génèrent des événements de signaux de présence ProbeWatch ; ce mécanisme d'auto-surveillance permet de surveiller la performance, de diagnostiquer les problèmes de performances, et de mettre en évidence les goulots d'étranglement avant qu'ils ne commencent à affecter le système.

Un événement de signal de présence ProbeWatch est généré par la sonde, et n'est pas déclenché par un événement (ou une absence d'événements) de l'entité gérée. L'événement de pulsations ProbeWatch est généré à un intervalle configurable, défini par la propriété **ProbeWatchHeartbeatInterval**. Cet intervalle est défini par défaut sur 60 secondes. Un événement de signal de présence ProbeWatch peut être utilisé comme signal de présence pour confirmer que la sonde est en état de marche, ou pour transporter des statistiques relatives à la sonde :

- La présence d'un événement de signal de présence ProbeWatch régulier vous permet de déterminer si une sonde est inactive en raison de l'absence d'événements entrants émis par sa source, en raison d'un échec de la sonde, ou d'un échec de la communication avec le serveur ObjectServer. La valeur spécifiée pour la propriété **ProbeWatchHeartbeatInterval** indique le délai d'attente maximum qui s'écoule entre son dernier signal et la notification indiquant qu'elle est toujours en marche. Si aucun événement n'a été envoyé dans la fenêtre de temps **ProbeWatchHeartbeatInterval** précédente, la sonde indique qu'elle est toujours active (bien qu'elle ne reçoive pas d'événements) en envoyant un événement de pulsations ProbeWatch au serveur ObjectServer. La zone Récapitulatif de cet événement de signal de présence ProbeWatch est renseignée avec le texte suivant : Heartbeat ...

- Un événement de signal de présence ProbeWatch permet également de transmettre des données statistiques pour les sondes, telles que le débit de traitement des sondes, et l'utilisation de la ressource mémoire et de l'unité centrale. Les sondes individuelles peuvent capturer les informations sur l'utilisation et les ressources, qui sont alors gérées dans le fichier de règles afin de calculer les métriques à l'aide d'un jeu de propriétés dédié. Ces métriques peuvent être transférées dans le serveur ObjectServer dans un événement de signal de présence ProbeWatch unique à l'aide des fonctions `nvp_add` spécifiant les paires nom-valeur des attributs étendus, ou dans plusieurs événements de signal de présence ProbeWatch générés à l'aide de la fonction `genevent`. La fréquence de transmission des événements générés au serveur ObjectServer dépend de l'intervalle défini par la propriété **ProbeWatchHeartbeatInterval**.

Les métriques fournies dans les événements de signal de présence ProbeWatch peuvent être analysées pour identifier le mode d'exécution des différents composants système, ainsi que les problèmes potentiels pouvant altérer la performance. Les données peuvent également être assemblées pour être utilisées dans des rapports et des graphiques afin de déterminer le retour sur investissement réalisé.

Référence associée:

«Messages ProbeWatch et TSMWatch», à la page 325

Dans certains cas, une sonde ou TSM génère des événements de lui-même. Ces événements peuvent fournir des informations (tels que les messages de démarrage ou d'arrêt) ou identifier des problèmes.

Configuration de l'auto-surveillance des sondes

Les sondes peuvent être configurées de façon à générer des données statistiques permettant d'évaluer la performance système et de calculer le retour sur investissement.

La figure suivante décrit la configuration pour l'auto-surveillance de sonde.

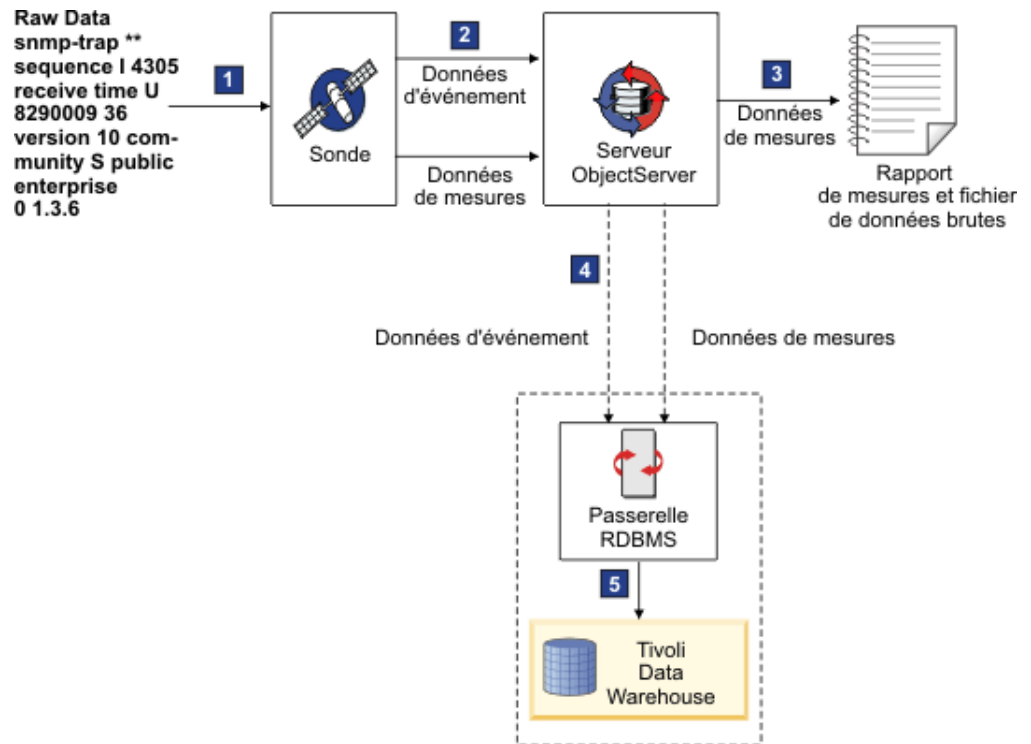


Figure 3. Configuration et flot de données de l'autosurveillance de la sonde

Le flux de configuration est le suivant :

- 1** Les informations d'utilisation et de ressource sont capturées avec des données brutes envoyées à la sonde.
- 2** La sonde traite les informations d'utilisation et de ressource pour mesurer les performances et générer des événements de signal de présence ProbeWatch définis par ces métriques. La sonde traite également les données brutes afin de générer des événements génériques (standard).
Les deux ensembles d'événements sont transmis au serveur ObjectServer.
- 3** Les automatisations ObjectServer sont utilisées pour produire un rapport textuel de base qui récapitule les informations statistiques générées par la sonde.
- 4 et 5** Les données d'événement génériques et les événements de signal de présence ProbeWatch peuvent éventuellement être exportés du serveur vers Tivoli Data Warehouse à l'aide d'une passerelle RDBMS (système de gestion de base de données relationnelle). Ces métriques peuvent également être utilisées dans les tâches ultérieures de gestion de rapports dans Tivoli Common Reporting.

Remarque : Ces étapes (4 et 5) sont hors de portée de la fonction d'auto-surveillance de sonde fournie par Tivoli Netcool/OMNIBus. L'intégration avec IBM Tivoli Monitoring (qui inclut Tivoli Data Warehouse) et des tâches de configuration supplémentaires sont requises pour intégrer la gestion de rapports.

Fichiers de configuration Tivoli Netcool/OMNIBus pour l'auto-surveillance des sondes

Lorsque vous installez Tivoli Netcool/OMNIBus, des fichiers de configuration sont fournis pour configurer les sondes afin qu'elles collectent et traitent des données statistiques pour l'auto-surveillance. Des modèles de fichiers de configuration sont disponibles dans le répertoire `$NCHOME/omnibus/extensions/roi`.

Les détails des fichiers de configuration sont les suivants :

- Fichier `probestats.sql` : ce fichier comporte un ensemble d'automatisations permettant de capturer les données statistiques entrantes collectées pour une sonde, et de consigner ces données dans un fichier. Les tables sont également créées dans le serveur ObjectServer pour stocker les mesures de sonde et d'enregistrer la dernière période de génération de rapport sur les données. Notez que les mesures de sonde sont stockées dans la table `master.probestats` spécialement créée, et non pas dans la table `alerts.status`.

Le fichier journal créé est similaire au journal de profilage, et inclut :

- Les mesures individuelles de chaque sonde connectée ; par exemple, le nombre d'événements traités, générés, et supprimés depuis la dernière période de génération de rapports
- Un ensemble de mesures assemblées ; par exemple, le nombre total d'insertions `alerts.details` et `alerts.journal` depuis la dernière période de génération de rapports

Vous pouvez consulter ce fichier SQL afin de vous familiariser avec les changements potentiels qui seront appliqués au serveur ObjectServer.

- Fichier `probewatch.include` : ce fichier de règles personnalisé peut être utilisé avec les sondes, et doit être intégré au fichier de règles principal de la sonde. Le fichier `probewatch.include` est basé sur les règles spécifiques à ProbeWatch originales par défaut. Ce fichier contient de nouvelles instructions CASE pour deux messages ProbeWatch supplémentaires et pour les événements de signal de présence ProbeWatch, qui transmettent les données statistiques.

Le fichier de règles `probewatch.include` est général et commun à toutes les sondes. Vous pouvez personnaliser et partager ce fichier sur plusieurs sondes (ou toutes les sondes) afin de centraliser l'administration d'événements de signal de présence ProbeWatch.

- Fichier `Omnibus_TDW_Reports_ROI.zip` : ce fichier d'archive contient un ensemble de modèles de rapport qui requièrent une personnalisation de l'utilisateur, et une intégration dans Tivoli Data Warehouse et Tivoli Common Reporting. Pour procéder à la configuration, vous devez savoir utiliser ces composants.

Un ensemble de propriétés statistiques est également ajouté à la configuration des sondes. Ces propriétés sont utilisées pour collecter les informations d'utilisation et de ressource spécifiques à chaque sonde. Les propriétés statistiques sont différentes des propriétés de sonde standard car elles ne peuvent pas être définies sur une valeur descriptive dans le fichier de propriétés, et ne peuvent pas être exécutées comme options de ligne de commande. Ces noms de propriétés portent toutes le préfixe **Op1Stats**, et sont affichés dans la sortie obtenue lorsque la sonde est exécutée avec l'option de ligne de commande `-dumpprops`.

Les propriétés statistiques sont les suivantes :

Tableau 24. Propriétés statistiques pour les sondes

Propriété	Description
Op1StatsCPUTimeSec	Le temps UC utilisé par la sonde en secondes. Exemple : Si 6,002345 de temps UC ont été utilisées par la sonde, Op1StatsCPUTimeSec = 6
Op1StatsCPUTimeUSec	Le composant inférieur à la seconde du temps UC utilisé par la sonde, en millionièmes de seconde. Exemple : Si 6.002345 secondes de temps UC a été utilisé par la sonde, Op1StatsCPUTimeUSec = 2345
Op1StatsRulesFileTimeSec	Le temps de traitement des règles en secondes. Exemple : Si 4,372700 secondes sont nécessaires au traitement des règles, Op1StatsRulesFileTimeSec = 4
Op1StatsRulesFileTimeUSec	Le composant inférieur à la seconde du temps UC utilisé par la sonde, en millionièmes de seconde. Exemple : Si 4,372700 secondes sont nécessaires au traitement de règles, Op1StatsRulesFileTimeUSec = 372700
Op1StatsProbeStartTime	L'heure (en temps UNIX epoch) à laquelle la sonde a démarré.
Op1StatsMemoryInUse	L'empreinte mémoire (en Ko) de la sonde.
Op1StatsNumberEvents	Nombre d'événements (y compris les événements ProbeWatch) que la sonde a reçu de sa source d'événements depuis le lancement de la sonde.
Op1StatsNumberEventsDiscarded	Nombre d'événements supprimés après le traitement de la règle.
Op1StatsNumberEventsGenerated	Nombre d'événements générés à l'aide de la fonction genevent du fichier de règles.

Configuration des sondes pour l'auto-surveillance

Pour appliquer l'auto-surveillance, vous pouvez configurer une sonde afin qu'elle collecte les données statistiques sur la quantité de mémoire utilisée pour diverses opérations de traitement, et sur le nombre d'événements reçus, supprimés et générés.

Pourquoi et quand exécuter cette tâche

Pour configurer une sonde afin qu'elle collecte et traite les données statistiques :

Procédure

1. Accédez au répertoire `$NCHOME/omnibus/extensions/roi`.
2. Copiez le fichier `probestats.sql` dans le répertoire `$NCHOME/omnibus/etc`, ou dans un autre emplacement privilégié. Appliquez la personnalisation de signal de présence ProbeWatch au schéma ObjectServer en exécutant la commande suivante dans l'interface SQL interactive :

```

UNIX      Linux      $NCHOME/omnibus/bin/nco_sql -user nom_utilisateur
-password mot_de_passe -server nom_serveur < chemin_répertoire/
probestats.sql

```

```
Windows "%NCHOME%\omnibus\bin\isql" -U nom_utilisateur -P mot_de_passe
-S nom_serveur -i chemin_répertoire\probestats.sql
```

Dans ces commandes, *nom_utilisateur* est un nom d'utilisateur valide, *mot_de_passe* est le mot de passe correspondant, *nom_serveur* est le nom du serveur ObjectServer et *chemin_répertoire* est le chemin qualifié complet du répertoire du fichier .sql.

Le fichier *probestats.sql* ajoute un ensemble de tables et de déclencheurs au serveur ObjectServer.

3. Copiez le fichier *\$NCHOME/omnibus/extensions/roi/probewatch.include* dans un répertoire local ou distant préféré, dans lequel est stocké le fichier de règles principal ou tout autre fichier de règles secondaire. Ce fichier est conçu pour remplacer la logique de la section ProbeWatch de votre fichier de règles principal, généralement codée comme suit :

```
if(match(@Manager, "ProbeWatch"))
{
    switch(@Summary)
    {
        case "Running ...":
            @Severity = 1
            @AlertGroup = "probestat"
            @Type = 2
        case "Going Down ...":
            @Severity = 5
            @AlertGroup = "probestat"
            @Type = 1
        default:
            @Severity = 1
    }
    @AlertKey = @Agent
    @Summary = @Agent + " probe on " + @Node + ": " + @Summary
}
else
{
    ...règles spécifiques à la sonde...
}
```

Le code affiché en gras doit être remplacé par une instruction `include` qui vous permet d'inclure le contenu du fichier *probewatch.include*, comme indiqué à l'étape 5, à la page 94.

4. Supprimez les droits d'accès en lecture seule par défaut de votre copie du fichier *probewatch.include* et consultez le fichier pour vous familiariser avec son contenu. Puis, modifiez le fichier comme suit :
 - Mettez à jour les éléments dans la partie supérieure du fichier pour déterminer le mode de traitement des événements de signal de présence ProbeWatch. Utilisez le signe dièse (#) pour commenter les éléments qui ne sont pas nécessaires. La logique de traitement de ces éléments est codée dans l'instruction `case "Heartbeat ..."` de la section ProbeWatch du fichier.

Tableau 25. *Éléments des événements de signal de présence ProbeWatch*

Élément	Action
\$Op1Heartbeat_discard	Définissez cette valeur sur 1 si vous souhaitez supprimer l'événement de signal de présence ProbeWatch. Définissez cette valeur sur 0 si vous souhaitez transmettre l'événement de signal de présence ProbeWatch au serveur ObjectServer.

Tableau 25. Eléments des événements de signal de présence ProbeWatch (suite)

Elément	Action
\$OplHeartbeat_populate_master_probestats	<p>Définissez cette valeur sur 1 pour permettre la génération d'un nouvel événement de mesure de sonde à l'aide de la fonction <code>genevent</code>, définie dans l'instruction <code>case "Heartbeat ..."</code>. Les données d'événements sont un ensemble de mesures de sonde <code>OplStats</code>, transmises à la table <code>master.probestats</code> créée lors de l'exécution du script <code>probestats.sql</code>.</p> <p>Définissez cette valeur sur 0 si vous ne souhaitez pas générer cet événement pour l'insérer dans la table <code>master.probestats</code>.</p>
\$OplHeartbeat_write_to_probe_log	<p>Définissez cette valeur sur 1 si vous souhaitez enregistrer les mesures <code>OplStats</code> dans le fichier journal de la sonde. Les détails sont consignés au niveau <code>INFO</code>. Les détails des mesures consignés sont définis dans l'instruction <code>case "Heartbeat ..."</code>.</p> <p>Définissez cette valeur sur 0 si vous ne souhaitez pas enregistrer les mesures dans le fichier journal.</p>
\$OplHeartbeat_generate_threshold_events	<p>Définissez cette valeur sur 1 si vous souhaitez générer les événements de seuil indiquant qu'une mesure de sonde spécifique a dépassé un seuil défini. Aucun code n'est fourni par défaut dans ce fichier de règles pour les événements de seuil car les préférences individuelles peuvent être très différentes. Si vous devez définir des événements de seuil, vous devez d'abord déterminer quels seuils vous souhaitez surveiller. Puis, dans l'instruction <code>case "Heartbeat ..."</code>, entrez le code permettant de générer les événements de seuil.</p>

- Outre les instructions `CASE` standard, ce fichier inclut les deux instructions `CASE` suivantes, qui contiennent la logique de deux nouveaux événements `ProbeWatch` apportant des commentaires lorsqu'une sonde relit ses fichiers de règles à réception d'un signal `SIGHUP`. La première instruction `CASE` s'applique lorsque la relecture aboutit :

```
case "Rules file reread upon SIGHUP successful ...":
    @Severity = 1
    @AlertGroup = "rules"
    @Type = 2
```

La deuxième instruction `CASE` s'applique lorsque la relecture échoue. Cette section de code inclut deux éléments (`$msg` et `$file`), où `$msg` est le message d'erreur figurant dans le fichier journal de sonde, et `$file` est le nom du fichier où se trouve l'erreur.

```
case "Rules file reread upon SIGHUP failed ...":
    @Severity = 4
    @AlertGroup = "rules"
    @Type = 1
    if( exists( $msg ) )
    {
```

```

        @Summary = @Summary + ("+$msg+")"
    }
    if( exists( $file ) )
    {
        @Summary = @Summary + " in file "+$file
    }

```

Si vous n'avez pas besoin de ces données, utilisez la fonction `discard` pour éviter qu'elles soient envoyées au serveur ObjectServer.

- L'instruction CASE finale (case "Heartbeat ...") contient un ensemble d'instructions conditionnelles permettant de calculer les mesures de sonde et de traiter les données. Les instructions IF sont fournies avec la logique afin de supprimer les événements et d'écrire les mesures de sonde dans un fichier journal. L'utilisateur doit également procéder à des saisies :

Tableau 26. Sections case "Heartbeat ..." nécessitant des saisies d'utilisateur

<p>Recherchez la section de code commençant par les lignes suivantes :</p> <pre> if(int(\$Op1Heartbeat_populate_master_probestats) == 1) { log(DEBUG, "HEARTBEAT - SENDING PROBESTATS TO MASTER.PROBESTATS") ... </pre> <p>Cette section de code contient une instruction <code>genevent</code> avec une marque de réservation <code>Default0S</code> qui identifie un serveur ObjectServer cible enregistré. Cette cible doit être définie dans une instruction <code>registertarget</code> du fichier de règles principal. Remplacez cette marque de réservation par le serveur ObjectServer cible vers lequel les événements doivent être envoyés.</p>
<p>Recherchez la section de code commençant par les lignes suivantes :</p> <pre> if(int(\$Op1Heartbeat_generate_threshold_events) == 1) { # # Zone pour générer les événements de seuil définis par # l'utilisateur à l'aide de genevent ... </pre> <p>Si vous définissez l'élément <code>\$Op1Heartbeat_generate_threshold_events</code> sur 1 en haut du fichier, vous devez entrer le code correspondant au type d'événements de seuil à surveiller.</p> <p>Vous pouvez ignorer cette section si vous n'avez pas besoin de définir d'événements de seuil.</p>

- Si vous avez modifié la section `ProbeWatch` de votre fichier de règles principal (généralement `$NCHOME/omnibus/probes/arch/nom_sonde.rules`), vous devez procéder aux mêmes modifications sur le fichier `probewatch.include`.
- Si le fichier de règles principal inclut des sections `ProbeWatch` supplémentaires contenant un code pour les différents messages `ProbeWatch` qui ne sont pas couverts dans le fichier `probewatch.include`, copiez ce code supplémentaire dans le fichier `probewatch.include`.

Conseil : Après avoir effectué toutes les modifications dans le fichier `probewatch.include`, exécutez le vérificateur de syntaxe de règles de sonde (`nco_p_syntax`) pour tester la syntaxe du fichier de règles.

5. Incorporez le fichier `probewatch.include` mis à jour dans votre fichier de règles de sonde principal à l'aide d'une instruction `include`. Vérifiez que le chemin dans l'instruction `include` pointe vers l'emplacement de stockage du fichier `probewatch.include` mis à jour.

```

if(match(@Manager, "ProbeWatch"))
{
    include "chemin_répertoire/probewatch.include"
}

```

```

}
else
{
    ...règles spécifiques à la sonde...
}

```

6. Indiquez l'intervalle, en secondes, à laquelle les messages de signal de présence sont générés, en définissant le paramètre **ProbWatchHeartbeatInterval** dans le fichier de propriétés de sonde.
 - Définissez un nombre positif pour générer les événements
 - Définissez 0 (zéro) ou un nombre négatif pour ne générer aucun événement
7. Vérifiez que le groupe de déclencheurs stats_triggers est activé. Les déclencheurs ajoutés à l'aide du fichier probestats.sql sont affectés à ce groupe de déclencheurs, qui doit être activé pour que les déclencheurs s'exécutent. Activez le groupe de déclencheurs à l'aide Youde Netcool/OMNibus Administrator ou de la commande ALTER TRIGGER GROUP command, comme indiqué dans le *Guide d'administration d'IBM Tivoli Netcool/OMNibus*.
8. Activez le déclencheur probe_statistics_cleanup, qui est configuré par défaut pour supprimer les statistiques de sonde générées plus d'une heure auparavant. Vous pouvez modifier ce paramètre par défaut afin d'augmenter la durée de conservation des statistiques.
9. Démarrez la sonde.

Les mesures de sonde collectées sont enregistrées dans le fichier journal \$NCHOME/omnibus/log/nom_serveur_probestats.log, où nom_serveur est le nom du serveur ObjectServer.

Concepts associés:

Chapitre 4, «Exécution des sondes», à la page 97

Lorsque vous exécutez une sonde, vous pouvez spécifier des propriétés dans un fichier de propriétés ou des commandes sur une ligne de commande pour configurer les paramètres de la sonde.

«Fichier de propriétés», à la page 6

Les propriétés de sonde définissent l'environnement dans lequel la sonde est exécutée.

«Fichier de règles», à la page 8

Le fichier de règles définit comment la sonde traite les données d'événement pour créer une alerte significative. Pour chaque alerte, le fichier de règles crée également un identificateur qui identifie de manière unique la source du problème.

Tâches associées:

«Test de fichiers de règles», à la page 64

Vous pouvez tester la syntaxe d'un fichier de règles à l'aide du vérificateur de syntaxe de règles de sonde, **nco_p_syntax**. Cela se révèle plus efficace que l'exécution de la sonde pour vérifier que la syntaxe du fichier de règles est correcte.

Référence associée:

«Envoi d'alertes à des serveurs ObjectServer et tables alternatifs», à la page 50

Les fonctions registertarget, genevent, settarget et setdefaulttarget permettent d'envoyer des alertes à un ou plusieurs serveurs ObjectServer et de définir la distribution de ces alertes entre ces serveurs.

«Intégration de plusieurs fichiers de règles dans un fichier de règles», à la page 70

Vous pouvez utiliser des instructions include pour appeler un ou plusieurs fichiers de règles secondaires à partir d'un fichier de règles principal.

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129
Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Chapitre 4. Exécution des sondes

Lorsque vous exécutez une sonde, vous pouvez spécifier des propriétés dans un fichier de propriétés ou des commandes sur une ligne de commande pour configurer les paramètres de la sonde.

Une sonde a des valeurs par défaut pour chaque propriété. Dans un fichier de propriétés non édité, toutes les propriétés sont répertoriées avec leurs valeurs par défaut commentées par le symbole dièse (#) au début de la ligne.

Vous pouvez modifier le fichier de propriétés avant d'exécuter la sonde ou pendant son exécution. Si vous éditez le fichier pendant l'exécution de la sonde, les modifications apportées entrent en vigueur au prochain démarrage de la sonde. Vous pouvez éditer les valeurs des propriétés à l'aide d'un éditeur de texte. Pour remplacer une valeur par défaut, vous devez modifier la définition dans le fichier de propriétés puis supprimer le symbole dièse.

Si vous modifiez la définition d'une propriété sur la ligne de commande au démarrage d'une sonde, la valeur par défaut et la définition dans le fichier de propriétés sont remplacées. Pour simplifier la commande que vous entrez pour exécuter la sonde, ajoutez autant de propriétés que possible dans le fichier de propriétés au lieu d'utiliser les options de ligne de commande.

Lorsque vous exécutez une sonde, vous devez également configurer votre fichier de règles pour définir comment la sonde doit traiter les données d'événement. Vous pouvez éditer le fichier de règles avant d'exécuter la sonde ou pendant son exécution. Si vous éditez le fichier de règles pendant l'exécution de la sonde, vous devez forcer la sonde à relire le fichier de règles pour que les modifications entrent en vigueur. Vous pouvez éditer le fichier de règles à l'aide d'un éditeur de texte.

Conseil : Lisez toujours la documentation spécifique à la sonde que vous exécutez pour obtenir des informations de configuration supplémentaires.

Concepts associés:

«Fichier de propriétés», à la page 6

Les propriétés de sonde définissent l'environnement dans lequel la sonde est exécutée.

«Fichier de règles», à la page 8

Le fichier de règles définit comment la sonde traite les données d'événement pour créer une alerte significative. Pour chaque alerte, le fichier de règles crée également un identificateur qui identifie de manière unique la source du problème.

Tâches associées:

«Relecture du fichier de règles», à la page 66

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Utilisation des variables d'environnement OMNIHOME et NCHOME

Tivoli Netcool/OMNIbus version 7.0 (et les versions antérieures) utilise la variable d'environnement OMNIHOME dans de nombreux fichiers de configuration. Pour utiliser ces fichiers sur Tivoli Netcool/OMNIbus V7.1 (et ultérieures), remplacez les occurrences de la variable d'environnement OMNIHOME par NCHOME/omnibus.

Sur les systèmes d'exploitation UNIX et Linux, remplacez \$OMNIHOME par \$NCHOME/omnibus.

Sur les systèmes d'exploitation Windows, remplacez %OMNIHOME% par %NCHOME%\omnibus.

Exécution de sondes sous UNIX

Sous UNIX, vous pouvez exécuter des sondes à partir de la ligne de commande, ou sous contrôle de processus.

Avant de commencer

Après avoir installé une sonde, configurez les fichiers de propriétés et de règles afin de l'adapter à votre environnement. Par exemple, si vous utilisez une sonde de fichier journal, comme la sonde HTTP Common Log Format, vous devez définir la propriété **LogFile** de sorte que la sonde puisse se connecter à la source d'événement.

Pourquoi et quand exécuter cette tâche

Important : Utilisez le contrôle de processus pour gérer les sondes. Pour obtenir des informations supplémentaires sur la configuration de sonde pour qu'elle s'exécute sous contrôle de processus, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Procédure

Pour exécuter une sonde, entrez la commande suivante :

```
$OMNIHOME/probes/nco_p_nom_sonde [-option [ valeur] ... ]
```

Dans cette commande, *nom_sonde* correspond au nom abrégé de la sonde que vous souhaitez exécuter. La variable *-option* est une option de ligne de commande et *valeur* correspond à la valeur sur laquelle vous définissez l'option. Toutes les options n'exigent pas que vous indiquiez une valeur.

Par exemple, pour exécuter la sonde Sybase en mode de capture brute, entrez :
\$NCHOME/omnibus/probes/nco_p_sybase -raw.

Les options de ligne de commande suivantes sont particulièrement utiles :

- Indiquez l'option de ligne de commande *-name* pour déterminer le nom utilisé pour les fichiers de sonde décrits dans le tableau suivant :

Tableau 27. Noms des fichiers de sonde

Type de fichier	Chemin d'accès et nom de fichier
Fichier de propriétés	\$NCHOME/omnibus/probes/arch/nco_p_nom_sonde.props
Fichier de règles	\$NCHOME/omnibus/probes/arch/nco_p_nom_sonde.rules

Tableau 27. Noms des fichiers de sonde (suite)

Type de fichier	Chemin d'accès et nom de fichier
Fichier de cache de règles	Lorsque la propriété CacheRules est définie sur 1, le fichier de règles est mise en cache dans le fichier spécifié par la propriété CacheRulesFile . La valeur par défaut est <code>\$NCHOME/omnibus/var/nom_sonde.rulescache</code> .
Fichier stocker-et-transmettre	<code>\$NCHOME/omnibus/var/nom_sonde.store.server</code>
Fichier journal des messages	<code>\$NCHOME/omnibus/log/nom_sonde.log</code>

Dans ces chemins d'accès, *arch* représente le nom du système d'exploitation sur lequel la sonde est installée ; par exemple, *solaris2* lors de l'exécution d'un système Solaris.

- Indiquez l'option de ligne de commande `-propsfile` pour que la valeur de l'option remplace le paramètre de nom du fichier de propriétés.
- Si vous avez besoin de réduire le nombre de connexions individuelles au serveur ObjectServer, vous pouvez connecter les sondes multiples via un serveur proxy. Utilisez l'option de ligne de commande `-server` et indiquez le nom du serveur proxy. Pour plus d'informations sur le serveur proxy, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Exécution de sondes sous SUID root

Vous pouvez exécuter des sondes en tant que SUID (SETUID) root pour utiliser des privilèges de superutilisateur avec un compte utilisateur de sonde non root. Cette configuration est normalement uniquement requise par la sonde SNMP Probe (**nco_p_mttrapd**) lorsqu'elle doit se connecter à un port d'hôte inférieur à 1024. Vous pouvez parfois également exécuter la sonde Ping Probe (**nco_p_ping**) en tant que SUID root.

Pourquoi et quand exécuter cette tâche

La procédure suivante montre comment configurer SUID root pour la sonde SNMP Probe. La sonde peut être exécutée en tant que SUID root sans compromettre la sécurité car elle abandonne ses privilèges de superutilisateur après l'ouverture de la session SNMP et avant le démarrage de la bibliothèque de sondes Netcool/OMNIbus.

Procédure

1. En tant que superutilisateur, exécutez la commande suivante à partir du répertoire `$NCHOME/omnibus/probes/arch` pour que le propriétaire du fichier binaire de la sonde soit root :


```
chown root nco_p_mttrapd
```

 Sur les systèmes 64 bits, exécutez la commande `chown` à partir du répertoire `$NCHOME/omnibus/platform/arch/probes64`.
2. En tant que superutilisateur, exécutez la commande suivante à partir du même répertoire pour permettre au fichier binaire de la sonde d'être exécuté par SUID root :

```
chmod +s nco_p_mttrapd
```

3. En fonction de votre système d'exploitation, utilisez la procédure suivante pour mettre les bibliothèques requises à la disposition de la sonde :

Linux

- a. En tant que superutilisateur, ouvrez `/etc/ld.so.conf` en édition et ajoutez les lignes suivantes à la fin du fichier :
- ```
$NCHOME/omnibus/platform/linux2x86/lib
$NCHOME/platform/linux2x86/lib
```
- b. En tant que superutilisateur, exécutez `ldconfig -v` pour créer les liens et le cache requis pour les bibliothèques partagées.

#### Solaris

- a. Exécutez la commande suivante pour enregistrer la bibliothèque Netcool/OMNIbus en tant que répertoire sécurisé :
- ```
crle -s /usr/lib/secure:/opt/netcool/omnibus/platform/arch/lib
```

AIX

- a. Copiez les bibliothèques Netcool/OMNIbus suivantes à partir du répertoire `$NCHOME/platform/aix5/lib` dans un répertoire de bibliothèque sécurisé, tel que `/usr/netcool/omnibus/platform/aix5/lib` :
- `libicudata40.1.a`
 - `libicui18n40.1.a`
 - `libicuuc40.1.a`
 - `libncrypt.so`
 - `libnetcool_r.so`
 - `libOpl_r.so`
 - `libPa_r.so`
 - `libslclient_r.so`
 - `libsybcomn_r.so`
 - `libsybcs_r.so`
 - `libsybct_r.so`
 - `libsybintl_r.so`
 - `libsybsrv_r.so`
 - `libsybtcl_r.so`
 - `libtre.a`
- b. Dans le répertoire de bibliothèque sécurisé, créez les liens symboliques suivants :
- `libicudata40.a -> libicudata40.1.a`
 - `libicui18n.a -> libicui18n40.1.a`
 - `libicui18n40.a -> libicui18n40.1.a`
 - `libicuuc.a -> libicudata40.1.a`
 - `libicuuc40.a -> libicuuc40.1.a`
 - `libnetcool_r.so.a -> libnetcool_r.so`
 - `libOpl_r.so.a -> libOpl_r.so`
 - `libPa_r.so.a -> libPa_r.so`
 - `libtre.so.4 -> libtre.a`

Résultats

Vous pouvez à présent exécuter la sonde à partir du répertoire \$NCHOME/omnibus/probes en tant que SUID root.

Pour plus d'informations sur la sonde SNMP, voir le guide de référence de la sonde.

Exécution de sondes sous Windows

Sous Windows, vous pouvez exécuter des sondes comme applications de console, en tant que services Windows ou sous contrôle de processus.

Pourquoi et quand exécuter cette tâche

Les sondes sont installées comme applications de console par défaut.

Pour obtenir des informations supplémentaires sur la configuration de sonde pour qu'elle s'exécute sous contrôle de processus, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Exécution d'une sonde en tant qu'application de console

Exécutez une sonde en tant qu'application de console à partir de la ligne de commande.

Pourquoi et quand exécuter cette tâche

Pour exécuter une sonde en tant qu'application de console, entrez la commande suivante à partir du répertoire de la sonde :

```
nco_p_nom_sonde [ -option [ valeur] ... ]
```

Dans cette commande, *nom_sonde* correspond au nom abrégé de la sonde que vous souhaitez exécuter. La variable *-option* est une option de ligne de commande et *valeur* correspond à la valeur sur laquelle vous définissez l'option. Toutes les options n'exigent pas que vous indiquiez une valeur.

D'autres options de ligne de commande sont disponibles pour la version Windows de chaque sonde. Pour les afficher, entrez la commande suivante :

```
nco_p_nom_sonde -?
```

Les options de ligne de commande spécifiques à Windows sont décrites dans le tableau suivant.

Tableau 28. Options de ligne de commande de sonde spécifiques à Windows

Option de ligne de commande	Description
-install	Cette option permet d'installer la sonde en tant que service Windows.

Tableau 28. Options de ligne de commande de sonde spécifiques à Windows (suite)

Option de ligne de commande	Description
-noauto	Cette option est utilisée avec l'option -install. Elle désactive le démarrage automatique pour que la sonde s'exécute en tant que service. Si cette option est utilisée, la sonde n'est pas démarrée automatiquement lors de l'amorçage de l'ordinateur.
-remove	Cette option permet de supprimer une sonde installée en tant que service. Il s'agit de l'élément opposé à la commande -install.
-group <i>chaîne</i>	Cette option est utilisée avec l'option de ligne de commande -depend. Vous pouvez regrouper toutes les sondes sous le même nom de groupe. Vous pouvez ensuite forcer ce groupe à dépendre d'un autre service.
-depend <i>srv @grp ...</i>	Cette option permet d'indiquer d'autres services ou groupes dont dépend la sonde. Si vous utilisez cette option, la sonde ne démarre pas avant que les services (<i>srv</i>) et les groupes (<i>@grp</i>) indiqués à l'aide de cette option n'aient été exécutés.
-cmdLine " <i>-option valeur...</i> "	Cette option permet d'indiquer une ou plusieurs options de ligne de commande à définir, lors de chaque redémarrage du service de la sonde.

Tâches associées:

«Exécution d'une sonde en tant que service»

Pour exécuter une sonde en tant que service, utilisez l'option de ligne de commande Windows /INSTALL lors de l'exécution de la sonde à l'aide de la commande `nco_p_nom_sonde`, où *nom_sonde* identifie la sonde de manière unique.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Exécution d'une sonde en tant que service

Pour exécuter une sonde en tant que service, utilisez l'option de ligne de commande Windows /INSTALL lors de l'exécution de la sonde à l'aide de la commande `nco_p_nom_sonde`, où *nom_sonde* identifie la sonde de manière unique.

Pourquoi et quand exécuter cette tâche

Après avoir configuré le service, vous pouvez régler le mode de démarrage de la sonde en définissant les paramètres des services Windows, comme suit :

Procédure

1. Cliquez sur **Démarrer** > **Panneau de configuration**. Le Panneau de configuration s'ouvre alors.
2. Cliquez deux fois sur l'icône **Outils d'administration**, puis sur l'icône **Services**. La fenêtre Services s'affiche alors.

La fenêtre Services répertorie tous les services Windows actuellement installés sur votre ordinateur. Tous les noms de services Tivoli Netcool/OMNIBus commencent par NCO.

3. Utilisez la fenêtre Services pour démarrer et arrêter les services Windows. Indiquez si le service doit être démarré automatiquement lorsque l'ordinateur est amorcé d'un simple clic sur le bouton **Démarrer**.

Remarque : Si le serveur ObjectServer et la sonde sont démarrés en tant que services, il se peut que cette dernière démarre en premier. La sonde ne peut pas se connecter au serveur ObjectServer avant que ce dernier ne s'exécute.

Résultats

Tâches associées:

«Exécution d'une sonde en tant qu'application de console», à la page 101
Exécutez une sonde en tant qu'application de console à partir de la ligne de commande.

Chapitre 5. Administration à distance des sondes

Vous pouvez utiliser une interface HTTP qui s'exécute sur un serveur HTTP ou HTTPS contenu dans la bibliothèque C de sondes standard (libOpl) pour surveiller et gérer à distance des sondes. Cette fonctionnalité peut être utilisée sur toutes les sondes qui sont compatibles avec Tivoli Netcool/OMNIBus version 8.1. Un identificateur URI commun est fourni. Il contient des ressources pour effectuer des tâches d'administration.

Pourquoi et quand exécuter cette tâche

Des utilitaires sont fournis pour envoyer des demandes HTTP à la sonde. Les sondes peuvent recevoir des demandes HTTP sur les ports HTTP et HTTPS. Les deux ports sont désactivés par défaut. Des tâches de configuration supplémentaires sont requises pour HTTPS.

Une sonde peut enregistrer son identificateur URI pour exposer des entités qui lui sont spécifiques. Ces identificateurs URI peuvent être utilisés pour déclencher des actions sur le périphérique ou le système auquel la sonde est connectée ou avec lequel elle communique. Pour plus d'informations, voir la documentation de la sonde concernée.

Activation de l'administration à distance des sondes

L'interface HTTP à la sonde est désactivée par défaut. Vous pouvez activer l'interface HTTP en définissant deux propriétés. Vous pouvez spécifier des valeurs pour les propriétés supplémentaires pour configurer l'interface HTTP (par exemple, le nombre d'unités d'exécution de tâche et le service de fichiers).

Pourquoi et quand exécuter cette tâche

Procédure

Pour activer et configurer l'administration à distance :

1. Pour activer l'administration à distance sans authentification ni transport HTTP, définissez les propriétés comme indiqué dans le tableau suivant :

Propriété	Instructions
<code>NHttpd.EnableHTTP</code>	Définissez cette propriété sur TRUE.
<code>NHttpd.ListeningPort</code>	Définissez cette propriété sur le nombre de ports libres.

Le port HTTP est activée et définie pour écouter les connexions HTTP sur le numéro de port indiqué. Le port est en mode écoute sur toutes les interfaces disponibles sur l'hôte.

2. Pour poursuivre la configuration de l'interface HTTP, définissez les propriétés suivantes selon vos besoins :

Propriété	Instructions
NHttpd.NumWorkThreads	Augmentez la valeur pour accroître le nombre d'unités d'exécution de tâche qui traitent les demandes HTTP entrantes. La valeur par défaut est 5.
NHttpd.ExpireTimeout	Définissez la durée d'inactivité maximale, en secondes, à l'issue de laquelle une connexion HTTP/1.1 est supprimée. La valeur par défaut est 15.
NHttpd.AccessLog	Définissez cette propriété sur le chemin du journal d'accès créé par la sonde. La valeur par défaut est <code>access.log</code> .
NHttpd.EnableFileServing	Définissez cette propriété sur TRUE pour activer le service de fichiers pris en charge par la sonde. Définissez cette propriété, avec la propriété NHttpd.DocumentRoot , pour activer la sonde pour agir en tant que serveur HTTP simple assurant le service de fichiers à partir du système de fichiers local. La valeur par défaut est FALSE.
NHttpd.DocumentRoot	Définissez cette propriété sur la racine de document pour les demandes HTTP. Définissez cette propriété, avec la propriété NHttpd.EnableFileServing , pour activer la sonde pour agir en tant que serveur HTTP simple assurant le service de fichiers à partir du système de fichiers local. La valeur par défaut est <code>./</code> .
NHttpd.AuthenticationDomain	Définissez cette propriété sur le domaine d'authentification qui s'affiche lorsque les détails d'authentification sont demandés au cours de la connexion HTTP ou HTTPS. La valeur par défaut est <code>omnibus</code> .

Que faire ensuite

Configurez l'authentification entre le système distant et l'interface HTTP et, si nécessaire, configurez une connexion SSL. Une fois que vous avez effectué ces tâches, redémarrez la sonde de sorte que toutes les modifications apportées aux valeurs de propriété puissent prendre effet.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Configuration de l'authentification entre les systèmes distants et les sondes

Vous pouvez configurer la fonctionnalité d'administration à distance pour authentifier les données d'identification de l'utilisateur pour chaque connexion à l'interface HTTP. L'authentification limite les connexions associées à une combinaison nom d'utilisateur-mot de passe valide. Par défaut, les données d'identification de l'utilisateur ne sont pas authentifiées. Par conséquent, elles ne sont pas autorisées.

Avant de commencer

Veillez à ce que le mot de passe associé au nom d'utilisateur soit haché avec AES-128. Utilisez l'utilitaire de ligne de commande **nco_crypt**, avec l'option **-aes** pour hacher le mot de passe. Par exemple :

```
nco_crypt -aes mot_de_passe
```

Pourquoi et quand exécuter cette tâche

L'authentification via l'interface HTTP est contrôlée par la propriété de sonde **Nhttpd.BasicAuth**. La valeur par défaut de cette propriété est "", qui considère comme valides n'importe quelles données d'identification d'utilisateur.

Une seule combinaison nom d'utilisateur-mot de passe est autorisée. Cette authentification ne communique pas avec d'autres méthodes d'authentification telles que l'authentification en mode sécurisé ou l'authentification PAM (Pluggable Authentication Modules).

La combinaison nom d'utilisateur-mot de passe est envoyée à l'interface HTTP sous une forme de texte quasiment en clair. Pour éviter le texte en clair, configurez l'interface HTTP pour les connexions SSL.

Procédure

Pour activer et configurer l'authentification :

Dans le fichier de propriétés de sonde, définissez la propriété **Nhttpd.BasicAuth** sur une chaîne au format "*nom_utilisateur*:*mot_de_passe*", où *nom_utilisateur* désigne toute combinaison de caractères qui n'inclut pas le signe deux-points (:) et *mot_de_passe* représente le hachage AES-128 du mot de passe généré par l'utilitaire **nco_crypt**.

Par exemple :

- La valeur suivante est une combinaison du nom d'utilisateur `auth` et du hachage du mot de passe `netcool` :

```
"auth:t0ufpMhIN7R3L1Np89XRvA=="
```

Si vous indiquez une chaîne vide ("") pour la propriété, toutes les tentatives d'authentification sont autorisées, quelles que soient les données d'identification fournies.

Que faire ensuite

Pour éviter d'envoyer les mots de passe en clair, configurez une connexion SSL entre le système distant et la sonde. Pour augmenter la sécurité de la combinaison utilisateur-mot de passe, changez les autorisations de système de fichiers associées

au fichier de propriétés de sonde. Une fois que vous avez effectué ces tâches, redémarrez la sonde de sorte que toutes les modifications apportées aux valeurs de propriété puissent prendre effet.

Tâches associées:

«Configuration des connexions SSL entre les systèmes distants et les sondes»
Pour sécuriser les communications et le trafic HTTP à destination et en provenance des sondes distantes, configurez une connexion SSL. Cela évite d'envoyer les communications en texte en clair.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129
Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Configuration des connexions SSL entre les systèmes distants et les sondes

Pour sécuriser les communications et le trafic HTTP à destination et en provenance des sondes distantes, configurez une connexion SSL. Cela évite d'envoyer les communications en texte en clair.

Avant de commencer

Pour la connexion SSL, assurez-vous que l'authentification est activée. Vérifiez également que la sonde est configurée pour être en mode écoute sur un port HTTPS. Par exemple, créez une base de données de clés sur l'hôte sur lequel la sonde est en cours d'exécution, puis créez un certificat autosigné (CA). Ensuite, créez une demande de certificat à partir de l'hôte de la sonde et signez avec le label de certificat CA. Recevez le certificat signé dans la base de données de clés sur l'hôte de la sonde, de sorte que cette base de données de clés contienne un certificat pour l'hôte de la sonde. Ensuite, extrayez le certificat de signataire. Ajoutez-le à la base de données de clés, sur l'hôte à partir duquel vous souhaitez vous connecter à l'interface HTTP, puis émettez les demandes.

Procédure

Pour activer et configurer des connexions SSL, définissez les propriétés suivantes dans le fichier de règles de sonde :

Propriété	Instructions
NHttpd.SSLEnable	Définissez cette propriété sur TRUE.
NHttpd.SSLListeningPort	Définissez cette propriété sur le numéro de port sur lequel la sonde doit être en mode écoute pour les connexions SSL.
NHttpd.SSLCertificate	Définissez cette propriété sur le label de certificat contenu dans le fichier de base de données de clés \$NCHOME/etc/security/keys/omni.kdb. Cette valeur est utilisée en tant que certificat pour le serveur HTTPS.
NHttpd.SSLCertificatePwd	Définissez cette propriété sur le mot de passe pour le fichier de base de données de clés \$NCHOME/etc/security/keys/omni.kdb.

Que faire ensuite

Redémarrez la sonde de sorte que toutes les modifications apportées aux valeurs de propriété puissent prendre effet.

Tâches associées:

«Configuration de l'authentification entre les systèmes distants et les sondes», à la page 107

Vous pouvez configurer la fonctionnalité d'administration à distance pour authentifier les données d'identification de l'utilisateur pour chaque connexion à l'interface HTTP. L'authentification limite les connexions associées à une combinaison nom d'utilisateur-mot de passe valide. Par défaut, les données d'identification de l'utilisateur ne sont pas authentifiées. Par conséquent, elles ne sont pas autorisées.

«Envoi de demandes à distance à des sondes (nco_http)»

Utilisez l'utilitaire **nco_http** pour vous connecter à l'interface HTTP ou HTTPS d'une sonde et envoyer des demandes.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Envoi de demandes à distance à des sondes (nco_http)

Utilisez l'utilitaire **nco_http** pour vous connecter à l'interface HTTP ou HTTPS d'une sonde et envoyer des demandes.

Avant de commencer

- Configurez l'authentification. Voir «Configuration de l'authentification entre les systèmes distants et les sondes», à la page 107.
- Si nécessaire, configurez une connexion SSL. Voir «Configuration des connexions SSL entre les systèmes distants et les sondes», à la page 108.

Procédure

Pour démarrer l'utilitaire **nco_http**, exécutez la commande suivante :

- **UNIX** **Linux** `$OMNIHOME/bin/nco_http options_ligne_commande`
- **Windows** `%OMNIHOME%\bin\nco_http.exe options_ligne_commande`

où les *options_ligne_commande* sont décrites dans tableau 29, à la page 110. Vous pouvez utiliser le fichier de propriétés `nco_http.props` pour stocker plusieurs valeurs d'option de ligne de commande. Vous pouvez stocker les options fréquemment utilisées, telles que le nom d'utilisateur et le mot de passe, plutôt que de les entrer chaque fois que vous exécutez l'utilitaire.

Les propriétés et les options de ligne de commande de **nco_http** sont décrites dans le tableau suivant.

Tableau 29. Propriétés et options de ligne de commande *nco_http*

Propriété	Option de ligne de commande	Description
ConfigCryptoAlg chaîne	N/D	<p>Indique l'algorithme de cryptographie à utiliser pour le déchiffrement des valeurs de chaîne (dont les mots de passe) chiffrées à l'aide de l'utilitaire nco_aes_crypt et stockées ensuite dans le fichier de propriétés. Définissez la valeur chaîne comme suit :</p> <ul style="list-style-type: none"> • En mode FIPS 140-2, utilisez AES_FIPS. • En mode FIPS 140-2 désactivé, vous pouvez utiliser AES_FIPS ou AES. Utilisez uniquement AES si vous devez conserver une compatibilité avec des mots de passe chiffrés à l'aide des outils fournis dans des versions antérieures à Tivoli Netcool/OMNIBus version 7.2.1. <p>La valeur que vous indiquez doit être identique à celle utilisée lors de l'exécution de nco_aes_crypt avec le paramètre -c pour chiffrer les valeurs de chaîne.</p> <p>Utilisez cette propriété avec la propriété ConfigKeyFile.</p> <p>La valeur par défaut est AES.</p>
ConfigKeyFile chaîne	N/D	<p>Indique le chemin et le nom du fichier de clés contenant la clé utilisée pour déchiffrer les valeurs de chaîne chiffrées (dont les mots de passe) dans le fichier de propriétés.</p> <p>La clé est utilisée au moment de l'exécution pour déchiffrer les valeurs de chaîne chiffrées à l'aide de l'utilitaire nco_aes_crypt. Le fichier de clés que vous avez spécifié doit être identique au fichier utilisé pour chiffrer les valeurs de chaîne lors de l'exécution de nco_aes_crypt avec le paramètre -k.</p> <p>Utilisez cette propriété avec la propriété ConfigCryptoAlg.</p> <p>La valeur par défaut est "".</p>
Data chaîne	-data chaîne	<p>Indique les données HTTP pour la demande.</p> <p>La valeur par défaut est "".</p>
DataType chaîne	-datatype chaîne	<p>Indique le type MIME des données HTTP pour la demande.</p> <p>La valeur par défaut est "".</p>
Header chaîne	-header chaîne	<p>Indique les données d'en-tête HTTP pour la demande.</p> <p>La valeur par défaut est "".</p>

Tableau 29. Propriétés et options de ligne de commande *nco_http* (suite)

Propriété	Option de ligne de commande	Description
N/D	-help	Affiche l'aide sur les options de ligne de commande.
N/D	-messagelevel	Indique le niveau de consignation des messages. Les niveaux de consignation disponibles sont FATAL, ERROR, WARN, INFO et DEBUG. Le niveau par défaut est INFO.
MessageLog chaîne	-messagelevel chaîne	Indique le nom et l'emplacement du fichier journal. Le niveau par défaut est stderr (erreur standard).
Method chaîne	-method chaîne	Indique la méthode HTTP pour la demande. Les méthodes disponibles sont GET, POST et PATCH. La méthode par défaut est GET.
NHtpd.SSLCertificatePwd chaîne	N/D	Indique le mot de passe requis pour l'accès au fichier de certificats SSL. La valeur par défaut est "".
NHtpd.SSLEnable booléen	N/D	Permet l'utilisation du support SSL. La valeur par défaut est FALSE.
Password chaîne	-password chaîne	Indique le mot de passe HTTP pour la demande. La valeur par défaut est "".
N/D	-propsfile	Indique le nom et l'emplacement du fichier de propriétés de l'utilitaire. La valeur par défaut est \$OMNIHOME/etc/nco_http.props
N/D	-timeout	Indique le délai d'attente pour la demande. La valeur par défaut est 60 secondes.
URI chaîne	-uri chaîne	Indique l'identificateur URI HTTP pour la demande. La valeur par défaut est "".
Username chaîne	-username chaîne	Indique le nom d'utilisateur HTTP pour la demande. La valeur par défaut est "".
N/D	-version	Affiche les informations de version de l'utilitaire. La valeur par défaut est "".

Exemple

L'exemple suivant renvoie les informations de version actuelle d'une sonde, la valeur de la zone reloadrulesflag de la sonde et les valeurs des propriétés de la sonde. La demande renvoie les propriétés chiffrées sous leur forme chiffrée.

```
$OMNIHOME/bin/nco_http -uri http://servername.ibm.com:numéro_port/probe/common
```

Les informations renvoyées se présentent comme suit:

```
2012-06-22T13:56:31: Information: I-UNK-104-002: {
  "version": {
    "ProbeVersion": "Netcool/OMNIBus Event Simulation Probe probe - Version 7.4.0
                    64-bit\n(C) Copyright IBM Corp. 1994, 2012\n",
    "APIVersion": "Netcool/OMNIBus Probe API Library Version 7.4.0 64-bit",
    "ReleaseID": "5.2012.0620",
    "APIReleaseID": "5.2012.0620",
    "SoftwareCompileInfo": "Wed Jun 20 20:17:13 BST 2012 on servername.ibm.com
                           (Linux 2.6.18-274.17.1.e15 #1 SMP Wed Jan 4 22:45:44 EST 2012)"
  },
  "reloadrulesflag": 0,
  "properties": {
    "NHttpd.SSLCertificate": "",
    "RawCaptureFileAppend": 0,
    "StoreAndForward": 1,
    "KeepLastBrokenSAF": 0,
    "MessageLevel": "warn",
    "ConfigCryptoAlg": "AES",
    "StoreSAFRejects": 0,
    "Op1StatsNumberEventsGenerated": 0,
    "Buffering": 0,
    "RawCapture": 0,
    "Op1DetailsTableName": "alerts.details",
    "SSLServerCommonName": "",
    "LookupTableMode": 3,
    "NHttpd.EnableHTTP": true,
    "NHttpd.AccessLog": "access.log",
    "Op1StatsNumberEventsDiscarded": 0,
    "PollServer": 0,
    "ProbeWatchHeartbeatInterval": 0,
    "RetryConnectionTimeout": 30,
    "Op1StatsRulesFileTimeSec": 0,
    "MessageLog": "/export/views/dev/omnibus74/omnibus/log/simnet.log",
    "NHttpd.AuthenticationDomain": "omnibus",
    "NHttpd.SSLListeningPort": 0,
    "SAFFileName": "/export/views/dev/omnibus74/omnibus/var/simnet.store",
    "LogFile": "/export/views/dev/omnibus74/omnibus/probes/linux2x86/simnet.def",
    "MaxLogFileSize": 1048576,
    "MsgDailyLog": 0,
    "BeatInterval": 2,
    "Server": "NCOMS",
    "CacheRulesFile": "/export/views/dev/omnibus74/omnibus/var/simnet.rulescache",
    "Props.CheckNames": true,
    "BeatThreshold": 1,
    "CacheRules": 0,
    "Op1StatsMemoryInUse": 359952,
    "BufferSize": 10,
    "Help": 0,
    "LogFilePoolSize": 10,
    "PidFile": "/export/views/dev/omnibus74/omnibus/var/simnet",
    "Manager": "Omnibus",
    "NHttpd.ExpireTimeout": 15,
    "NHttpd.SSLCertificatePwd": "",
    "Peerport": 9999,
    "RulesFile": "/export/views/dev/omnibus74/omnibus/probes/linux2x86/simnet.rules",
    "OldTimeStamp": "FALSE",
    "SingleThreadedComms": 0,
  },
}
```

```

"Op1StatusTableName": "alerts.status",
"ServerBackup": "",
"Op1StatsProbeStartTime": 1340369540,
"TimeBetweenEvents": 1000,
"LogFileUsePool": 0,
"MaxRawFileSize": -1,
"Op1PaName": "",
"Op1StatsCPUTimeSec": 0,
"NetworkTimeout": 0,
"PeerHost": "localhost",
"NHttpd.NumWorkThreads": 5,
"NHttpd.DocumentRoot": "./",
"LogFileUseStdErr": 0,
"RegexpLibrary": "TRE",
"AuthPassword": "",
"NHttpd.EnableFileServing": false,
"Op1StatsNumberEvents": 251,
"AuthUserName": "",
"SecureLogin": 0,
"Version": 0,
"Mode": "standard",
"RetryConnectionCount": 15,
"MsgTimeLog": "0000",
"Op1StatsCPUTimeUsec": 50000,
"NHttpd.SSLEnable": false,
"Op1DumpProps": 0,
"PropsFile": "/export/views/dev/omnibus74/omnibus/probes/linux2x86/simnet.props",
"Op1StatsRulesFileTimeUsec": 7675,
"MaxSAFFileSize": 1048576,
"ConfigKeyFile": "",
"RawCaptureFile": "/export/views/dev/omnibus74/omnibus/var/simnet.cap",
"Name": "simnet",
"Op1PaID": 0,
"RollSAFInterval": 90,
"NHttpd.ListeningPort": 4444,
"SAFPoolSize": 3,
"NHttpd.BasicAuth": "",
"Op1PacketSize": 512,
"AutoSAF": 0
}
}
}

```

L'exemple suivant définit la propriété **MessageLevel** de la sonde pour le débogage :

```

$OMNIHOME/bin/nco_http -uri http://servername.ibm.com:4444/probe/common
-datatype application/json -data '{ "properties": { "MessageLevel": "debug" } }'
-method patch

```

Fix Pack 1 Dans l'exemple qui suit, la propriété **DisableDetails** est définie pour désactiver les détails d'alerte, de sorte qu'aucune instruction INSERT n'est écrite dans la table alert.details chaque fois qu'une alerte est envoyée.

```

$OMNIHOME/bin/nco_http -uri http://servername.ibm.com:4444/probe/common
-datatype application/json -data '{ "properties": { "DisableDetails": 1 } }'
-method patch

```

Tâches associées:

«Configuration des connexions SSL entre les systèmes distants et les sondes», à la page 108

Pour sécuriser les communications et le trafic HTTP à destination et en provenance des sondes distantes, configurez une connexion SSL. Cela évite d'envoyer les communications en texte en clair.

Référence associée:

«A propos de l'identificateur URI commun», à la page 117
Utilisez l'identificateur URI commune fourni dans la bibliothèque C de sondes standard (libOpl) pour exécuter des commandes HTTP standard sur toutes les sondes.

Rechargement de fichiers de règles (nco_probereloadrules)

Vous pouvez utiliser l'utilitaire **nco_probereloadrules** pour recharger à distance un fichier de règles de sonde sans avoir à redémarrer la sonde.

Avant de commencer

Vérifiez que l'authentification et, si nécessaire, une connexion SSL sont configurées. La sonde avec laquelle vous souhaitez communiquer doit également avoir son interface HTTP ou HTTPS activée.

Pourquoi et quand exécuter cette tâche

Pour démarrer l'utilitaire **nco_probereloadrules**, utilisez la commande suivante :

```
UNIX $OMNIHOME/bin/nco_probereloadrules options_ligne_commande
```

```
Windows %OMNIHOME%\bin\nco_probereloadrules.cmd command_line_options
```

L'utilitaire **nco_probereloadrules** utilise l'utilitaire **nco_http** pour recharger le fichier de règles sur une connexion HTTP ou HTTPS. Il peut également utiliser le fichier de propriétés **nco_http.props**. Vous pouvez stocker les options fréquemment utilisées, telles que le nom d'utilisateur et le mot de passe, dans **nco_http.props** plutôt que de les entrer chaque fois que vous exécutez l'utilitaire.

Les options de ligne de commande de l'utilitaire **nco_probereloadrules** sont décrites dans le tableau suivant.

Tableau 30. Options de ligne de commande nco_probereloadrules

Option de ligne de commande	Description
-help	Affiche l'aide sur les options de ligne de commande.
-host	Indique le nom de l'hôte sur lequel la sonde est installée.
-messagelevel	Indique le niveau de consignation des messages. Les niveaux de consignation disponibles sont FATAL, ERROR, WARN, INFO et DEBUG. Le niveau par défaut est INFO.
-password	Indique le mot de passe HTTP pour la mise à jour.
-port	Indique le numéro de port sur lequel communiquer avec la sonde.
-ssl	Indique l'utilisation d'une connexion HTTPS.
-timeout	Indique le délai d'attente de la réponse HTTP.
-username	Indique le nom d'utilisateur HTTP pour la demande.

Exemple

L'exemple suivant recharge le fichier de règles associé à une sonde dont l'hôte est **servername** et où le port est **2020**.

```
$OMNIHOME/bin/nco_probereloadrules -host servername -port 2020
```

Tâches associées:

«Relecture du fichier de règles», à la page 66

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Envoi de mises à jour de propriété à des sondes (nco_setprobeprop)

Vous pouvez utiliser l'utilitaire **nco_setprobeprop** pour mettre à jour la valeur d'une propriété de sonde. L'utilitaire **nco_setprobeprop** utilise l'utilitaire **nco_http** pour effectuer la mise à jour de propriété sur une connexion HTTP ou HTTPS. **nco_setprobeprop** est également utilisé par les déclencheurs de contrôle des débordements d'événements pour faire passer des sondes en mode de contrôle des débordements d'événements.

Avant de commencer

Vérifiez que l'authentification et, si nécessaire, une connexion SSL sont configurées. La sonde avec laquelle vous souhaitez communiquer doit également avoir son interface HTTP ou HTTPS activée.

Pourquoi et quand exécuter cette tâche

Pour démarrer l'utilitaire **nco_setprobeprop**, utilisez la commande suivante :

```
UNIX $OMNIHOME/bin/nco_setprobeprop options_ligne_commande
```

```
Windows %OMNIHOME%\bin\nco_setprobeprop.cmd options_ligne_commande
```

Si vous indiquez une nouvelle valeur pour une propriété de sonde existante, l'utilitaire met à jour la valeur de la propriété. Si vous n'indiquez aucune valeur, la nouvelle valeur de la propriété est une chaîne vide.

Vous pouvez également indiquer un nom de propriété qui n'existe pas. Dans ce cas, l'utilitaire crée la propriété et la définit sur la valeur que vous précisez. Dans ce cas, vous devez avoir modifié le fichier de règles de la sonde pour gérer la propriété (par exemple, utilisez *%nom_propriété* pour obtenir la valeur de la propriété). Les propriétés créées par l'utilitaire sont transitoires et sont donc perdues lorsque la sonde s'arrête.

Les options de ligne de commande de l'utilitaire **nco_setprobeprop** sont décrites dans le tableau suivant. Etant donné que **nco_setprobeprop** emploie l'utilitaire **nco_http** pour communiquer avec la sonde, il peut utiliser le fichier de propriétés `nco_http.props`. Vous pouvez stocker les options fréquemment utilisées, telles que le nom d'utilisateur et le mot de passe, dans `nco_http.props` plutôt que de les entrer chaque fois que vous exécutez l'utilitaire.

Tableau 31. Options de ligne de commande `nco_setprobeprop`

Option de ligne de commande	Description
-help	Affiche l'aide sur les options de ligne de commande.
-host	Indique le nom de l'hôte sur lequel la sonde est installée.

Tableau 31. Options de ligne de commande `nco_setprobeprop` (suite)

Option de ligne de commande	Description
<code>-messagelevel</code>	Indique le niveau de consignation des messages. Les niveaux de consignation disponibles sont FATAL, ERROR, WARN, INFO et DEBUG. Le niveau par défaut est INFO.
<code>-name</code>	Indique le nom de la propriété à mettre à jour ou à créer.
<code>-password</code>	Indique le mot de passe HTTP pour la mise à jour.
<code>-port</code>	Indique le numéro de port sur lequel communiquer avec la sonde.
<code>-ssl</code>	Indique l'utilisation d'une connexion HTTPS.
<code>-timeout</code>	Indique le délai d'attente de la réponse HTTP.
<code>-username</code>	Indique le nom d'utilisateur HTTP pour la demande.
<code>-value</code>	Indique la nouvelle valeur de la propriété. Si vous n'indiquez aucune valeur, la nouvelle valeur de la propriété est une chaîne vide.

Exemple

L'exemple suivant définit la valeur de la propriété **FloodControl** sur `flood`. L'hôte de la sonde est `servername` et le port est `2000`.

```
$OMNIHOME/bin/nco_setprobeprop -name FloodControl -value flood -host servername
-port 2000
```

Génération d'événements à l'aide de sondes (`nco_probeeventfactory`)

Vous pouvez utiliser l'utilitaire **`nco_probeeventfactory`** à distance pour générer un événement avec une sonde. L'utilitaire force la sonde à traiter le fichier de règles à l'aide des paires nom-valeur que vous indiquez à la place des valeurs fournies par la source d'événement habituelle. Cette méthode est particulièrement utile pour tester les modifications apportées au fichier de règles. Vous pouvez également modifier le fichier de règles afin que les paires nom/valeur que vous fournissez provoquent le changement de variables internes, sans avoir besoin d'un événement «réel» dans une source d'événement.

Avant de commencer

Vérifiez que l'authentification et, si nécessaire, une connexion SSL sont configurées. La sonde avec laquelle vous souhaitez communiquer doit également avoir son interface HTTP ou HTTPS activée.

Pourquoi et quand exécuter cette tâche

Pour démarrer l'utilitaire **`nco_probeeventfactory`**, utilisez la commande suivante :

```
UNIX $OMNIHOME/bin/nco_probeeventfactory options_ligne_commande
[nom=valeur]...
```

```
Windows %OMNIHOME%\bin\nco_probeeventfactory.cmd options_ligne_commande
[nom=valeur]...
```

Vous pouvez spécifier les éléments d'événement en tant que paires nom-valeur. Vous devez placer les éléments ou les valeurs qui contiennent des espaces entre guillemets doubles (""). Par exemple :

```
Summary="Voici le récapitulatif."
```

Vous pouvez spécifier autant de paires nom-valeur que votre limite de caractères de ligne de commande le permet.

L'utilitaire **nco_probeeventfactory** utilise l'utilitaire **nco_http** pour générer l'événement sur une connexion HTTP ou HTTPS. Il peut également utiliser le fichier de propriétés `nco_http.props`. Vous pouvez stocker les options fréquemment utilisées, telles que le nom d'utilisateur et le mot de passe, dans `nco_http.props` plutôt que de les entrer chaque fois que vous exécutez l'utilitaire.

Les options de ligne de commande de l'utilitaire **nco_probeeventfactory** sont décrites dans le tableau suivant.

Tableau 32. Options de ligne de commande `nco_probeeventfactory`

Option de ligne de commande	Description
<code>-help</code>	Affiche l'aide sur les options de ligne de commande.
<code>-host</code>	Indique le nom de l'hôte sur lequel la sonde est installée.
<code>-messagelevel</code>	Indique le niveau de consignation des messages. Les niveaux de consignation disponibles sont FATAL, ERROR, WARN, INFO et DEBUG. Le niveau par défaut est INFO.
<code>-password</code>	Indique le mot de passe HTTP pour la mise à jour.
<code>-port</code>	Indique le numéro de port sur lequel communiquer avec la sonde.
<code>-ssl</code>	Indique l'utilisation d'une connexion HTTPS.
<code>-timeout</code>	Indique le délai d'attente de la réponse HTTP.
<code>-username</code>	Indique le nom d'utilisateur HTTP pour la demande.

Exemple

L'exemple suivant génère un événement avec une sonde dont l'hôte est `servername` et où le port est `2020`. Les paires nom-valeur indiquent que le groupe d'alertes est défini sur `Netcool`, le nom d'hôte du périphérique est défini sur `testserver.ibm.com`, et la gravité de l'alerte est défini sur `1`.

```
$OMNIHOME/bin/nco_probeeventfactory -host servername -port 2020 AlertGroup=Netcool Host=testserver.ibm.com Severity=1
```

A propos de l'identificateur URI commun

Utilisez l'identificateur URI commune fourni dans la bibliothèque C de sondes standard (`libOpl`) pour exécuter des commandes HTTP standard sur toutes les sondes.

Vous pouvez utiliser l'identificateur URI pour effectuer les actions suivantes sur les sondes :

- Afficher un rapport sur la version et l'identificateur de la sonde ainsi que des informations sur l'état en cours de la sonde, y compris ses propriétés
- Indiquer à la sonde de recharger le fichier de règles associé
- Lister les propriétés de la sonde

- Obtenir ou définir une propriété
- Créer des événements synthétiques traités par le fichier de règles.

Une sonde peut enregistrer son identificateur URI pour exposer des entités qui lui sont spécifiques. Ces identificateurs URI peuvent être utilisés pour déclencher des actions sur le périphérique ou le système auquel la sonde est connectée ou avec lequel elle communique.

Le tableau suivant décrit les fonctions de l'identificateur URI.

Fonction	Description
Chemin	/probe/common
Méthodes	GET, POST, PATCH
Types MIME acceptés	application/json, si applicable
Content-type	application/json, si applicable
Mise en cache	Une demande GET définit la zone d'en-tête HTTP Cache-Control sur no-cache, pour indiquer le caractère volatile des données renvoyées par une demande GET de l'identificateur URI.
Codes retour HTTP	Une liste de codes retour HTTP accompagnés de leur signification (en relation avec l'action) est fournie. Cette liste n'est pas exhaustive et tout code retour HTTP peut être renvoyé. Construisez des applications ou des scripts pouvant attendre tout code retour possible.

Pour plus d'informations sur JSON (JavaScript Object Notification), voir <http://www.json.org/>.

Extraire l'état en cours d'une sonde

Cette demande accède aux informations de version actuelle d'une sonde, aux valeurs des propriétés de sonde et à la valeur de la zone reloadrulesflag de la sonde. La demande renvoie les propriétés chiffrées sous leur forme chiffrée.

Vérifiez que l'application ou le script que vous concevez pour changer des valeurs extrait un ensemble récent de valeurs via une demande GET préalable à la demande PATCH.

Le tableau suivant décrit les fonctions de cette demande.

Fonction	Description
URI	/probe/common
Méthode	GET
Content-type	application/json
Paramètres de requête	search Pour filtrer les propriétés renvoyées par une demande GET.

Fonction	Description
Mise en cache	La réponse à la demande est marquée no-cache en raison de la volatilité des données contenues dans la réponse. Les éléments HTTP ETag ne sont pas pris en charge.

Exemple de réponse

Voici un exemple de réponse à la demande. La liste des propriétés renvoyées est abrégée.

```
{
  "version": {
    "ProbeVersion": "Netcool/OMNIBus Event Simulation Probe probe - Version 7.4\n
      (C) Copyright IBM Corp. 1994, 2007\n",
    "APIVersion": "Netcool/OMNIBus Probe API Library Version 7.4",
    "ReleaseID": "3.0.4082",
    "APIReleaseID": "5.2012.0410",
    "SoftwareCompileInfo": "Tue Jan 29 13:40:19 GMT 2008 on sol8-build2.hursley.ibm.com
      (SunOS 5.8 Generic_117350-33)"
  },
  "reloadrulesflag": 0,
  "properties": {
    "NHttpd.SSLCertificate": "",
    "RawCaptureFileAppend": 0,
    "StoreAndForward": 1,
    ...
    "Op1PacketSize": 512,
    "AutoSAF": 0
  }
}
```

Codes de réponse HTTP

Le tableau suivant présente les codes de réponse pouvant être renvoyés par la demande. Ce tableau n'est pas exhaustif. Les clients doivent être préparés pour tous les codes d'état HTTP.

Code de réponse	Explication
200	La demande a abouti.

Recharger le fichier de règles

Cette demande accède à l'attribut reloadrulesflag de la sonde, contenu dans la bibliothèque C de sondes standard (libOpl). En définissant la valeur de cet attribut, la demande force la sonde à recharger son fichier de règles avant le traitement de l'événement suivant. Un signal SIGHUP sur UNIX déclenche la même action.

L'attribut reloadrulesflag peut être défini sur 1 pour forcer la sonde à recharger son fichier de règles.

La demande est actionnée juste avant le traitement de l'événement suivant. Cela signifie qu'un retard peut se produire entre le moment où la demande est formulée et celui où elle est actionnée (par exemple, en cas d'inactivité de la sonde).

Le tableau suivant décrit les fonctions de cette demande.

Fonction	Description
URI	probe/common
Méthode	PATCH
Types MIME acceptés	application/json
Paramètres de requête	Aucun

Exemple de demande

Voici un exemple de demande forçant une sonde à recharger son fichier de règles.

```
{ "reloadrulesflag" : 1 }
```

Codes de réponse HTTP

Le tableau suivant présente les codes de réponse pouvant être renvoyés par la demande. Ce tableau n'est pas exhaustif. Les clients doivent être préparés pour tous les codes d'état HTTP.

Code de réponse	Explication
200 OK	La demande a abouti.
400 Bad Request	Par exemple, lorsqu'une demande PATCH est effectuée avec des données utiles non valides.
415 Unsupported Media Type	Les données ont été fournies en entrée dans un autre format que application/json.

Tâches associées:

«Relecture du fichier de règles», à la page 66

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Référence associée:

«Définir les demandes PATCH ou POST comme bloquantes ou non bloquantes», à la page 127

Les demandes qui changent les valeurs ou les actions des déclencheurs avec PATCH ou POST peuvent être définies comme bloquantes ou non bloquantes.

Répertoirer les propriétés de sonde

Cette demande accède aux propriétés de la sonde. Vous pouvez indiquer une variable de requête de recherche pour restreindre les propriétés renvoyées.

Restriction : La demande ne renvoie pas les propriétés qui sont chiffrées dans le fichier de propriétés.

Le tableau suivant décrit les fonctions de cette demande.

Fonction	Description
URI	/probe/common
Méthode	GET
Content-type	application/json
Paramètre de recherche	search La recherche est une recherche de sous-chaîne insensible à la casse. Une chaîne de recherche vide est équivalente à aucune chaîne de recherche et renvoie toutes les propriétés. La variable de requête de recherche n'empêche le renvoi des informations de version et de la valeur de l'attribut reloadrulesflag.

Exemple

Voici un exemple de réponse à la demande dans lequel la variable de requête de recherche «buff» est spécifiée (GET /probe/common?search=buff). Dans cet exemple, les informations de version qui sont normalement renvoyées sont omises.

```
{
  "version": {
    ...
  },
  "reloadrulesflag": 0,
  "properties": {
    "Buffering": 0,
    "BufferSize": 10
  }
}
```

Codes de réponse HTTP

Le tableau suivant présente les codes de réponse pouvant être renvoyés par la demande. Ce tableau n'est pas exhaustif. Les clients doivent être préparés pour tous les codes d'état HTTP.

Code de réponse	Explication
200 OK	La demande a abouti

Créer un événement synthétique

Cette demande crée un événement synthétique pouvant déclencher l'exécution du fichier de règles de sonde ou d'un fragment de ce fichier de règles.

Le tableau suivant décrit les fonctions de cette demande.

Fonction	Description
URI	/probe/common
Méthode	POST
Types MIME acceptés	application/json

Exemples

L'exemple de demande suivant crée un événement pour parcourir le fichier de règles, avec les jetons suivants définis (dans la syntaxe du fichier de règles) :

- \$token1="value1"
- \$token2="value2"
- \$token3="value3"

```
{ "eventfactory" :  
  [  
    { "token1" : "value1", "token2" : "value2", "token3" : "value3" }  
  ]  
}
```

L'exemple de demande suivant crée un événement pour parcourir le fichier de règles, avec les jetons suivants définis (dans la syntaxe du fichier de règles) :

- Pour le premier événement :
 - \$token1="value1"
 - \$token2="value2"
 - \$token3="value3"
- Pour le second événement :
 - \$a1="v1"
 - \$a2="v2"

```
{ "eventfactory" :  
  [  
    { "token1" : "value1", "token2" : "value2", "token3" : "value3" },  
    { "a1" : "v1", "a2" : "v2" }  
  ]  
}
```

Codes de réponse HTTP

Le tableau suivant présente les codes de réponse pouvant être renvoyés par la demande. Ce tableau n'est pas exhaustif. Les clients doivent être préparés pour tous les codes d'état HTTP.

Code de réponse	Explication
200 OK	La demande a abouti.
400 Bad Request	Par exemple, lorsqu'une demande PATCH est effectuée avec des données utiles non valides.
415 Unsupported Media Type	Les données ont été fournies en entrée dans un autre format que application/json.
500 Internal Server Error	Une erreur interne a empêché l'exécution de l'action. Pour résoudre les problèmes liés à la demande, effectuez une demande GET pour vérifier si des propriétés requises ont été modifiées. Si tel n'est pas le cas, soumettez à nouveau la demande.

Référence associée:

«Définir les demandes PATCH ou POST comme bloquantes ou non bloquantes», à la page 127

Les demandes qui changent les valeurs ou les actions des déclencheurs avec PATCH ou POST peuvent être définies comme bloquantes ou non bloquantes.

Définir une propriété de sonde

Cette demande change les valeurs de propriétés de sonde. Elle peut également créer de nouvelles propriétés. Toutes les nouvelles propriétés créées de cette manière nécessitent une valeur de chaîne.

Important : Bien que cette demande puisse changer ou définir les valeurs de toutes les propriétés de sonde, le comportement de la sonde peut rester inchangé. La plupart des propriétés de sonde sont en lecture seule lorsque la sonde démarre et ne sont pas relues par la suite. Le redémarrage de la sonde peut être nécessaire pour que son comportement reflète le changement de valeur de propriété. Si vous travaillez avec une paire de reprise en ligne, vérifiez que les modifications que vous apportez à la sonde maître sont également propagées à l'esclave.

Le tableau suivant décrit les fonctions de cette demande.

Fonction	Description
URI	probe/common
Méthode	PATCH
Types MIME acceptés	application/json
Paramètres de requête	Aucun

Exemples

Voici un exemple de demande ayant pour résultat de définir la propriété **RawCapture** sur 1 :

```
{
  "properties": {
    "RawCapture": 1
  }
}
```

L'exemple suivant montre une demande non valide car la propriété **RawCapture** nécessite un entier, et non une chaîne. La demande entraîne la réponse 400 Bad Request.

```
{
  "properties": {
    "RawCapture": "yes"
  }
}
```

Voici un exemple de demande créant deux nouvelles propriétés : **MyProperty** avec la valeur busy et **AnotherProp** avec la valeur no.

```
{
  "properties": {
    "MyProperty": "busy",
    "AnotherProp": "no"
  }
}
```

Codes de réponse HTTP

Le tableau suivant présente les codes de réponse pouvant être renvoyés par la demande. Ce tableau n'est pas exhaustif.

Code de réponse	Explication
200 OK	La demande a abouti et les valeurs de propriété ont été modifiées.
400 Bad Request	Ce code de réponse peut être émis pour l'une des raisons suivantes : <ul style="list-style-type: none">• Des données JSON non valides ont été envoyées.• Un type de données non valide a été fourni pour une propriété existante.• Une tentative a été effectuée pour remplacer la valeur d'une propriété chiffrée.• Une tentative a été effectuée pour créer une nouvelle propriété dont la valeur n'est pas une chaîne.
500 Internal Server Error	Une erreur interne a empêché l'exécution de l'action. Pour résoudre les problèmes liés à la demande, effectuez une demande GET pour vérifier si des propriétés requises ont été modifiées. Si tel n'est pas le cas, soumettez à nouveau la demande.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

«Définir les demandes PATCH ou POST comme bloquantes ou non bloquantes», à la page 127

Les demandes qui changent les valeurs ou les actions des déclencheurs avec PATCH ou POST peuvent être définies comme bloquantes ou non bloquantes.

Événement d'accusé de réception et données utiles

Vous pouvez utiliser la section "options" de la charge JSON de l'URI /probe/common pour demander la génération d'un événement d'accusé de réception en réponse à une demande HTTP. L'accusé de réception d'événement contient les détails de la demande HTTP et ses résultats et est envoyé via le fichier de règles de sonde au serveur ObjectServer. Vous pouvez ajouter des données utiles à l'événement d'accusé de réception ; ces données peuvent être des informations spécifiques à l'événement ou au serveur ObjectServer.

Vous pouvez utiliser ces données pour corréler l'événement d'accusé de réception avec des événements ou des serveurs ObjectServer spécifiques. En outre, si vous ajoutez des données utiles à l'événement d'accusé de réception, vous pouvez écrire des déclencheurs qui capture les résultats des demandes HTTP et mettre à jour les événements correspondants. Ces déclencheurs peuvent être déclenchés depuis des outils par clic droit ou des automatisations.

Pour un événement de réception, définissez l'objet "event_ack" de la section "options" sur true. Ce paramètre indique la bibliothèque C de sondes standard (lipOpl) pour envoyer un événement d'accusé de réception au serveur ObjectServer, via le fichier de règles. Pour plus d'informations, voir «Exemple 1 : Événement d'accusé de réception».

Pour un événement de réception avec des données utiles, définissez l'objet "event_ack" de la section "options" sur true. De plus, dans l'objet "event_ack_payload", ajoutez un ensemble de paires nom-valeur de données utiles à joindre à l'événement de réception. L'objet "event_ack" doit être une liste à plat et ne peut contenir aucun objet JSON enfant, tableau ou élément NULL. Tous les éléments de la liste de paires nom-valeur doivent être des chaînes car ces éléments sont mappés avec les éléments du fichier de règles de sonde avant que le traitement de l'événement de réception via le fichier de règles. Pour plus d'informations, voir «Exemple 2 : Événement d'accusé de réception avec données utiles», à la page 126.

Exemple 1 : Événement d'accusé de réception

L'exemple suivant montre la section "options" des données utiles JSON pour demander la génération d'un accusé de réception à l'aide d'une demande PATCH.

```
{
  "options": { "block_response" : false,
              "event_ack" : true
            },
  "properties": {
    "MyNewProperty": "myvalue"
  }
}
```

L'objet {"event_ack" : true } indique à libOpl à envoyer un événement d'accusé de réception au serveur ObjectServer par défaut. Pour la demande utilisée dans cet exemple, un événement d'accusé de réception entrant dans le fichier de règles de sonde équivaut aux éléments suivants :

```
@Name=nom_propriété
@Agent=propriété_agent
@Node=nom_hôte
@OwnerGID=1
@Severity=1
@Manager="ProbeWatch"
@Summary="HTTP event_ack ..."
@Identifier=@Name+"@"+@Node+": "+@Summary
```

```
$req_method="PATCH"
$request_outcome="OK"
$req_payload_type="application/json"
$req_port="6789"
$req_uri="/probe/common"
$req_http_type="https"
$req_hostname="probehost.example.com"
$req_auth_type="none"
$http_return_reason="Accepted"
$req_username="test"
$req_ipaddr="192.168.1.1"
$http_return_code="202"
$req_payload_length="178"
```

Dans cet exemple, *nom_propriété* représente la valeur de la propriété **Name** de la sonde, *propriété_agent* correspond à la valeur de la propriété **Agent** de la sonde, *nom_hôte* désigne le nom de l'ordinateur hôte sur lequel la sonde est installée.

Exemple 2 : Événement d'accusé de réception avec données utiles

L'exemple suivant montre l'objet "event_ack_payload" de la section "options" des données utiles JSON, défini avec un ensemble arbitraire de paires nom-valeur de données utiles pour une demande PATCH.

```
{
  "options": {
    "block_response" : false,
    "event_ack" : true,
    "event_ack_payload" : {
      "name1" : "value1",
      "name2" : "value2"
    }
  },
  "properties": {
    "MyNewProperty": "myvalue"
  }
}
```

Pour la demande utilisée dans cet exemple, un événement d'accusé de réception entrant dans le fichier de règles de sonde équivaut aux éléments suivants :

```
@Name=nom_propriété
@Agent=propriété_agent
@Node=nom_hôte
@OwnerGID=1
@Severity=1
@Manager="ProbeWatch"
@Summary="HTTP event_ack ..."
@Identifier=@Name+"@"+@Node+": "+@Summary

$req_method="PATCH"$request_outcome="OK"
$req_payload_type="application/json"
$req_port="6789"
$req_uri="/probe/common"
$req_http_type="https"
$req_hostname="probehost.example.com"
$name1="value1"
$name2="value2"
$req_auth_type="none"
$http_return_reason="Accepted"
$req_username="test"
$req_ipaddr="192.168.1.1"
$http_return_code="202"
$req_payload_length="178"
```

Dans cet exemple, *nom_propriété* représente la valeur de la propriété **Name** de la sonde, *propriété_agent* correspond à la valeur de la propriété **Agent** de la sonde, *nom_hôte* désigne le nom de l'ordinateur hôte sur lequel la sonde est installée.

L'événement d'accusé de réception est identique à l'événement demandé dans «Exemple 1 : Événement d'accusé de réception», à la page 125, à l'exception des données utiles qui comprennent les éléments "name1" et "name2".

Définir les demandes PATCH ou POST comme bloquantes ou non bloquantes

Les demandes qui changent les valeurs ou les actions des déclencheurs avec PATCH ou POST peuvent être définies comme bloquantes ou non bloquantes.

Dans une demande bloquante (valeur par défaut), l'action décrite par les données utiles est validée. La demande est complétée avec le code de réponse HTTP obtenu pour signaler la réussite ou l'échec. La réussite est indiquée par le code retour 200 – OK et l'échec, par les codes retour HTTP les plus appropriés.

Pour effectuer une demande non bloquante, utilisez la section "options" des données utiles JSON d'une action. L'exemple de données /probe/common PATCH ci-après est utilisé pour créer une nouvelle propriété et ajouter une section "options" qui indique les données JSON supplémentaires.

```
{
  "options": {"block_response" : false },
  "properties": {
    "MyNewProperty" : "myvalue"
  }
}
```

Après la validation syntaxique et sémantique de cette demande, le serveur HTTP répond avec le code de réponse 202-Accepted. A ce stade, l'action est demandée mais n'est pas exécutée.

En cas de demande non valide non bloquante, un code de réponse HTTP d'échec est renvoyé. L'exemple suivant renvoie un code de réponse 400-BAD REQUEST car la nouvelle valeur de propriété n'est pas une chaîne. Chaque nouvelle propriété créée doit comporter une valeur de chaîne.

```
{
  "options" : {
    "block_response" : false },
  "properties": {
    "MyNewProperty": 1
  }
}
```

Les demandes non bloquantes acceptées (avec { "block_response" : false }) peuvent encore échouer en raison de problèmes internes inattendus. La connexion HTTP étant effectivement fermée, l'interface HTTP ne peut pas informer le client des problèmes ultérieurs. C'est pourquoi un «événement d'accusé de réception» facultatif peut être généré par l'interface HTTP pour ces demandes.

Référence associée:

«Recharger le fichier de règles», à la page 119

Cette demande accède à l'attribut reloadrulesflag de la sonde, contenu dans la bibliothèque C de sondes standard (libOpl). En définissant la valeur de cet attribut, la demande force la sonde à recharger son fichier de règles avant le traitement de l'événement suivant. Un signal SIGHUP sur UNIX déclenche la même action.

«Créer un événement synthétique», à la page 121

Cette demande crée un événement synthétique pouvant déclencher l'exécution du fichier de règles de sonde ou d'un fragment de ce fichier de règles.

«Définir une propriété de sonde», à la page 123

Cette demande change les valeurs de propriétés de sonde. Elle peut également créer de nouvelles propriétés. Toutes les nouvelles propriétés créées de cette manière nécessitent une valeur de chaîne.

Chapitre 6. Propriétés et options de ligne de commande de sonde communes

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

Pour les propriétés et options de ligne de commande spécifiques à une sonde ou à un gestionnaire TSM (Telco Service Manager) précis, voir le guide de référence de chaque sonde et gestionnaire TSM.

Conseil : Vous pouvez chiffrer les valeurs de chaîne dans un fichier de propriétés en utilisant le chiffrement de valeur de propriété.

Lors du démarrage, une sonde lit les valeurs de propriété indiquées dans son fichier de propriétés. Certaines propriétés ne sont lues qu'au démarrage et, par conséquent, leur modification pendant l'exécution de la sonde est sans effet. En revanche, les valeurs d'autres propriétés sont modifiables lorsque la sonde est en cours d'exécution. Parmi ces propriétés, certaines modifications de valeur prennent effet immédiatement tandis que d'autres n'entrent en vigueur qu'après la relecture du fichier de règles. Lorsque vous forcez une relecture du fichier de règles, la connexion au serveur ObjectServer est supprimée puis rétablie, ce qui pousse la sonde à revérifier la valeur de la propriété.

Pour modifier la valeur d'une propriété, vous pouvez utiliser les méthodes suivantes :

- Modifiez la valeur de propriété dans le fichier de propriétés. Vous devez arrêter une sonde puis la redémarrer pour que les modifications prennent effet.
- Lors du démarrage d'une sonde, utilisez les options de ligne de commande disponibles pour affecter des valeurs temporaires à une ou plusieurs propriétés.
- Pour mettre à jour une valeur de propriété d'une sonde en cours d'exécution, utilisez l'interface HTTP (si disponible) de cette sonde. Selon la propriété, ces modifications prennent effet immédiatement ou après le rechargement du fichier de règles.
- Editez le fichier de règles et utilisez le paramètre *%Propriété* pour modifier une valeur de propriété d'une sonde en cours d'exécution. Vous devez recharger le fichier de règles pour que les modifications prennent effet.

Le tableau ci-dessous répertorie les propriétés et les options de ligne de commande communes et disponibles pour toutes les sondes et fournit leurs paramètres par défaut. Les valeurs de chaînes en caractères non ASCII ne sont pas prises en charge. La colonne **Type de mise à jour** indique le moment où les modifications des valeurs de propriété prennent effet : lors du démarrage uniquement, immédiatement ou lors de la reconnexion de la sonde au serveur ObjectServer.

Tableau 33. Propriétés et options de ligne de commande de sonde communes.

Propriété	Option de ligne de commande	Description	Type de mise à jour
AuthPassword chaîne	N/D	Spécifie le mot de passe associé au nom d'utilisateur utilisé pour authentifier la sonde lorsqu'elle se connecte à un serveur proxy ou à un serveur ObjectServer exécuté en mode sécurisé. La valeur par défaut est ''. Lorsque le mode FIPS 140-2 est activé, le mot de passe peut être spécifié en texte normal ou chiffré à l'aide de l'utilitaire nco_aes_crypt . Si vous chiffrez des mots de passe à l'aide de l'utilitaire nco_aes_crypt en mode FIPS 140-2, vous devez spécifier AES_FIPS comme algorithme de chiffrement. Lorsque le mode FIPS 140-2 est désactivé, le mot de passe peut être chiffré à l'aide des utilitaires nco_g_crypt ou nco_aes_crypt . Si vous chiffrez des mots de passe à l'aide de l'utilitaire nco_aes_crypt en mode FIPS 140-2 désactivé, vous pouvez spécifier AES_FIPS ou AES comme algorithme de chiffrement. Utilisez uniquement AES si vous devez conserver une compatibilité avec des mots de passe chiffrés à l'aide des outils fournis dans des versions antérieures à Tivoli Netcool/OMNIBus V7.2.1.	Lors de la reconnexion
AuthUserName chaîne	N/D	Spécifie un nom d'utilisateur utilisé pour authentifier la sonde lorsqu'elle se connecte à un serveur proxy ou à un serveur ObjectServer exécuté en mode sécurisé. La valeur par défaut est ''.	Lors de la reconnexion
AutoSAF 0 1	-autosaf -noautosaf	Spécifie si le mode stocker-et-transmettre automatique est activé. Avec ce mode, si la sonde démarre mais ne peut pas envoyer d'événements au serveur ObjectServer, elle passe en mode stockage au lieu de s'arrêter. Ce mode est désactivé par défaut (0). Remarque : Pour que la fonction stocker-et-transmettre automatique fonctionne, la sonde doit s'être connectée au moins une fois au serveur ObjectServer pour connaître le format de stockage des événements dans cet ObjectServer. Si la sonde tente de se connecter à une paire virtuelle de serveurs ObjectServer et que les deux serveurs ObjectServer sont hors tension, la sonde vérifie le paramètre de la propriété AutoSAF . Si le stockage et le réacheminement automatiques sont activés, la sonde commence à stocker des événements dans le fichier stocker-et-transmettre. Dans le cas contraire, la sonde s'arrête.	Lors du démarrage uniquement
BeatInterval entier	-beatinterval entier	Spécifie l'intervalle des signaux de présence pour la reprise en ligne d'égal à égal. La valeur par défaut est 2 secondes.	Lors du démarrage uniquement
BeatThreshold entier	-beatthreshold entier	Spécifie la période d'attente supplémentaire de la sonde esclave d'une relation de reprise en ligne d'égal à égal avant de passer en mode actif. La valeur par défaut est 1 seconde.	Lors du démarrage uniquement
BufferFlush Interval entier	-bufferflushinterval entier	Indique l'intervalle de temps (en secondes) pendant lequel une sonde attend avant de vider des alertes vers le serveur ObjectServer. Cette propriété limite le temps d'attente d'une alerte dans la mémoire tampon alors que cette dernière est à la taille maximale spécifiée par la propriété BufferSize . Pour utiliser cette propriété, vous devez activer la propriété Buffering et définir la propriété BufferSize avec une valeur supérieure à 0. Remarque : Certaines sondes ont la propriété FlushBufferInterval qui fournit une fonction équivalente à la propriété BufferFlushInterval . Si les deux propriétés sont définies pour une sonde, un avertissement est consigné dans le journal et la propriété BufferFlushInterval n'est pas activée. La valeur par défaut est 0 secondes.	Lors du démarrage uniquement

Tableau 33. Propriétés et options de ligne de commande de sonde communes. (suite)

Propriété	Option de ligne de commande	Description	Type de mise à jour
Buffering 0 1	-buffer -nobuffer	Spécifie si la mise en mémoire tampon est utilisée lors de l'envoi d'alertes au serveur ObjectServer. Par défaut, la mise en mémoire tampon est désactivée (0). Toutes les alertes envoyées à la même table sont envoyées dans l'ordre dans lequel elles sont traitées par la sonde. Si des alertes sont envoyées à plusieurs tables, l'ordre est conservé pour chaque table, mais pas entre les tables. Si le traitement multiprocessus fonctionne (la valeur par défaut), une unité d'exécution de communication distincte est utilisée pour envoyer des données à chaque ObjectServer cible enregistré et une mémoire tampon distincte est par conséquent conservée pour chaque ObjectServer. Remarque : Les messages ProbeWatch ne sont jamais placés en mémoire tampon. Un événement ProbeWatch entraîne le vidage de la mémoire tampon de sorte que le message soit envoyé au serveur ObjectServer dès que possible.	Lors du démarrage uniquement
BufferSize entier	-buffersize entier	Indique le nombre d'alertes que la sonde met en cache lorsque la propriété Buffering est définie sur 1. Le nombre est défini par défaut sur 10.	Lors du démarrage uniquement
CacheRules 0 1	-cacherules -dontcacherules	Utilisez cette propriété pour que la sonde place son fichier de règles en cache chaque fois qu'elle est redémarrée ou qu'elle doit relire son fichier de règles. Au démarrage de la sonde, si cette dernière ne parvient pas à lire le fichier de règles, elle lit le fichier de cache à la place. Par défaut, la mise en cache du fichier de règles est désactivée, c'est-à-dire que la propriété a pour valeur 0.	Lors de la reconnexion
CacheRulesFile chaîne	-cacherulesfile chaîne	Utilisez cette propriété pour spécifier le fichier dans lequel vous souhaitez que la sonde place son fichier de règles dans la mémoire cache. La valeur par défaut est \$OMNIHOME/var/nom_sonde.rulescache, où <i>nom_sonde</i> représente le nom abrégé utilisé pour identifier la sonde ; par exemple <i>simnet</i> pour la sonde <i>simnet</i> . Vous ne devez définir cette propriété que si vous définissez la propriété CacheRules sur 1 et que vous souhaitez modifier l'emplacement du fichier de cache en lui donnant une valeur autre que la valeur d'emplacement par défaut.	Lors de la reconnexion
ConfigCryptoAlg chaîne	N/D	Indique l'algorithme de cryptographie à utiliser pour le déchiffrement des valeurs de chaîne (dont les mots de passe) chiffrés à l'aide de l'utilitaire nco_aes_crypt et stockés ensuite dans le fichier de propriétés. Définissez la valeur <i>chaîne</i> comme suit : <ul style="list-style-type: none"> • En mode FIPS 140-2, utilisez AES_FIPS. • En mode FIPS 140-2 désactivé, vous pouvez utiliser AES_FIPS ou AES. Utilisez uniquement AES si vous devez conserver une compatibilité avec des mots de passe chiffrés à l'aide des outils fournis dans des versions antérieures à Tivoli Netcool/OMNIBus version 7.2.1. La valeur que vous indiquez doit être identique à celle utilisée lors de l'exécution de nco_aes_crypt avec le paramètre -c pour chiffrer les valeurs de chaîne. Utilisez cette propriété avec la propriété ConfigKeyFile .	Lors du démarrage uniquement
ConfigKeyFile chaîne	N/D	Indique le chemin et le nom du fichier de clés contenant la clé utilisée pour déchiffrer les valeurs de chaîne chiffrées (dont les mots de passe) dans le fichier de propriétés. La clé est utilisée au moment de l'exécution pour déchiffrer les valeurs de chaîne chiffrées à l'aide de l'utilitaire nco_aes_crypt . Le fichier de clés que vous avez spécifié doit être identique au fichier utilisé pour chiffrer les valeurs de chaîne lors de l'exécution de nco_aes_crypt avec le paramètre -k . Utilisez cette propriété avec la propriété ConfigCryptoAlg .	Lors du démarrage uniquement

Tableau 33. Propriétés et options de ligne de commande de sonde communes. (suite)

Propriété	Option de ligne de commande	Description	Type de mise à jour
Fix Pack 1 DisableDetails 0 1	-disabdetails ou -enabledetails	Demande à la sonde d'omettre les détails d'alerte, c'est-à-dire les instructions INSERT dirigées vers la table alerts.details, quand des alertes sont transmises au serveur ObjectServer. Pour omettre les détails, indiquez 1 pour la propriété DisableDetails ou exécutez la sonde avec l'option -disabdetails. Pour envoyer les détails, indiquez 0 pour la propriété ou exécutez la sonde avec l'option -enabledetails. Les deux option -disabdetails et -enabledetails prévalent sur la définition de la propriété. Ainsi, si la propriété est définie pour l'omission des détails, vous pouvez l'ignorer en exécutant la sonde avec l'option -enabledetails. Si le serveur ObjectServer est exécuté en mode débogage, une ligne est écrite dans le journal de la sonde chaque fois qu'une alerte est reçue et que les détails sont omis. Exemple : 2014-07-16T11:24:21: Debug: D-UNK-000-000: Details disabled - Not sending alert details to ObjectServer	Immédiatement
Dumpprops chaîne	-dumpprops chaîne	Affiche les paramètres de propriété en cours d'une sonde. Exécutez l'analyse avec l'option -dumpprops - exemple : \$OMNIHOME/probes/nco_p_tivoli_eif -dumpprops	Lors du démarrage uniquement
N/D	-help	Affiche les options de ligne de commande prises en charge et quitte.	Lors du démarrage uniquement
KeepLastBroken SAF 0 1	-keeplastbrokensaf -dontkeeplastbroken saf	Spécifie si les enregistrements stocker-et-transmettre corrompus doivent être automatiquement sauvegardés pour un diagnostic ultérieur. Si cette option est définie sur 1, les enregistrements stocker-et-transmettre corrompus sont automatiquement sauvegardés. La valeur par défaut est 0. Utilisez cette propriété avec la propriété StoreSAFRejects .	Immédiatement
LogFilePoolSize <i>entier</i>	-logfilepoolsize entier	Spécifie le nombre de fichiers journaux à utiliser si le système de consignation écrit dans un pool de fichiers. Cette propriété ne fonctionne que si la propriété LogFileUsePool est définie sur TRUE. La taille du pool peut être comprise entre 2 et 99. Le nombre est défini par défaut sur 10.	Lors du démarrage uniquement

Tableau 33. Propriétés et options de ligne de commande de sonde communes. (suite)

Propriété	Option de ligne de commande	Description	Type de mise à jour
LogFileUsePool 0 1	-logfileusepool -nologfileusepool	<p>Spécifie si un pool de fichiers journaux doit être utilisé pour les messages de consignation.</p> <p>Si cette option est définie sur 1, le système de consignation ouvre le nombre de fichiers spécifié pour le pool au démarrage et les garde ouverts pendant la durée de son exécution. (Vous définissez le nombre de fichiers du pool en utilisant la propriété LogFilePoolSize.) Lorsqu'un fichier du pool atteint sa limite maximale (spécifiée par la propriété MaxLogFileSize), le système écrit dans le prochain fichier. Lorsque tous les fichiers du pool ont atteint leur taille maximale, le système de consignation tronque le premier fichier du pool et recommence à écrire dans ce fichier. Les fichiers du pool sont désignés à l'aide du format <i>nom_sonde.log_ID</i>, où <i>ID</i> est un numéro à deux chiffres de 01 au nombre maximum spécifié pour la propriété LogFilePoolSize. Lorsque le système de consignation commence à utiliser un pool de fichiers, il écrit dans le numéro de fichier le plus faible disponible, quel que soit le fichier dans lequel il écrivait lors de sa dernière exécution.</p> <p>La valeur par défaut est 0. Lorsqu'elle est définie sur 0, le fichier <i>nom_sonde.log</i> est renommé en <i>nom_sonde.log_OLD</i> et un nouveau fichier journal est démarré lorsque la taille maximale est atteinte. Si le fichier ne peut pas être renommé, par exemple en raison d'un verrou de lecture sur le fichier <i>_OLD</i> et que LogFileUseStdErr est défini sur 0, le système de consignation démarre automatiquement en utilisant un pool de fichiers journaux. Si le fichier ne peut pas être renommé et que LogFileUseStdErr est défini sur 1, les messages sont consignés dans la console si la sonde a été exécutée depuis la ligne de commande. Si le fichier ne peut pas être renommé et que LogFileUseStdErr est défini sur 1, les messages sont consignés dans un fichier appelé <i>%CHOME%\omnibus\log\nom_sonde.err</i> si la sonde est exécutée en tant que service Windows.</p>	Lors du démarrage uniquement
LogFileUseStdErr 0 1	-logfileusestderr -nologfileusestderr	<p>Indique si l'erreur standard (stderr) doit être utilisée comme flux de sortie pour les messages de consignation.</p> <p>La valeur par défaut est 0 : lorsqu'elle est utilisée, le système de consignation copie les données dans le fichier journal par défaut ou dans un pool de fichiers journaux, cela étant défini par la propriété LogFileUsePool.</p> <p>Si ce paramètre a pour valeur 1, les messages sont consignés sur la console lorsque la sonde est exécutée à partir de la ligne de commande. Remarque :</p> <p>La définition de la propriété LogFileUsePool prévaut sur celle de LogFileUseStdErr.</p>	Lors du démarrage uniquement
LookupTableMode <i>entier</i>	-lookupmode <i>entier</i>	<p>Spécifie comment les recherches de tables sont effectuées. Cette option peut être définie sur 1, 2 ou 3. La valeur par défaut est 3.</p> <p>Si cette propriété est définie sur 1, toutes les tables de consultation externes sont supposées avoir une seule colonne de valeur. Les tabulations ne sont pas utilisées comme délimiteurs de colonne.</p> <p>Si elle est définie sur 2, toutes les tables de consultation externes sont supposées avoir plusieurs colonnes. Si le nombre de colonnes de chaque ligne est différent, une erreur est générée et comprend le nom du fichier ainsi que la ligne sur laquelle l'erreur s'est produite.</p> <p>Si la propriété est définie sur 3, le moteur de règles tente de déterminer le nombre de colonnes dans la table de consultation externe. Une erreur est générée pour chaque ligne qui a un nombre de colonnes différent de la ligne précédente. L'erreur inclut le nom du fichier et la ligne sur laquelle l'erreur est apparue.</p>	Lors de la reconnexion
Manager chaîne	-manager <i>chaîne</i>	Spécifie la valeur de la zone Manager pour l'alerte. La valeur par défaut est déterminée par la sonde.	Immédiatement
MaxLogFileSize <i>entier</i>	-maxlogfilesiz <i>entier</i>	Spécifie la taille maximale du fichier journal, en octets. La valeur par défaut est 1 Mo. Lorsque le fichier journal atteint la taille spécifiée, un deuxième fichier journal est démarré. Lorsque le deuxième fichier atteint la taille maximale, le premier fichier est écrasé par un nouveau fichier journal et le processus recommence.	Lors du démarrage uniquement

Tableau 33. Propriétés et options de ligne de commande de sonde communes. (suite)

Propriété	Option de ligne de commande	Description	Type de mise à jour
MaxRawFileSize entier	N/D	Spécifie la taille maximale du fichier de capture brute en ko. La valeur par défaut est illimitée (-1).	Immédiatement
MaxSAFFileSize entier	-maxsaffilesize entier	Indique la taille maximale (en octets) que le fichier stocker-et-transmettre peut atteindre lorsque la connexion au serveur ObjectServer n'est pas établie. La valeur par défaut est 1 Mo.	Immédiatement
MessageLevel chaîne	-messagelevel chaîne	Indique le niveau de consignation du message. Les valeurs possibles sont : debug, info, warn, error et fatal. Le niveau par défaut est warn. Les messages qui sont consignés à chaque niveau sont les suivants : fatal : fatal uniquement. error : fatal et error. warn : fatal, error et warn. info : fatal, error, warn, et info. debug : fatal, error, warn, info, et debug.	Lors du démarrage uniquement
MessageLog chaîne	-messagelog chaîne	Spécifie l'endroit où sont stockés les messages. La valeur par défaut est \$OMNIHOME/log/nom_sonde.log. MessageLog peut également être défini sur stdout ou stderr.	Lors du démarrage uniquement
Mode chaîne	-master -slave	Spécifie le rôle de l'instance de la sonde dans une relation de reprise en ligne d'égal à égal. La propriété peut être définie sur l'une des valeurs suivantes : master : cette instance est l'instance maître. slave : cette instance est l'instance esclave. standard : il n'y a pas de relation de reprise en ligne. La valeur par défaut est standard.	Lors du démarrage uniquement
MsgDailyLog 0 1	-msgdailylog 0 1	Spécifie si la consignation quotidienne est activée. Par défaut, la sauvegarde quotidienne des fichiers journaux est désactivée (0). Remarque : Etant donné que l'heure est vérifiée régulièrement, il peut y avoir une légère baisse des performances lorsque MsgDailyLog est activé.	Lors du démarrage uniquement
MsgTimeLog chaîne	-msgtimelog chaîne	Spécifie l'heure de création du journal quotidien. La valeur par défaut est 0000 (minuit). Si MsgDailyLog est défini sur 0, la valeur est ignorée.	Lors du démarrage uniquement
Name chaîne	-name chaîne	Spécifie le nom de la sonde. Cette valeur détermine les noms du fichier de propriétés, du fichier de règles, du fichier journal des messages, du fichier stocker-et-transmettre et du fichier de capture brute. Remarque : Vous pouvez spécifier des noms de fichiers alternatifs à l'aide des propriétés PropsFile , RulesFile , MessageLog , SAFFilename et RawCaptureFile . Si vous souhaitez définir un de ces noms de fichiers dans le fichier de propriétés, ils doivent être spécifiés après la propriété Name . Sinon, la propriété Name remplacera toute définition précédente des noms de fichiers. La valeur de la propriété Name est incluse dans la clé primaire de l'entrée de sonde dans la table registry.probes ObjectServer.	Lors de la reconnexion

Tableau 33. Propriétés et options de ligne de commande de sonde communes. (suite)

Propriété	Option de ligne de commande	Description	Type de mise à jour
NetworkTimeout entier	-networktimeout entier	Spécifie la durée (en secondes) pendant laquelle une sonde peut attendre sans recevoir de réponse ; lorsqu'elle est écoulée, la connexion avec le serveur ObjectServer expire. La valeur maximale est 2147483 et la valeur par défaut est 0, ce qui signifie que la connexion n'expire pas. En cas de dépassement du délai d'attente, la sonde tente de se connecter au serveur ObjectServer de secours identifié par la propriété ServerBackup . Si le délai d'attente est dépassé mais qu'aucun serveur ObjectServer de secours n'est spécifié, la sonde entre en mode stocker-et-transmettre. Dans une configuration de reprise standard, la valeur de la propriété NetworkTimeout doit être inférieure à la valeur de la propriété PoolServer . Le paramètre NetworkTimeout remplace le paramètre d'expiration TCP/IP au niveau du système d'exploitation.	Lors de la reconnexion
NHttpd.Access Log chaîne	-nhttpd_accesslog chaîne	Indique le nom du fichier journal dans lequel le serveur consigne toutes les demandes qu'il traite. La valeur par défaut est access.log.	Lors du démarrage uniquement
NHttpd.AuthenticationDomain chaîne	-nhttpd_authdomain chaîne	Indique le domaine d'authentification utilisé lors de la demande des détails d'authentification sur la connexion HTTP ou HTTPS. La valeur par défaut est omnibus.	Lors du démarrage uniquement
NHttpd.BasicAuth chaîne	-nhttpd_basicauth chaîne	Indique la combinaison utilisateur-mot de passe qui est autorisée pour les connexions à l'interface HTTP. Veillez à ce que le mot de passe associé au nom d'utilisateur soit chiffré en mode de hachage AES-128. Utilisez nco_crypt avec l'option -aes pour chiffrer le mot de passe. Indiquez le mot de passe selon le format " <i>nom_utilisateur:mot_de_passe</i> ", où <i>nom_utilisateur</i> désigne toute combinaison de caractères qui n'inclut pas le signe deux-points (:) et <i>mot_de_passe</i> représente le hachage AES-128 du mot de passe. La valeur par défaut de cette propriété est "", ce qui permet d'accepter toute donnée d'identification utilisateur.	Lors du démarrage uniquement
NHttpd.DocumentRoot chaîne	-nhttpd_docroot chaîne	Indique la racine de document des demandes de service Web intégré. La valeur par défaut est ./.	Lors du démarrage uniquement
NHttpd.EnableFileServing TRUE FALSE	-nhttpd_enablefs TRUE FALSE	Utilisez cette propriété pour activer la réalisation du service de fichiers par défaut par la sonde. Cela permet à la sonde d'agir comme un serveur HTTP simple qui sert les fichiers à partir du système de fichiers local. La valeur par défaut est FALSE.	Lors du démarrage uniquement
NHttpd.EnableHTTP TRUE FALSE	-nhttpd_enablehttp	Utilisez cette propriété pour activer l'utilisation du port HTTP. La valeur par défaut est TRUE.	Lors du démarrage uniquement
NHttpd.ExpireTimeout chaîne	-nhttpd_expire timeout chaîne	Indique la durée d'inactivité maximale, en secondes, à l'issue de laquelle une connexion HTTP/1.1 est supprimée. La valeur par défaut est 15 secondes.	Lors du démarrage uniquement

Tableau 33. Propriétés et options de ligne de commande de sonde communes. (suite)

Propriété	Option de ligne de commande	Description	Type de mise à jour
NHttpd. Listening Hostname chaîne	-nhhttpd_hostname chaîne	Indique le nom d'hôte ou l'adresse IP en mode écoute qui peut être utilisé comme composant hostname d'un identificateur URI pour l'interface HTTPS ou HTTP de la sonde. La sonde enregistre la valeur de cette propriété dans la colonne Hostname de la table registry.probes. Vous pouvez utiliser cette propriété pour spécifier un nom d'hôte différent que le nom d'hôte local par défaut. Vous pouvez être amené à effectuer cette opération si le nom d'hôte qui est automatiquement résolu n'est pas un nom de domaine qualifié complet (FQDN) ou s'il n'existe pas de résolution de nom d'hôte à distance pour ce nom. Par exemple, cela peut se produire avec des machines qui sont mises à disposition en tant que machines virtuelles dans un sous-réseau virtuel privé, avec des machines qui possèdent plusieurs cartes réseau, avec des machines qui sont configurées sans paramètre de domaine local, ou encore avec des machines qui n'ont pas accès à des services de résolution d'espace-noms. La valeur par défaut est le nom d'hôte de l'ordinateur local. Vous pouvez connaître ce nom d'hôte en utilisant la commande de système d'exploitation hostname .	Lors du démarrage uniquement
NHttpd. Listening Port entier	-nhhttpd_port entier	Indique le port sur lequel la sonde écoute les demandes HTTP. La valeur par défaut est 8080.	Lors du démarrage uniquement
NHttpd.NumWork Threads entier	-nhhttpd_numworkthrs entier	Indique le nombre maximum d'unités d'exécution de tâche pouvant être utilisées pour effectuer le service des demandes HTTP ou HTTPS entrantes. Utilisez cette propriété pour spécifier le nombre de demandes HTTP ou HTTPS que la sonde peut gérer simultanément. Chaque demande de la sonde est traitée dans une unité d'exécution de tâche unique, qui est rendue au pool d'unités d'exécution une fois la demande traitée. La valeur par défaut est 5.	Lors du démarrage uniquement
NHttpd.SSL Certificate chaîne	-nhhttpd_sslcert chaîne	Indique le nom du certificat SSL (Secure Sockets Layer) du serveur. La valeur par défaut est cacert.pem.	Lors du démarrage uniquement
NHttpd.SSL Certificate Pwd chaîne	-nhhttpd_sslcertpwd chaîne	Remarque : Cette propriété est obsolète avec Tivoli Netcool/OMNIbus version 8.1. La définir est sans effet.	Lors du démarrage uniquement
NHttpd.SSL Enable TRUE FALSE	-nhhttpd_sslenable	Utilisez cette propriété pour activer l'utilisation du support SSL. La valeur par défaut est FALSE.	Lors du démarrage uniquement
NHttpd.SSL Listening Port entier	-nhhttpd_sslport entier	Indique le port sur lequel la sonde écoute les demandes HTTPS. La valeur par défaut est 0.	Lors du démarrage uniquement
OldTimeStamp TRUE FALSE	-oldtimestamp TRUE FALSE	Indique le format d'horodatage à utiliser dans le fichier journal. Définissez la valeur sur TRUE pour afficher le format d'horodatage utilisé dans Tivoli Netcool/OMNIbus version 7.2.1 ou supérieure. Par exemple : jj/MM/AAAA hh:mm:ss AM ou jj/MM/AAAA hh:mm:ss PM lorsque les paramètres régionaux sont définis sur en_GB sur un ordinateur Solaris 9. Définissez la valeur sur FALSE pour afficher le format ISO 8601 dans le fichier journal. Par exemple : AAAA-MM-JJThh:mm:ss, où T sépare la date de l'heure, et hh représente l'heure au format 24 heures. La valeur par défaut est FALSE.	Lors du démarrage uniquement
PeerHost chaîne	-peerhost chaîne	Spécifie le nom d'hôte de l'élément réseau qui agit en tant qu'homologue de cette instance de sonde dans une relation de reprise par en ligne d'égal à égal. La valeur par défaut est localhost.	Lors du démarrage uniquement
PeerPort entier	-peerport entier	Spécifie le port via lequel le maître et l'esclave communiquent dans une relation de reprise en ligne d'égal à égal. Le port par défaut est 9999.	Lors du démarrage uniquement

Tableau 33. Propriétés et options de ligne de commande de sonde communes. (suite)

Propriété	Option de ligne de commande	Description	Type de mise à jour
PollServer entier	-pollserver entier	<p>Si elle est connectée à un serveur ObjectServer de sauvegarde suite à une reprise en ligne, une sonde tente régulièrement de se reconnecter au serveur ObjectServer principal. Cette propriété indique la fréquence (en secondes) à laquelle la sonde interroge le serveur ObjectServer principal pour déterminer s'il est de nouveau disponible. Elle fait cela en créant une connexion de test temporaire au serveur ObjectServer principal. Si la connexion de test temporaire aboutit, la sonde reprend le serveur ObjectServer principal et supprime la connexion au serveur ObjectServer de secours. Cette connexion n'est supprimée que si le basculement vers le serveur ObjectServer principal réussit.</p> <p>L'interrogation est la seule manière qu'a la sonde pour déterminer si le serveur ObjectServer principal est disponible. La valeur par défaut est 0, ce qui signifie qu'aucune interrogation n'a lieu.</p> <p>Lorsqu'une sonde se connecte à un serveur ObjectServer, elle recherche la définition de la propriété BackupObjectServer du serveur auquel elle se connecte. L'interrogation périodique n'a lieu que si cette propriété a la valeur TRUE, ce qui indique la présence d'un serveur ObjectServer de secours.</p> <p>Remarque : Une sonde peut entrer en mode stocker-et-transmettre lorsque le serveur ObjectServer principal devient indisponible. La première alerte n'est transmise au serveur ObjectServer de sauvegarde que lorsque la deuxième alerte ouvre une connexion à ce serveur. Si PollServer est définie sur une valeur inférieure à la durée moyenne entre les alertes, la connexion au serveur ObjectServer est interrogée avant l'envoi d'une alerte et la sonde n'entre pas en mode stocker-et-transmettre. Pour la reprise par restauration contrôlée, définissez PollServer sur 0 pour désactiver la reprise par restauration automatique d'une sonde connectée à une paire de serveurs ObjectServer de reprise en ligne.</p>	Lors de la reconnexion
ProbeWatch Heartbeat Interval entier	-probewatchheart beatinterval entier	Génère un événement de signal de présence ProbeWatch si cette propriété est définie sur un nombre positif. Ce numéro spécifie l'intervalle (en secondes) de génération des signaux de présence. Si la propriété est définie sur 0 (zéro) ou un nombre négatif, aucun signal de présence ProbeWatch n'est généré.	Immédiatement
Props.Check Names TRUE FALSE	N/D	Si TRUE, la sonde n'est pas exécutée et une propriété spécifiée est invalide. La valeur par défaut est TRUE.	Lors du démarrage uniquement
PropsFile chaîne	-propsfile chaîne	Spécifie le nom du fichier de propriétés. La valeur par défaut est \$OMNIHOME/probes/arch/nom_sonde.props, où nom_sonde est le nom de la sonde et arch représente le système d'exploitation.	Lors du démarrage uniquement
RawCapture 0 1	-raw -noraw	<p>Contrôle le mode de capture brute. Ce mode est généralement utilisé sur demande du service de support logiciel IBM. Le mode de capture brute est désactivé par défaut (0).</p> <p>Remarque : La capture brute peut générer une quantité importante de données. Par défaut, le fichier de capture brute peut grandir indéfiniment, bien que vous puissiez limiter la taille en utilisant la propriété MaxRawFileSize. La capture brute peut également ralentir les performances de la sonde en raison de la quantité d'activité de disque requise pour une sonde occupée.</p>	Immédiatement
RawCaptureFile chaîne	-capturefile chaîne	Spécifie le nom du fichier de capture brute. La valeur par défaut est \$OMNIHOME/var/nom_sonde.cap, où nom_sonde est le nom de la sonde.	Immédiatement
RawCaptureFile Append 0 1	-rawcapappend -norawcapappend	<p>Indique que les nouvelles données sont ajoutées au fichier de capture de données brutes existant, au lieu d'écraser les données présentes dans le fichier.</p> <p>Par défaut, le fichier est écrasé (0).</p>	Immédiatement
RegexLibrary chaîne	-regexplib chaîne	<p>Définit la bibliothèque d'expressions régulières à utiliser. Les valeurs possibles sont : NETCOOL et TRE.</p> <p>La valeur par défaut TRE permet d'utiliser la syntaxe d'expression régulière étendue dans les langues en caractères codés sur un octet et sur plusieurs octets. Ce paramètre entraîne une baisse des performances du système.</p> <p>La valeur NETCOOL est utile pour le traitement des caractères codés sur un octet et fournit des performances système optimales.</p>	Lors du démarrage uniquement

Tableau 33. Propriétés et options de ligne de commande de sonde communes. (suite)

Propriété	Option de ligne de commande	Description	Type de mise à jour
RetryConnection <i>Count entier</i>	N/D	Spécifie le nombre d'événements traités par la sonde en mode stocker-et-transmettre avant de tenter de se reconnecter au serveur ObjectServer. La valeur par défaut est 15.	Lors de la reconnexion
RetryConnection <i>TimeOut entier</i>	N/D	Spécifie le nombre de secondes pendant lesquelles la sonde traite les événements en mode stocker-et-transmettre avant de tenter de se reconnecter au serveur ObjectServer. La valeur par défaut est 30.	Lors de la reconnexion
RollSAFInterval <i>entier</i>	-rollsafinterval <i>entier</i>	Utilisé lorsque la sonde est connectée à un serveur ObjectServer, et lorsque la fonction stocker-et-transmettre circulaire est activée avec le paramètre StoreAndForward défini sur 2. Indique l'intervalle en secondes au bout duquel un fichier stocker-et-transmettre passe sur le fichier suivant dans un pool de deux fichiers utilisés pour stocker une copie des événements envoyés sur un serveur ObjectServer connecté. Pour réduire la perte d'événements lors de la reprise en ligne et la reprise par restauration, définissez l'intervalle de temps sur une valeur supérieure ou égale à la granularité du serveur ObjectServer. En cas d'échec, la sonde disposera d'une copie des événements de la dernière période de granularité, qui peuvent être rejoués sur le serveur ObjectServer de secours. La valeur par défaut est 90 secondes, soit une valeur 1,5 fois supérieure à la période de granularité de 60 secondes définie par défaut pour un serveur ObjectServer.	Immédiatement
RulesFile <i>chaîne</i>	-rulesfile <i>chaîne</i>	Spécifie le nom et l'emplacement du fichier de règles. Il peut s'agir d'un nom de fichier ou d'une adresse URI qui spécifie un fichier de règles situé sur un serveur distant accessible via HTTP. La valeur par défaut est \$OMNIHOME/probes/arch/nom_sonde.rules, où <i>nom_sonde</i> représente le nom de la sonde. Remarque : Pour des raisons de sécurité, il est conseillé de n'exécuter les sondes qui utilisent l'interface HTTP que dans un environnement de réseau privé sécurisé.	Lors de la reconnexion
SAFFileName <i>chaîne</i>	-saffilename <i>chaîne</i>	Spécifie le nom du fichier stocker-et-transmettre. La valeur par défaut est \$OMNIHOME/var/nom_sonde.store, où <i>nom_sonde</i> est le nom de la sonde. Une extension <i>.nom_serveur</i> est ajoutée automatiquement au nom du fichier, où <i>nom_serveur</i> est le nom du serveur ObjectServer cible. Un fichier stocker-et-transmettre est créé pour chaque ObjectServer cible enregistré.	Lors du démarrage uniquement
SAFPoolSize <i>entier</i>	-safpoolsize <i>entier</i>	Utilisé lorsque la sonde n'est pas connectée à un serveur ObjectServer. Indique le nombre de fichiers stocker-et-transmettre dans un pool de fichiers pouvant stocker des alertes. La valeur par défaut est 3. Le roulement sur le fichier suivant s'effectue lorsqu'un fichier atteint la taille maximale définie par la propriété MaxSAFFileSize .	Immédiatement
SecureLogin 0 1	-securelogin -nosecurelogin	Spécifie si la sonde utilise une connexion chiffrée sécurisée pour accéder au système hôte : <ul style="list-style-type: none">• 0 : la sonde n'utilise pas de connexion sécurisée chiffrée.• 1 : la sonde utilise une connexion sécurisée chiffrée. La valeur par défaut est 0. Remarque : La connexion sécurisée n'est pas disponible en mode FIPS 140-2. SSL est plus sûr que la connexion sécurisée.	Lors de la reconnexion
Server <i>chaîne</i>	-server <i>chaîne</i>	Spécifie le nom du serveur ObjectServer principal ou du serveur proxy auquel les alertes sont envoyées. La valeur par défaut est NCOMS. Si vous souhaitez que la sonde fonctionne en mode stockage et retransmission circulaire, ne spécifiez aucune définition du serveur ObjectServer virtuelle pour définir la valeur de cette propriété.	Lors de la reconnexion

Tableau 33. Propriétés et options de ligne de commande de sonde communes. (suite)

Propriété	Option de ligne de commande	Description	Type de mise à jour
ServerBackup chaîne	N/D	Spécifie le nom d'un serveur ObjectServer de secours auquel la sonde doit se connecter en cas d'échec de la connexion au serveur ObjectServer principal. Si NetworkTimeout est défini, utilisez ServerBackup pour identifier un serveur ObjectServer de secours. Si vous souhaitez que la sonde fonctionne en mode stockage et retransmission circulaire, ne spécifiez aucune définition du serveur ObjectServer virtuelle pour définir la valeur de cette propriété.	Lors de la reconnexion
SingleThreaded Comms TRUE FALSE	-singlethreadedcomms	Spécifie si un traitement multiprocesseur ou à un seul processeur est utilisé pour traiter et envoyer des alertes aux serveurs ObjectServer cibles. La valeur par défaut est FALSE, qui permet la communication multiprocesseur. Vous pouvez également utiliser la propriété SingleThreadedComms pour appliquer un ordre d'envoi des alertes aux serveurs ObjectServer. Avec le traitement multiprocesseur, les alertes sont envoyées simultanément aux serveurs ObjectServer cible. En mode à un seul processeur, l'ordre est défini par celui dans lequel les instructions registertarget sont définies dans le fichier de règles.	Lors de la reconnexion
SSLServer CommonName chaîne1,...	N/D	Si la sonde se connecte à un serveur ObjectServer via SSL et que la zone Nom commun du certificat reçu ne correspond pas au nom spécifié par la propriété Server , utilisez cette propriété pour spécifier une liste séparée par des virgules de noms communs SSL acceptables. La valeur par défaut est l'utilisation de la propriété Server .	Lors de la reconnexion
StoreAndForward entier	-saf entier	Contrôle les opérations stocker-et-transmettre. Les valeurs possibles de la propriété sont les suivantes : <ul style="list-style-type: none"> • 0 : ne pas utiliser la fonction stocker et transmettre. • 1 : utiliser la fonction stocker et transmettre héritée, qui stocke les alertes dans un fichier stocker-et-transmettre uniquement si les alertes ne peuvent pas être envoyées à un serveur ObjectServer. • 2 : utiliser la fonction stocker et transmettre circulaire, qui stocke toutes les alertes générées dans un pool de fichiers stocker-et-transmettre tournant lorsque la sonde est connectée à un serveur ObjectServer. Si la sonde est déconnectée, le comportement stocker et transmettre circulaire est similaire au comportement stocker et transmettre hérité. Par défaut, le mode stocker-et-transmettre hérité est activé (1).	Lors de la reconnexion
StoreSAFRejects 0 1	-storesafrejects -dontstoresafrejects	Spécifie si la sonde sauvegarde continuellement les enregistrements stocker-et-transmettre corrompus pour analyse. Si cette option est définie sur 1, les enregistrements stocker-et-transmettre sont continuellement sauvegardés. La valeur par défaut est 0. Utilisez cette propriété avec la propriété KeepLastBrokenSAF .	Immédiatement
N/D	-version	Affiche les informations de version et ferme la fenêtre.	Lors du démarrage uniquement

Concepts associés:

«Utilisation des propriétés de sonde par rapport aux options de ligne de commande des sondes», à la page 7

Chaque propriété de sonde a une option de ligne de commande correspondante.

«Types de propriétés de sonde», à la page 7

Les propriétés de sonde peuvent être divisées en deux catégories : les propriétés communes et les propriétés spécifiques à la sonde.

«Mode stocker-et-transmettre pour les sondes», à la page 12

L'exécution des sondes peut continuer si le serveur ObjectServer cible est arrêté.

Pendant cette période, la sonde passe en mode *stocker*. Elle repasse en mode *transmettre* lorsque le serveur ObjectServer fonctionne à nouveau.

«Mode de capture brute pour les sondes», à la page 16

Le mode capture brute vous permet de sauvegarder l'intégralité d'un flux de données d'événement acquis par une sonde dans un fichier, sans traitement par le fichier de règles. Cela peut être utile à des fins d'audit, d'enregistrement ou de débogage du fonctionnement d'une sonde.

«Mode sécurisé pour les sondes», à la page 17

Vous pouvez exécuter le serveur ObjectServer en mode sécurisé. Lorsque vous démarrez le serveur ObjectServer à l'aide de l'option de ligne de commande `-secure`, il authentifie les connexions de sonde, de passerelle et de serveur proxy en demandant un nom d'utilisateur et un mot de passe.

«Mode de reprise en ligne d'égal à égal pour les sondes», à la page 17

Deux instances d'une sonde peuvent être exécutées simultanément dans une relation de reprise en ligne d'égal à égal. Une instance est désignée comme maître. L'autre agit comme esclave et est en mode secours automatique. En cas d'échec de l'instance maître, l'instance esclave est activée.

Tâches associées:

Chapitre 5, «Administration à distance des sondes», à la page 105

Vous pouvez utiliser une interface HTTP qui s'exécute sur un serveur HTTP ou HTTPS contenu dans la bibliothèque C de sondes standard (libOpl) pour surveiller et gérer à distance des sondes. Cette fonctionnalité peut être utilisée sur toutes les sondes qui sont compatibles avec Tivoli Netcool/OMNIbus version 8.1. Un identificateur URI commun est fourni. Il contient des ressources pour effectuer des tâches d'administration.

«Relecture du fichier de règles», à la page 66

Etant donné que la sonde ne lit le fichier de règles qu'au démarrage, vous devez la forcer à relire les règles chaque fois que vous y apportez des modifications. La sonde traite la demande de relecture uniquement sur réception d'un nouvel événement. Si la sonde est inactive ou occupée par le traitement d'un événement, elle relit le fichier de règles uniquement lors de la réception d'un nouvel événement.

Référence associée:

«Traitement multiprocessus des données d'alerte», à la page 55

Lorsqu'un fichier de règles de sonde est traité, le traitement multiprocessus est utilisé par défaut pour appliquer les règles de sonde aux données d'événements brutes obtenues sur la source d'événements, et pour envoyer les alertes générées aux serveurs ObjectServer enregistrés. Notez que ce traitement multiprocessus est différent de la capture d'événements mutliprocessus ou monoprocesus implémentée sur certaines classes de sondes.

«Opérations de table de consultation», à la page 45

Les tables de consultation permettent d'ajouter des informations supplémentaires à un événement. Une table de consultation est constituée d'une liste de clés et de valeurs.

Chapitre 7. Netcool MIB Manager

Netcool MIB Manager est une application d'interface graphique IBM® basée sur Eclipse. Vous pouvez l'utiliser pour analyser les fichiers MIB (Management Information Base) SNMP (Simple Network Management Protocol) à partir desquels vous pouvez générer des fichiers de règles Netcool. Cette application est destinée à remplacer l'utilitaire **mib2rules**.

Tivoli Netcool/OMNIBus inclut également un Netcool MIB Manager utilitaire de ligne de commande (**nco_mibmanager_batch**) que vous pouvez utiliser pour émettre des commandes manuellement ou à l'aide d'un programme.

Netcool MIB Manager propose les fonctions suivantes :

- Il importe des fichiers MIB SNMP et résout les dépendances MIB afin de générer une arborescence d'identificateurs d'objet (OID).
- Il peut exporter tout ou partie des objets SNMP importés dans des fichiers de règles Netcool et d'autres formats de fichier.
- Il inclut les MIB de base et les MIB RFC le plus couramment requises par d'autres fichiers MIB.
- Il inclut des définitions de périphérique afin de permettre le groupement centré sur les périphériques de modules MIB importés.
- Il peut filtrer des modules MIB et l'arbre d'ID objet et y effectuer des recherches par nom d'objet ou ID objet.
- Il peut générer des interceptions SNMP à envoyer à l'analyse SNMP (**nco_p_mttrapd**) ou à d'autres agents SNMP.
- Il comporte une interface de ligne de commande que vous pouvez utiliser pour importer des données MIB et exporter des fichiers de règles. Vous pouvez exécuter manuellement des commandes ou les appeler à l'aide d'un programme.

Démarrage de MIB Manager

Démarrez l'interface graphique de MIB Manager à partir de la ligne de commande. Il n'est pas conçu pour être exécuté sous le contrôle de processus ou en tant que service Windows.

Pourquoi et quand exécuter cette tâche

Voir la documentation sur les systèmes d'exploitation pris en charge pour plus d'informations sur les conditions requises pour exécuter MIB Manager sur les systèmes d'exploitation AIX et Solaris.

Ces informations sur les conditions requises sont disponibles dans le *Guide d'installation et de déploiement de Tivoli Netcool/OMNIBus*.

UNIX **Linux** Avant de démarrer MIB Manager, vérifiez que la variable d'environnement `$DISPLAY` est définie sur un serveur X11 en fonctionnement.

Procédure

1. Utilisez la commande suivante pour démarrer l'interface graphique de MIB Manager :

`NCHOME/omnibus/bin/nco_mibmanager`

2. Utilisez la commande suivante pour démarrer l'utilitaire de ligne de commande de MIB Manager :

`NCHOME/omnibus/bin/nco_mibmanager_batch [options]`

Référence associée:

«Options de ligne de commande de MIB Manager»

MIB Manager comporte un utilitaire de ligne de commande que vous pouvez utiliser pour importer des données MIB et exporter des fichiers de règles. Vous pouvez exécuter manuellement des commandes ou les appeler à l'aide d'un programme.

Options de ligne de commande de MIB Manager

MIB Manager comporte un utilitaire de ligne de commande que vous pouvez utiliser pour importer des données MIB et exporter des fichiers de règles. Vous pouvez exécuter manuellement des commandes ou les appeler à l'aide d'un programme.

Pour démarrer l'utilitaire de ligne de commande, utilisez la commande suivante :

- **UNIX** **Linux** `$NCHOME/omnibus/bin/nco_mibmanager_batch [options]`
- **Windows** `%NCHOME%\omnibus\bin\nco_mibmanager_batch [options]`

Sur les systèmes d'exploitation Windows, la sortie de MIB Manager s'affiche sur une nouvelle console.

Le tableau suivant décrit les options de ligne de commande disponibles. Les options `-exportmibs`, `-exportobjects` et `-exportoids` peuvent analyser les expressions régulières qui sont conformes avec le moteur de modèle Java.

Windows Utilisez des chemin de répertoire complets pour indiquer les répertoires. Par exemple : `nco_mibmanager_batch -importdir "C:\anydir\mibs"`

Tableau 34. Options de ligne de commande de MIB Manager

Option de ligne de commande	Description
<code>-clean</code>	Utilisez cette option lorsque vous exécutez pour la première fois l'utilitaire de ligne de commande MIB Manager, si vous avez utilisé auparavant l'utilitaire graphique MIB Manager. Si vous n'utilisez pas cette option, une erreur s'affiche.
<code>-exportdir chaîne</code>	Utilisez cette option pour indiquer le répertoire vers lequel les fichiers de règles générés sont exportés. Placez le chemin de répertoire entre guillemets (""). Exemple : <code>nco_mibmanager_batch -exportdir "/home/user/export_cmd" -filetype csv</code>

Tableau 34. Options de ligne de commande de MIB Manager (suite)

Option de ligne de commande	Description
-exportmibs <i>chaîne</i>	<p>Utilisez cette option pour indiquer le nom de la base MIB à exporter. Vous pouvez utiliser des expressions régulières pour indiquer plusieurs bases MIB.</p> <p>Par exemple, la commande suivante exporte les objets à partir d'une base MIB appelée «RMON-MIB» :</p> <pre>nco_mibmanager_batch -exportmibs RMON-MIB -filetype csv</pre> <p>La commande suivante exporte les objets de toutes les bases MIB dont le nom contient «MON» :</p> <pre>nco_mibmanager_batch -exportmibs .*MON.* -filetype csv</pre>
-exportnewdirectory	<p>Utilisez cette option pour demander que les fichiers exportés soient écrits dans un nouveau répertoire, au sein du répertoire d'exportation existant. Le nouveau sous-répertoire est nommé automatiquement avec un horodatage. Cette option permet d'éviter que les fichiers de règles existants soient écrasés.</p> <p>Exemple :</p> <pre>nco_mibmanager_batch -exportdir "/home/user/export_cmd" -exportnewdirectory -filetype csv</pre>
-exportobjects <i>chaîne</i>	<p>Utilisez cette option pour indiquer le nom de l'objet MIB à exporter. Vous pouvez utiliser des expressions régulières pour indiquer plusieurs objets.</p> <p>Par exemple, la commande suivante exporte tous les objets nommés «coldStart» :</p> <pre>nco_mibmanager_batch -exportobjects coldStart -filetype csv</pre> <p>La commande suivante exporte tous les objets dont le nom commence par «cold» :</p> <pre>nco_mibmanager_batch -exportobjects cold.* -filetype csv</pre>

Tableau 34. Options de ligne de commande de MIB Manager (suite)

Option de ligne de commande	Description
<p><code>-exportoids</code> chaîne</p>	<p>Utilisez cette option pour indiquer l'ID objet (OID) à exporter. Vous pouvez utiliser des expressions régulières pour indiquer plusieurs ID objet.</p> <p>Par exemple, la commande suivante exporte tous les objets avec l'ID objet «1.3.6.1.6.3.1.1.5.1»:</p> <pre>nco_mibmanager_batch -exportoids 1.3.6.1.6.3.1.1.5.1 -filetype csv</pre> <p>La commande suivante exporte tous les objets dont l'ID objet commence par «1.3.6.1.6.3.1.1.5»:</p> <pre>nco_mibmanager_batch -exportoids 1.3.6.1.6.3.1.1.5.* -filetype csv</pre> <p>La commande suivante exporte tous les objets avec l'ID objet «1.3.6.1.6.3.1.1.5.1» ou «1.3.6.1.6.3.1.1.5.2» :</p> <pre>nco_mibmanager_batch -exportoids 1.3.6.1.6.3.1.1.5.1 1.3.6.1.6.3.1.1.5.2 -filetype csv</pre>
<p><code>-exportscope</code> ALL TRAPS OBJECTS</p>	<p>Utilisez cette option pour indiquer le type des objets MIB exportés. Cette option utilise les valeurs suivantes :</p> <ul style="list-style-type: none"> • ALL : Tous les objets MIB sont exportés. • TRAPS : Seules les interceptions sont exportées. • OBJECTS : Seuls les objets sont exportés. <p>Exemple :</p> <pre>nco_mibmanager_batch -exportdir "/home/user/export_cmd" -exportnewdirectory -filetype csv -exportscope ALL</pre>

Tableau 34. Options de ligne de commande de MIB Manager (suite)

Option de ligne de commande	Description
<p>-filetype</p> <p>csv csv_trap_objects html_with_frames html_without_frames lookup ncil nckl_1_1 nckl_3_0 standalone text tivoli_universal_agent</p>	<p>Utilisez cette option pour indiquer le format de fichier dans lequel les fichiers générés sont exportés. Cette option est obligatoire pour les opérations d'exportation.</p> <p>Les paramètres disponibles sont les suivants :</p> <ul style="list-style-type: none"> • csv : Toutes les données objet sont écrites dans des fichiers CSV. • csv_trap_objects : Toutes les données d'interception et les données d'objet de liaisons de variable associées sont écrites dans un fichier CSV unique. • html_with_frames : Crée une page Web adaptée à la publication. • html_without_frames : Crée un fichier unique appelé oids.html qui contient la liste de tous les objets sélectionnés pour la sortie. • lookup : Place toutes les valeurs spécifiées dans une liste à tabulations pour inclusion dans tout fichier de règles. • ncil : Crée un fichier d'inclusion distinct pour chaque entreprise et génère un fichier unique pour les interceptions génériques. • nckl_1_1 : Crée un fichier adapté à une utilisation avec Netcool Knowledge Library V1.1. • nckl_3_0 : Crée un fichier adapté à une utilisation avec Knowledge Library V3.x. • standalone : Crée un fichier de règles unique qui contient toutes les interceptions sélectionnées. • text : Crée un fichier texte distinct pour chaque ID objet. • tivoli_universal_agent : Crée les fichiers .mdl et trapcnfg.
<p>-help</p>	<p>Utilisez cette option pour afficher des informations d'aide sur les options de ligne de commande.</p>
<p>-importdir <i>chaîne</i></p>	<p>Utilisez cette option pour indiquer le répertoire depuis lequel les fichiers MIB sont importés. Placez le chemin de répertoire entre guillemets (").</p> <p>Exemple :</p> <pre>nco_mibmanager_batch -importdir "/home/user/mibs"</pre>
<p>-importdirtraverse</p>	<p>Utilisez cette option pour indiquer que MIB Manager explore les sous-répertoires lorsqu'il recherche les fichiers MIB à importer.</p> <p>Exemple :</p> <pre>nco_mibmanager_batch -importdir "/home/user/mibs" -importdirtraverse</pre>

Tableau 34. Options de ligne de commande de MIB Manager (suite)

Option de ligne de commande	Description
-messagelevel ERROR WARN INFO DEBUG VERBOSE NONE	Utilisez cette option pour indiquer le niveau de consignation de messages. Cette option utilise les valeurs suivantes : <ul style="list-style-type: none"> • ERROR : Seuls les messages d'erreur sont consignés. • WARN : Les messages d'avertissement et d'erreur sont consignés. • INFO : Les messages d'information, d'avertissement et critiques sont consignés. • DEBUG : Tous les messages sont consignés. • VERBOSE : Les messages en mode prolix incluent l'état des variables, des tableaux et des hachages au fur et à mesure que leurs valeurs et leur état changent. • NONE : Aucun message n'est consigné.
-messagelog chaîne	Utilisez cette option pour indiquer le répertoire dans lequel le fichier journal des messages, <code>mibmanager.log</code> , est écrit. Placez le chemin de répertoire entre guillemets (""). La valeur par défaut est <code>NCHOME/omnibus/log</code> .
-searchpath chaîne	Utilisez cette option pour indiquer une liste de répertoire séparés par des points-virgules que MIB Manager peut explorer pour résoudre des dépendances de MIB. Placez le chemin de répertoire entre guillemets (""). Exemple : <code>nco_mibmanager_batch -importdir "/home/user/mibs" -searchpath "/home/user/mib_dependencies"</code> La commande suivante recherche les dépendances dans 3 répertoires spécifiques : <code>nco_mibmanager_batch -importdir "/home/user/mibs" -searchpath "/home/user/mib_dependencies;/home/user/ mib_other;/home/user/mib_test"</code>

Utilisation de Netcool MIB Manager

Vous pouvez utiliser MIB Manager pour analyser des modules MIB et créer des fichiers de règles Netcool.

MIB Manager inclut un ensemble de fichiers MIB de base et RFC, dans le répertoire `NCHOME/omnibus/var/mibmanager/mibs`. Ces bases MIB sont fournies pour vous permettre d'importer des nouvelles bases MIB. MIB Manager utilise automatiquement ces MIB groupées pour résoudre les dépendances au cours de l'importation de nouvelles MIB. Les modules MIB groupés sont visibles dans la fenêtre Modules MIB lorsque vous démarrez MIB Manager pour la première fois.

MIB Manager comprend également des fichiers de données XML qui représentent une importation des MIB de base groupées. MIB Manager utilise ces fichiers XML pour stocker les données MIB qu'il importe à partir de fichiers MIB de texte brut.

Ajoutez vos bases MIB à un répertoire accessible à partir du serveur à partir duquel est lancé MIB Manager. Toutes les MIB basées sur des normes doivent être

installées dans un répertoire distinct qui peut être placé dans le chemin de recherche pour faciliter la résolution des dépendances. Par défaut, les fichiers MIB SNMP de base sont installés dans le répertoire `NCHOME/omnibus/var/mibmanager/mibs/base` et tous les fichiers MIB RFC sont installés dans le répertoire `NCHOME/omnibus/var/mibmanager/mibs/rfc`.

Fenêtre Netcool MIB Manager

Lorsque vous lancez MIB Manager, les éléments suivants sont affichés dans la fenêtre MIB Manager initiale. La figure suivante présente ces éléments.

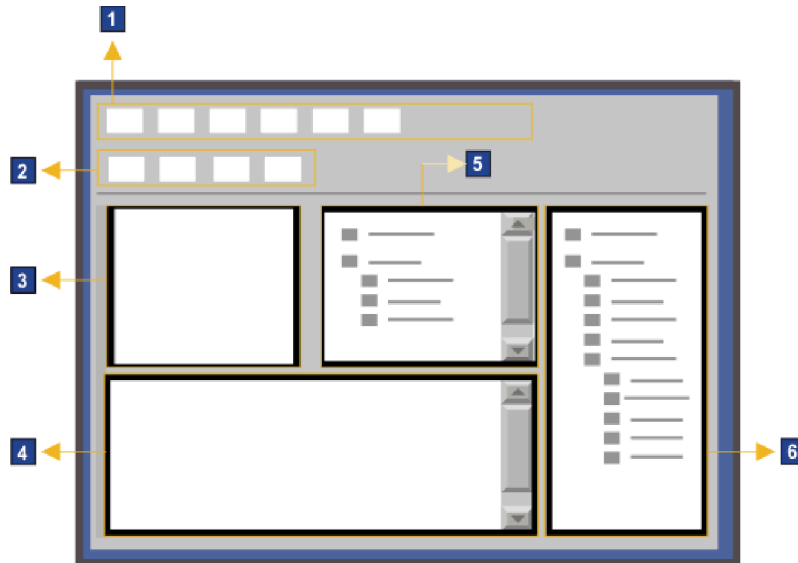


Figure 4. Fenêtre initiale MIB Manager

1 Barre d'outils

Permet d'accéder aux fonctions MIB Manager les plus couramment utilisées.

2 Perspectives

Vous pouvez afficher des données MIB à l'aide de l'une des deux perspectives, **MIB Manager** ou **mib2rules**. **MIB Manager** est la perspective par défaut. La perspective **mib2rules** se concentre sur les identificateurs d'objet et est destinée aux utilisateurs familiarisés avec l'utilitaire **mib2rules**.

3 Vue Unité

Utilisez cette vue pour créer, mettre à jour et supprimer des unités MIB Manager.

4 Vue Détails et vue Console

La vue Détails affiche des informations détaillées sur les éléments MIB sélectionnés. La vue Console affiche tous les messages de journal générés par MIB Manager pendant l'exécution de l'instance en cours.

5 Modules MIB

Utilisez cette vue pour rechercher et supprimer des modules MIB et pour rechercher des valeurs MIB spécifiques MIB.

6 Vue Arborescence d'ID objet

Cette vue fournit une représentation de tous les objets MIB qui se trouvent dans les modules MIB importés.


Vue Modules MIB


La vue Modules MIB fournit une représentation de tous les modules MIB importés et vous permet de rechercher des modules MIB, de supprimer des modules MIB et de rechercher des valeurs MIB spécifiques.

Vous pouvez naviguer dans la vue Modules MIB en développant et en réduisant les noeuds dans l'arborescence MIB. Les branches du niveau supérieur de l'arborescence MIB contiennent les noms des modules MIB, et dans chaque branche de module MIB se trouvent les autres éléments qui composent la base MIB.

Vous pouvez filtrer les données dans la vue Arborescence d'ID objet en sélectionnant l'une des options suivantes dans la liste déroulante **Vue** :

- **Tous**
- **Modules**
- **Enumérations**
- **Conventions textuelles**
- **Syntaxes**
- **Importations**
- **Exportations**
- **Colonnes**
- **Descriptions**

Pour rechercher une valeur spécifique dans la vue Modules MIB, saisissez votre texte dans la zone **Recherche**, puis appuyez sur **Entrée** ou cliquez sur  (Recherche).

MIB Manager commence la recherche à partir de l'élément sélectionné ou, si aucun élément n'est sélectionné, à partir du premier élément de la liste. Pour rechercher l'élément correspondant suivant dans la liste, continuez à cliquer sur  (Recherche).

Lorsqu'un élément est trouvé, il est automatiquement mis en évidence et affiché dans la vue Modules MIB.

Dans la zone **Recherche**, entrez une expression régulière valide ou entrez du texte en clair pour représenter un nom d'objet complet ou partiel.

Pour supprimer un ou plusieurs modules MIB, cliquez avec le bouton droit sur le ou les modules MIB sélectionnés, puis, dans le menu contextuel, sélectionnez **Supprimer**. Le module MIB sélectionné est supprimé de MIB Manager. Cependant, les fichiers source MIB d'origine, fournis pour l'importation MIB, ne sont pas supprimés.

Remarque : Vous devez uniquement supprimer un module MIB s'il contient des erreurs et qu'il existe une nouvelle version de remplacement du module MIB. La suppression de modules MIB à l'aide de cette méthode peut entraîner une absence de dépendances MIB. Les fichiers de règles incomplets peuvent également être générés lors du processus d'exportation avec des messages d'avertissement ou d'erreur dans le fichier journal.

Le tableau suivant répertorie les icônes de la vue Modules MIB et décrit ce que chacune représente.

Tableau 35. Icônes de la vue Modules MIB et les objets qu'elles représentent





























Icône	Représentation
	EXPORTS
	EXPORT
	IMPORTS
	IMPORT
	OBJECT IDENTIFIER, OBJECT-IDENTITY
	AGENT-CAPABILITIES
	MODULE-COMPLIANCE
	MODULE, MODULE-IDENTITY
	OBJECT-GROUP
	NOTIFICATION-GROUP
	TRAP-TYPE (interception v1)
	NOTIFICATION-TYPE (interception v2)
	OBJECT-TYPE générique
	COUNTER
	COUNTER32
	COUNTER64
	GAUGE
	GAUGE32
	INTEGER
	INTEGER32
	Adresse IP
	Adresse réseau
	OID

Tableau 35. Icônes de la vue Modules MIB et les objets qu'elles représentent (suite)


Icône	Représentation
	OCTET STRING
	SEQUENCE (table)
	SEQUENCE (Entrée, ligne de table)
	TIMETICKS
	TEXTUAL-CONVENTION


Vue Arborescence d'ID objet

La vue Arborescence d'ID objet fournit une représentation de tous les objets MIB qui se trouvent dans les modules MIB importés.

Vous pouvez filtrer les données dans la vue Arborescence des ID objet en sélectionnant l'une des options suivantes dans la liste déroulante **Vue** :

- **Tous**
- **Interceptions/Notifications**
- **Objets**
- **Modules**

Pour rechercher une valeur spécifique dans la vue Arborescence des ID objet, saisissez votre texte dans la zone **Recherche**, puis appuyez sur **Entrée** ou cliquez sur  (Recherche).

MIB Manager commence la recherche à partir de l'élément sélectionné ou, si aucun élément n'est sélectionné, à partir du premier élément de la liste. Pour rechercher l'élément correspondant suivant dans la liste, continuez à cliquer sur  (Recherche).

Lorsqu'un élément est trouvé, il est automatiquement mis en évidence et affiché dans la vue Arborescence d'ID objet.

Dans la zone **Recherche**, entrez une expression régulière valide ou entrez du texte en clair pour représenter un nom d'objet complet ou partiel ou un ID objet. Le chiffre affiché sur chaque noeud de branche est le dernier chiffre de l'ID objet pour cet objet précis.

Le tableau suivant répertorie les icônes de la vue Arborescence d'ID objet et décrit ce que chacune représente.

Tableau 36. Icônes de l'Arborescence d'ID objet et les objets associés qu'elles représentent

























Icône	Représentation
	OBJECT IDENTIFIER, OBJECT-IDENTITY
	AGENT-CAPABILITIES

Tableau 36. Icônes de l'Arborescence d'ID objet et les objets associés qu'elles représentent (suite)

Icône	Représentation
	MODULE-COMPLIANCE
	MODULE, MODULE-IDENTITY
	OBJECT-GROUP
	NOTIFICATION-GROUP
	TRAP-TYPE (message d'alerte v1)
	NOTIFICATION-TYPE (message d'alerte v2)
	OBJECT-TYPE générique
	COUNTER
	COUNTER32
	COUNTER64
	GAUGE
	GAUGE32
	INTEGER
	INTEGER32
	Adresse IP
	Adresse réseau
	OID
	OCTET STRING
	SEQUENCE (table)
	SEQUENCE (Entrée, ligne de table)
	TIMETICKS
	TEXTUAL-CONVENTION

Importation de données MIB

Comment importer des données MIB SNMP dans MIB Manager.


Avant de commencer

Dans la fenêtre Préférences , spécifiez les emplacements pour les MIB qui sont régulièrement utilisées par les bases MIB de fournisseurs. Vérifiez que vos MIB sont situées dans des répertoires spécifiques d'un fournisseur et que tous les numéros d'équipements associés sont situés dans des sous-répertoires. Assurez-vous également que ces répertoires ne contiennent aucun fichier texte autre qu'une MIB. La plupart des fichiers MIB sont installés dans le répertoire *NCHOME/omnibus/var/mibmanager/mibs/base* et tous les fichiers MIB RFC sont installés dans le répertoire *NCHOME/omnibus/var/mibmanager/mibs/rfc* .




Les MIB que vous importez peuvent être dépendantes d'autres bases MIB. Pour vous assurer que MIB Manager peut rechercher les bases MIB dépendantes lors de l'exécution de l'importation, définissez le chemin de recherche vers les emplacements des fichiers MIB dépendants.

Procédure

Pour importer des données MIB :

1. Cliquez sur  **Importer** ou, dans le menu, cliquez sur **Fichier > Importer**.
2. Dans la zone **Répertoire**, indiquez l'emplacement des modules MIB que vous souhaitez importer.
3. Pour que MIB Manager analyse toutes les MIB trouvées dans les sous-répertoires disponibles, cochez la case **Sous-répertoires transversaux**.
4. Cliquez sur **Importer**.

La fenêtre Etat de l'importation s'affiche pendant que les modules MIB sont analysés. La partie gauche de cette fenêtre affiche le nombre de types d'objet MIB différents au fur et à mesure que chacun d'eux est découvert lors de l'analyse syntaxique des fichiers MIB importés. Une fois les MIB analysées, MIB Manager passe en revue chaque module MIB et analyse tous les objets MIB qu'il localise. Une fois que tous les objets sont analysés, MIB Manager passe en revue chaque MIB et traite les instructions d'importation afin de localiser tous les modules MIB référencés. Si des modules MIB référencés ne peuvent pas être localisés dans des modules MIB préalablement importés, MIB Manager effectue la recherche dans les répertoires indiqués dans le chemin de recherche. La partie droite de la fenêtre Etat de l'importation affiche des icônes qui indiquent l'état des instructions d'importation figurant dans les MIB importées, comme illustré dans le tableau suivant :

Icône	Description
	Indique que l'instruction d'importation n'a pas été résolue.
	Indique que l'instruction d'importation a été résolue.
	Indique que le module MIB référencé dans l'instruction d'importation n'a pas pu être résolu.

Vous pouvez développer l'arborescence de l'instruction d'importation pour afficher quels modules MIB utilisent des instructions d'importation pour faire référence à d'autres modules MIB.

Importation terminée s'affiche au début de la fenêtre d'état lorsque tous les fichiers MIB importés ont été traités. Vous pouvez afficher tous les messages d'état générés lors de l'importation en cliquant sur **Afficher l'historique de l'état** et en utilisant la liste déroulante **Vue** pour sélectionner une option de filtre.

5. Cliquez sur **Ignorer** pour fermer la fenêtre Etat de l'importation.

Résultats

La vue Modules MIB est automatiquement régénérée pour afficher les noms de module MIB que vous venez d'importer. La vue Arborescence d'ID objet s'actualise automatiquement pour afficher tous les objets MIB que vous venez d'importer. Pour reconstituer l'arborescence, MIB Manager recherche dans les noeuds parent de l'arborescence (*iso*, *ccitt* et *joint-iso-ccitt*) de manière récursive les noeuds enfant jusqu'à ce qu'il trouve un noeud enfant qui n'a pas d'enfant associé. Si aucune erreur n'est détectée, une arborescence MIB entièrement remplie s'affiche dans la vue Arborescence d'ID objet laquelle contient les objets analysés.

Cliquez sur un objet dans la vue Arborescence des ID objet pour afficher automatiquement l'objet dans la vue Modules MIB. La vue Détails affiche des informations détaillées sur l'objet sélectionné.

Les modules MIB importés sont stockés dans un fichier xml qui représente la structure de ce module MIB. Les fichiers xml sont situés dans le répertoire *NCHOME/omnibus/var/mibmanager/data* .

Si des instructions d'importation non résolues sont détectées, prenez contact avec le fournisseur responsable de la ou des MIB manquantes, ajoutez le ou les fichiers manquants au chemin de recherche et répétez l'importation.

Remarque :

Les fournisseurs indiquent parfois des noms d'objet en double, ou un nom d'objet identique à un nom d'objet spécifié dans une RFC.

Par exemple, un objet nommé *system* est défini dans le module *SNMPv2-MI* avec l'ID *1.3.6.1.2.1.1* et un autre objet nommé *system* est défini dans le module *WINDOWS-NT-PERFORMANCE*, avec l'ID *1.3.6.1.4.1.311.1.1.3.1.1.23*. En outre, un objet nommé *sysDescr* est défini en tant que *system.1* dans le module *SNMPv2-MIB* et un autre objet nommé *sysFileReadOperationsPerSec* est défini en tant que *system.1* dans le module *WINDOWS-NT-PERFORMANCE*. Dans ce cas, il est difficile de déterminer à quel objet parent les objets enfant sont associés.

Lorsqu'un objet avec un nom de parent en double est localisé, MIB Manager tente de localiser le parent qui est défini dans le même module MIB. Dans l'exemple précédent, il s'agit de *WINDOWS-NT-PERFORMANCE*. Si aucun des deux parents n'est défini dans le même module MIB, MIB Manager recherche alors si le parent est défini dans d'autres modules MIB dont l'objet est dépendant. MIB Manager identifie également d'autres modules MIB qui font référence au parent et sélectionne le nom de parent le plus couramment référencé. Toutefois, cela peut générer une arborescence MIB incorrectement remplie et des ID objet incorrectement calculés. Par conséquent, chaque fois que MIB Manager détecte un nom d'objet en double, il consigne un avertissement et enregistre les mesures prises pour résoudre la duplication dans un fichier de débogage.

Référence associée:

«Définition des préférences de répertoire», à la page 160
Utilisez les préférences de répertoire pour indiquer les répertoires utilisés pour le stockage des MIB importées, des données de MIB et des fichiers de règles exportés.


«Définition des préférences de recherche», à la page 163
Cette section explique comment définir les préférences de MIB Manager qui sont utilisées pour rechercher les fichiers MIB.

Exportation des données MIB

Exporter les données MIB SNMP à partir de MIB Manager vers différents formats.

Procédure

Pour exporter des données MIB :

1. Cliquez sur  **Exporter** ou, dans le menu, cliquez sur **Fichier > Exporter**.
2. Dans la zone **Répertoire** , indiquez le répertoire de destination pour les fichiers exportés.
3. Dans la zone **Type de fichier**, spécifiez un type de fichier pour les fichiers exportés.

Vous devez utiliser Netcool Knowledge Library (NCKL) chaque fois que cela est possible, car les fichiers de règles NCKL sont basés sur la fonctionnalité du périphérique. Lorsque vous utilisez NCKL, vous devez spécifier le type de fichier **Netcool Knowledge Library version 3.x**. Ceci permet aux fichiers générés d'être ajoutés aux installations de Netcool Include Library existantes.

Les options suivantes sont disponibles dans la liste déroulante **Type de fichier** :

Type de fichier	Description
CSV	Sélectionnez cette option pour indiquer que toutes les données d'objet sont chargées dans des fichiers CSV afin de pouvoir être visualisées à l'aide d'une application de tableur.
Objets d'interception CSV	Sélectionnez cette option pour indiquer que toutes les données d'interception et les données d'objet de liaisons de variable associées sont chargées dans un fichier CSV unique afin de pouvoir être visualisées à l'aide d'une application de tableur.

Type de fichier	Description
HTML avec cadres	<p>Sélectionnez cette option pour créer une page Web adaptée à la publication.</p> <p>L'affichage de l'arborescence MIB sélectionnée apparaît sur le côté gauche de la page Web générée. Vous pouvez naviguer dans la vue d'arborescence MIB en développant et en réduisant les noeuds dans l'arborescence MIB. Lorsqu'un élément est sélectionné dans l'arborescence MIB, les informations associées s'affichent dans la partie droite de la page Web.</p> <p>Ce format peut souvent créer un grand nombre de fichiers d'exportation, chacun contenant un fichier HTML pour chaque objet répertorié dans l'arborescence MIB. Le répertoire cible peut être compressé et situé sous n'importe quel serveur Web, ou les pages peuvent être visualisées directement à partir d'un disque via l'ouverture du fichier <code>index.html</code> créé dans le répertoire d'exportation.</p> <p>Si vous souhaitez utiliser des fichiers HTML individuels à l'aide d'un navigateur Web, exportez vers HTML avec cadres, puis utilisez les fichiers du sous-répertoire données situé dans le répertoire d'exportation.</p>
HTML sans cadres	<p>Sélectionnez cette option pour créer un fichier unique nommé <code>oids.html</code>. Ce fichier va contenir une liste de tous les objets sélectionnés pour la sortie, chacun séparé par des règles horizontales (à l'aide de la balise <code><HR></code>). Chaque ID objet est encadré par une balise d'ancrage de sorte que lorsqu'il est chargé, le navigateur défile automatiquement vers un ID objet qui inclut un symbole dièse (#) dans l'URL, par exemple : <code>file:c:/oids.html#1.3.6.1.4.1.9</code>.</p> <p>Ce fichier peut devenir très volumineux si vous exportez un grand nombre d'objets. Par conséquent, cette présentation devrait uniquement être utilisée pour l'exportation d'un petit nombre d'interceptions.</p>
Netcool Lookup Table	<p>Sélectionnez cette option pour générer toutes les valeurs spécifiées dans une liste à onglets, avec les ID objet affichés dans la colonne de gauche et les noms d'objet affichés dans la colonne de droite. Approprié pour une inclusion dans n'importe quel fichier de règles.</p>

Type de fichier	Description
Netcool Include Library	Sélectionnez cette option pour créer des fichiers d'inclusion distincts pour chaque entreprise et pour générer un fichier unique pour les interceptions génériques (c'est-à-dire pour toutes les interceptions qui ne sont pas situées dans la sous-arborescence de l'entreprise). Il est supposé que les fichiers d'inclusion sont insérés dans des fichiers snmp.rules existants, soit ceux précédemment créés par MIB Manager, soit ceux créés par la bibliothèque Netcool Include Library (NCIL).
Netcool Knowledge Library version 1.1	Sélectionnez cette option pour générer des fichiers depuis MIB Manager. Les fichiers seront inclus dans une implémentation Netcool Knowledge Library existante.
Netcool Knowledge Library version 3.x	Sélectionnez cette option pour générer des fichiers depuis MIB Manager. Les fichiers seront inclus dans une implémentation Netcool Knowledge Library existante.
Règles autonomes	Sélectionnez cette option pour créer un fichier de règles unique qui contient toutes les interceptions sélectionnées. Vous pouvez spécifier ce fichier dans le fichier de propriétés de n'importe quelle sonde d'interception SNMP Tivoli Netcool/OMNIbus.
Fichiers texte individuels	<p>Le format des fichiers texte individuels est utilisé pour le menu contextuel de la liste d'événements Netcool.</p> <p>Lorsque vous effectuez une exportation vers des fichiers de texte individuels, chaque ID objet est généré vers un fichier texte distinct, par exemple 1.3.6.1.4.1.9.1.1.txt. Un outil externe peut ensuite être ajouté au menu contextuel qui utilise nco_message pour afficher toutes les informations d'objet pour un événement dont l'ID objet est spécifié dans une zone, par exemple : cat 1.3.6.1.4.1.9.1.1.txt nco_message stdin title "trap info".</p>

Type de fichier	Description
Tivoli Universal Agent	<p>Le format Tivoli Universal Agent exporte des fichiers .mdl et trapcnfg en fonction du contenu sélectionné.</p> <p>Plusieurs sous-arborescences peuvent être sélectionnées lorsque ce format est sélectionné pour l'exportation de fichiers. Si plusieurs sous-arborescences sont sélectionnées, les fichiers .mdl et trapcnfg sont créés pour tous les objets situés sous un élément sélectionné dans l'arborescence MIB.</p> <p>Remarque : Les trois premières lettres du nom d'application, à la suite de la balise //APPL dans le fichier .mdl, doivent être uniques au sein d'une entreprise. MIB Manager tente de se conformer à cette exigence, mais il ne peut pas reconnaître tous les autres agents qui peuvent être en cours d'exécution. Vous devrez peut-être modifier le fichier .mdl pour garantir que le nom d'application est unique dans l'entreprise.</p>

4. Dans la zone **Portée**, utilisez la liste déroulante pour indiquer si MIB Manager génère uniquement des interceptions, uniquement des objets ou tous les ID objet.

Les options disponibles dépendent du type de fichier que vous exportez.

5. Pour limiter le nombre d'objets inclus dans le fichier généré, cochez la case **Sous-arborescence(s) sélectionnée(s) uniquement**.
6. Cliquez sur **Exporter** pour exporter les fichiers dans le répertoire de destination.

Référence associée:

«Définition des préférences d'exportation», à la page 161

Cette section décrit comment définir vos préférences d'exportation de fichier de règles.

Edition des messages d'alerte SNMP

Comment éditer des messages d'alerte SNMP.

Pourquoi et quand exécuter cette tâche

Dans la vue Détails, vous pouvez éditer les zones suivantes d'un message d'alerte SNMP :

- @Severity
- @Type
- @ExpireTime
- CODEBLOCK

Remarque : SNMP ne fournit pas de mécanisme standard pour définir les informations de définition (@Type) ou de suppression (@Severity). Vous devez indiquer ces informations manuellement.

Dans la vue Détails, cliquez deux fois sur une valeur de zone pour la rendre modifiable. Après avoir modifié une zone, cliquez sur ✓ pour enregistrer la modification ou cliquez sur ✗ pour annuler la modification.

Procédure

Pour éditer un message d'alerte SNMP :

1. Dans la vue Modules MIB la vue Arborescence d'ID objet, sélectionnez un message d'alerte.
Des informations détaillées sur le message d'alerte apparaissent dans la vue Détails.
2. Pour modifier la zone **@Severity**, cliquez deux fois sur la valeur de la zone et choisissez une nouvelle valeur dans la liste déroulante.
3. Pour modifier la zone **@Type**, cliquez deux fois sur la valeur de la zone et choisissez une nouvelle valeur dans la liste déroulante.
4. Pour modifier la zone **@ExpireTime**, cliquez deux fois sur la valeur de la zone et entrez une nouvelle période (en secondes, minutes, heures ou jours).
5. Pour modifier la zone **CODEBLOCK**, cliquez deux fois sur la valeur de la zone (même si la zone est vide) et entrez un nouveau bloc de code.

Génération d'interceptions SNMP

Comment générer des interceptions SNMP. Vous pouvez également utiliser cette fonctionnalité pour tester les fichiers de règles générés par MIB Manager.

Procédure

Pour générer une interception SNMP :

1. Cliquez sur **! Interception**.
2. Dans la zone **Adresse de destination**, indiquez le nom d'hôte ou l'adresse IP d'un ordinateur exécutant un service SNMP Trapd.
3. Dans la zone **Port de destination**, indiquez le port UDP écouté par le service SNMP Trapd.
4. Dans la zone **Communauté**, indiquez une chaîne valide pour le gestionnaire Trapd.
5. En fonction du type d'interception que vous générez, une section **Liaisons de variable** peut apparaître dans la fenêtre Générer une interception SNMP. Les zones disponibles dans la section **Liaisons de variable** dépendent également du type d'interception généré.

Survolez les noms de la zone **Liaisons de variable** pour plus de détails sur la syntaxe de la valeur de la zone. Vous pouvez indiquer plusieurs valeurs, séparées par des virgules. Ces valeurs sont choisies de manière aléatoire pour générer l'interception.

6. Dans la zone **Métrique récurrente**, indiquez le nombre d'interceptions (de 1 à 1000) qui peuvent être générées.

Si vous indiquez un nombre élevé, les interceptions se succèdent rapidement et peuvent provoquer une tempête d'alarme sur le gestionnaire de Trapd. Pour les valeurs supérieures à un (> 1) et pour les valeurs qui sont spécifiées en tant que listes séparées par un point-virgule, MIB Manager effectue une sélection aléatoire dans les listes spécifiées et génère des interceptions avec les valeurs arbitraires présentées comme des liaisons de variable (varbind).

7. Cliquez sur **Exécuter** pour générer l'interception.

Résultats

La zone **Résultat** dans la fenêtre Générer une interception SNMP indique si l'interception a été envoyée avec succès.

Création et suppression d'unités


Une unité MIB Manager est utilisée pour regrouper des modules MIB associés et pour définir tous les objets requis pour l'exportation. Vous pouvez créer, mettre à jour et supprimer des unités dans l'interface graphique de MIB Manager.

Pourquoi et quand exécuter cette tâche

Les unités MIB Manager sont affichées dans la vue Unité. Sélectionnez une unité dans l'arborescence Unité pour afficher le fabricant, le numéro de modèle et la version du système d'exploitation de l'unité.

Procédure

Création d'une unité

1. Pour créer une unité :
 - a. Dans la vue Modules MIB, sélectionnez au moins un module MIB, puis cliquez sur  (Créer une unité).
 - b. Dans la zone **Fabricant**, sélectionnez un fabricant d'unité dans la liste déroulante.


Les zones **Modèle** **Version du système d'exploitation** sont remplies avec les options disponibles.
 - c. Dans la zone **Modèle**, sélectionnez une option dans la liste déroulante.

Si aucun modèle n'est disponible, entrez un nom de modèle explicite.
 - d. Dans la zone **Version du SE**, sélectionnez une option dans la liste déroulante.

Si aucune version du système d'exploitation n'est disponible, entrez un nom de version du système d'exploitation explicite.
 - e. Dans la zone **Modules MIB mappés**, sélectionnez les modules MIB que vous voulez associer à la nouvelle unité.
 - f. Cliquez sur **OK** pour créer l'unité.

La nouvelle unité est affichée dans la vue Unité.

Mise à jour d'une unité

2. Pour mettre à jour une unité :
 - a. Dans la vue Unité, sélectionnez une unité et développez ses sous-noeuds jusqu'à ce que la version du système d'exploitation soit visible.
 - b. Sélectionnez la version du système d'exploitation, puis cliquez sur  (Editer une unité).
 - c. Dans la zone **Fabricant**, sélectionnez un fabricant d'unité dans la liste déroulante.

Les zones **Modèle** **Version du système d'exploitation** sont remplies avec les options disponibles.
 - d. Dans la zone **Modèle**, sélectionnez une option dans la liste déroulante.


Si aucun modèle n'est disponible, entrez un nom de modèle explicite.
 - e. Dans la zone **Version du SE**, sélectionnez une option dans la liste déroulante.


Si aucune version du système d'exploitation n'est disponible, entrez un nom de version du système d'exploitation explicite.

- f. Dans la zone **Modules MIB mappés**, sélectionnez les modules MIB que vous voulez associer à la nouvelle unité.
- g. Cliquez sur **OK** pour mettre à jour les détails de l'unité.

Les détails actualisés de l'unité sont affichés dans la vue Unité.

Suppression d'une unité

3. Dans la vue Unité, sélectionnez l'unité, puis cliquez sur  (Supprimer une unité).

Pour supprimer plusieurs unités, maintenez les touches Ctrl+Alt enfoncées, sélectionnez les éléments à l'aide du bouton gauche de la souris, puis cliquez sur  (Supprimer une unité).

Configuration des préférences globales


Vous pouvez configurer plusieurs préférences globales de MIB Manager à partir de la fenêtre Préférences.

Pour configurer les préférences globales de MIB Manager, cliquez sur  **Préférences** ou, dans le menu, cliquez sur **Fichier > Préférences**.

Dans la fenêtre Préférences, vous pouvez configurer les préférences de répertoire, d'exportation, générales, de consignation et de recherche.

Définition des préférences de répertoire

Utilisez les préférences de répertoire pour indiquer les répertoires utilisés pour le stockage des MIB importées, des données de MIB et des fichiers de règles exportés.

Pour définir les préférences de répertoire, cliquez sur  **Préférences** ou, dans le menu, cliquez sur **Fichier > Préférences**, puis, dans la sous-fenêtre de gauche de la fenêtre Préférences, cliquez sur **Répertoire**.

- Dans la zone **Répertoire de données**, indiquez l'emplacement où les fichiers MIB importés seront stockés en tant que fichiers XML. L'emplacement par défaut est *NCHOME/omnibus/var/mibmanager/data*.
- Dans la zone **Répertoire d'exportation**, indiquez l'emplacement où les fichiers exportés sont générés. Un sous-répertoire basé sur le type d'exportation est créé. L'emplacement par défaut est *NCHOME/omnibus/var/mibmanager/export*.
- Dans la zone **Répertoire d'importation**, indiquez l'emplacement où se trouvent les fichiers importés. L'emplacement par défaut est *NCHOME/omnibus/var/mibmanager/mibs*.


Tâches associées:

«Importation de données MIB», à la page 152

Comment importer des données MIB SNMP dans MIB Manager.

Définition des préférences d'exportation

Cette section décrit comment définir vos préférences d'exportation de fichier de règles.

Pour définir les préférences de répertoire, cliquez sur  **Préférences** ou, dans le menu principal, cliquez sur **Fichier > Préférences**, puis, dans la sous-fenêtre de gauche de la fenêtre Préférences, cliquez sur **Exportation**.

- Dans la zone **Chemin d'inclusion**, spécifiez l'emplacement des règles qui ont été installées avec les modules de fichiers de règles Netcool Include Library (NCiL) (s'ils ont été installés). Cette option n'est obligatoire que si vous exportez des données au format de fichier Netcool Include Library et lorsque le chemin d'accès au répertoire doit pointer vers les règles incluses. En conséquence, il n'est pas nécessaire d'éditer manuellement les nouveaux fichiers de règles. La valeur par défaut est *NCHOME/omnibus/probes/arch*.
- Dans la zone **Format de fichier d'exportation par défaut**, indiquez soit le format Windows/DOS (CRLF), soit le format Unix (LF) pour le fichier de sortie. Cette propriété vous permet de créer des règles sur une machine Windows et de les copier sur une machine UNIX sans exécuter d'utilitaire, par exemple, *dos2unix*.

Remarque : Cette sélection s'applique à tous les fichiers de sortie et n'est pas limitée aux fichiers de règles. La valeur par défaut est au format Unix.

- Dans la zone **Types d'alerte**, indiquez les options de liste déroulante affichées lorsque la zone @Type d'une interception est éditée dans la vue Détails.

Les options suivantes sont disponibles uniquement si le format d'exportation NCKL a été préalablement sélectionné :

- Si la case **Inclure des descriptions varbind** est sélectionnée, les descriptions varbind sont ajoutées en tant que commentaires. La valeur par défaut est Activé.
- Si la case **Calculer une meilleure valeur pour @AlertKey** est cochée, la valeur @AlertKey est générée à partir d'un objet parent varbind. La valeur par défaut est Activé.
- Si la case **Définir @Agent une fois par entreprise** est cochée, la zone @Agent est définie pour chaque entreprise. La zone @Agent est basée sur les noms des modules MIB en cours de traitement pour le fichier de règles. La valeur par défaut est Activé.
- Si la case **Ajouter un modèle @Class** est sélectionnée, la zone @Class est définie. Si la case **Ajouter un modèle @Class** est sélectionnée, la zone @Class est définie par 300. La valeur par défaut est Activé.


Tâches associées:

«Exportation des données MIB», à la page 154


Exporter les données MIB SNMP à partir de MIB Manager vers différents formats.

Définition des préférences générales

Utilisez les préférences générales pour spécifier les options d'importation MIB.


Pour définir les préférences de répertoire, cliquez sur  **Préférences** ou, dans le menu principal, cliquez sur **Fichier > Préférences**, puis, dans la sous-fenêtre de gauche de la fenêtre Préférences, cliquez sur **Général**.

- Si la case **Traverser les sous-répertoires à l'importation** est cochée, tous les sous-répertoires sont explorés pendant le processus d'importation afin de localiser tous les fichiers MIB.

- Dans la zone **Nombre d'unités d'exécution maximal du gestionnaire de fractionnement de fichier**, indiquez le nombre d'unités d'exécution de calcul utilisées pour détecter des MIB lors du processus d'importation. Cela peut améliorer les performances lors de l'importation d'un grand nombre de MIB. La valeur par défaut est 5.
- Dans la zone **Nombre d'unités d'exécution maximal de l'analyseur MIB**, indiquez le nombre d'unités d'exécution de calcul utilisées pour analyser des fichiers MIB lors du processus d'importation. Cela peut améliorer les performances lors de l'importation d'un grand nombre de MIB. La valeur par défaut est 10.
- Pour modifier la couleur du surligneur de toutes les sélections effectuées dans la fenêtre MIB Manager, cliquez sur  , sélectionnez une couleur dans la palette de couleurs, puis cliquez sur **OK**.

Définition de préférences de journalisation

Utilisez les préférences de consignation pour spécifier le répertoire du fichier journal MIB Manager et pour définir le niveau de message par défaut.

Pour définir les préférences de répertoire, cliquez sur  **Préférences** ou, dans le menu principal, cliquez sur **Fichier > Préférences**, puis, dans la sous-fenêtre de gauche de la fenêtre Préférences, cliquez sur **Consignation**.

- Dans la zone **Répertoire de consignation**, spécifiez l'emplacement d'écriture du fichier journal d'application. L'emplacement par défaut est *NCHOME/omnibus/log*.
- Dans la zone **Niveau de message par défaut**, indiquez le niveau de message par défaut de l'application. Vous pouvez également utiliser le menu **Editer** pour modifier le niveau de consignation de messages par défaut. Le niveau de message est valide à partir du premier démarrage de l'application jusqu'à l'arrêt de celle-ci, excepté si le niveau est modifié à l'aide du menu **Editer**.

Remarque : La modification du niveau de message dans la zone **Niveau de message par défaut** ne modifie pas le niveau de message en cours de l'application. Elle ne concerne que le niveau de message ayant été initialement défini lors du premier démarrage de l'application (le niveau de message par défaut). Pour éviter de générer des fichiers journaux volumineux, définissez le niveau de message par défaut **warn**.

Définition du niveau de consignation

Vous pouvez modifier le niveau de consignation afin de fournir des fichiers journaux ou des fichiers de trace plus détaillés pour vous aider à déboguer MIB Manager. Les informations supplémentaires fournies par l'augmentation des niveaux de consignation peuvent vous aider lors du débogage de problèmes spécifiques pendant l'analyse syntaxique des fichiers MIB.

Pour modifier le niveau de consignation, cliquez sur **Editer > Déboguer** dans la barre de menus de la fenêtre principale. Les fichiers journaux ont six niveaux de message que vous pouvez utiliser pour augmenter le niveau de détail capturé : aucun, erreur, avertissement, info, débogage et prolix.


Le tableau suivant donne une brève description de chaque niveau de consignation :

Tableau 37. Liste des niveaux de consignation avec la description associée

Niveau de consignation	Description
Aucune	Indique un niveau de consignation dans lequel aucun message de débogage n'est généré. Les performances de l'application sont améliorées, mais aucun message d'avertissement ou d'erreur n'est affiché lors de l'analyse syntaxique.
erreur	Indique un niveau de consignation dans lequel seules les erreurs qui affectent les résultats sont consignées. Ceci inclut toute erreur identifiée par MIB Manager qui entraînerait l'affichage de données incorrectes dans la fenêtre principale.
avertissement	Indique un niveau de consignation dans lequel des messages d'avertissement et d'erreur sont consignés. Un avertissement indique une condition qui pourrait conduire à des données erronées et qui doit donc être vérifiée. Toutefois, cette condition ne cause pas de détérioration des données.
info	Indique un niveau de consignation dans lequel des messages d'information, d'avertissement et critiques sont consignés. Les messages d'information contiennent des détails concernant la progression globale de la tâche. Par exemple, ils fournissent le texte complet pour des objets en cours d'analyse.
débogage	Indique un niveau de consignation dans lequel tous les messages sont consignés. Remarque : Exécutez MIB Manager uniquement en mode débogage lorsque vous essayez de résoudre une condition d'erreur. Le volume de données produites lors de l'analyse syntaxique d'un nombre modéré de MIB remplit rapidement votre système de fichiers.
prolix	Indique un niveau de consignation dans lequel tous les messages d'information sont consignés. Un message en mode prolix inclut l'état des variables, des tableaux et des hachages au fur et à mesure que leurs valeurs et leur état changent.

Définition des préférences de recherche

Cette section explique comment définir les préférences de MIB Manager qui sont utilisées pour rechercher les fichiers MIB.

Pour définir les préférences de recherche, cliquez sur  **Préférences** ou, dans le menu principal, cliquez sur **Fichier > Préférences**, puis, dans la sous-fenêtre de gauche de la fenêtre Préférences, cliquez sur **Recherche**.

Dans la zone **Chemin de recherche MIB**, définissez le chemin de recherche dans lequel se trouvent les fichiers MIB. Les répertoires doivent être séparés par un point-virgule (;). Tous les sous-répertoires sont automatiquement explorés pour localiser les fichiers MIB dépendants.

Le chemin d'accès par défaut inclut l'emplacement des fichiers MIB de base et les fichiers MIB RFC qui sont livrés avec MIB Manager. Vous pouvez spécifier n'importe quel nouveau répertoire à la fin du chemin par défaut. Cela garantit que

le répertoire de base est initialement exploré pour localiser les dépendances contenant les MIB le plus couramment utilisées, puis le répertoire rfc est exploré pour localiser les MIB le moins couramment utilisées.

Tâches associées:

«Importation de données MIB», à la page 152
Comment importer des données MIB SNMP dans MIB Manager.

A propos de SNMP

Cette section fournit des informations supplémentaires sur l'architecture SNMP (Simple Network Management Protocol), la base d'informations de gestion SNMP (MIB) et certains concepts MIB supplémentaires.

Dans une architecture SNMP, un composant gestionnaire gère un agent. L'agent est un logiciel qui s'exécute sur un périphérique ou une application réseau, répond aux demandes d'informations (SET et GET), puis génère des notifications autonomes appelés interceptions. Le gestionnaire est un logiciel qui reçoit les interceptions et fournit un mécanisme permettant de définir (SET) ou d'extraire (GET) des objets SNMP sur le périphérique réseau.

Pour recevoir des interceptions autonomes, le gestionnaire exécute une application qui écoute le port d'interception SNMP TCP/IP (port 162). Les actions SNMP SET et GET utilisent le port 161. Cette application est généralement appelée Trapd ou démon d'interception. Un démon d'interruption est un processus qui s'exécute en arrière-plan et gère un service sur un ordinateur. L'application Trapd Netcool est appelée SNMP Probe (nco_p_mttrapd, où mt est une abréviation de multithreaded) et elle se trouve dans le répertoire \$OMNIHOME/probes/. Les fichiers de règles générés par MIB Manager sont conçus pour être utilisés par la sonde SNMP.

Sécurité

La sécurité de base dans SNMP version v1 et v2 est fournie par l'utilisation de chaînes de communauté. Les chaînes de communauté sont des mots de passe en texte en clair qui sont envoyés avec toutes les demandes. Il existe des chaînes de communauté séparées pour l'accès en lecture seule et l'accès en lecture et en écriture. MIB Manager doit connaître la chaîne de communauté définie sur un périphérique pour pouvoir exécuter des requêtes (en lecture seule pour une extraction (GET) et en lecture-écriture pour une définition (SET)) et des interceptions et des notifications SNMP sont envoyées à MIB Manager avec une chaîne de communauté prédéfinie.

Tous les périphériques réseau prenant en charge SNMP ont un mécanisme permettant de définir la chaîne de communauté. Le mot de passe en lecture seule par défaut standard est public et le mot de passe en lecture-écriture par défaut standard est privé. Si aucune chaîne de communauté n'a été définie sur un périphérique, la chaîne est normalement l'un des deux mots de passe suivants. Pour des raisons de sécurité, les mots de passe par défaut doivent être modifiés dès que possible.

Informations complémentaires

Les manuels suivants sont utiles pour obtenir une bonne compréhension de la structure SNMP et des MIB :

- *Managing Internetworks with SNMP (Third Edition)* by Mark Miller (Wiley, 1999)

- *Understanding SNMP MIBs* by David Perkins and Evan McGinnis (Prentice Hall, 1996)

Pour une discussion exhaustive sur l'ASN.1 et les règles BER (Basic Encoding Rules) qui sont utilisées pour le codage de données SNMP en paquets PDU (Protocol Data Unit) pour transmission sur le réseau, consultez le manuel suivant :

SNMP, SNMPv2, and CMIP: The Practical Guide to Network-Management Standards by William Stallings (Addison-Wesley, 1993).

Concepts et conception de MIB

Tous les modules MIB SNMP qui sont définis pour être utilisés par un périphérique spécifique constituent la base MIB de ce périphérique. Le terme MIB est souvent utilisé pour décrire une définition de module unique, cependant, cela est techniquement incorrect. En effet, MIB est la combinaison de tous les modules utilisés pour la gestion d'une unité spécifique, que cette unité se rapporte à du matériel ou du logiciel. Par conséquent, le nom plus précis pour chaque module défini par un fournisseur ou dans une RFC, est module MIB SNMP.

Tous les modules MIB sont finalement des extensions du module racine. Tous les modules MIB publiés par chaque fournisseur sont des extensions de l'objet d'entreprise défini dans RFC1155-SMI. Par conséquent, tous les agents SNMP doivent prendre en charge RFC1155 et tous les modules MIB sont des extensions de RFC1155.

Structure of Management Information (SMI)

Pour rendre la base SNMP MIB (Management Information Base) extensible, les éléments associés sont organisés en modules MIB qui forment une hiérarchie structurée. Chaque module MIB est défini à l'intérieur de la construction suivante :

```
ModuleName DEFINITIONS ::= BEGIN END
```

Les balises BEGIN et END dans le module permettent de définir plusieurs modules dans un fichier texte unique. Les compilateurs MIB doivent être en mesure de gérer n'importe quel nombre de modules définis dans un fichier unique, mais ne devraient pas avoir à le faire.

Il existe des conventions pour chaque objet défini dans le module. Par exemple, un nom de module doit commencer par un caractère alphabétique en majuscule et contenir uniquement des lettres, des chiffres, des tirets (-) ou des traits de soulignement (_). Un nom d'objet doit commencer par un caractère alphabétique en minuscule et doit contenir uniquement des lettres, des chiffres, des tirets ou des traits de soulignement. Les commentaires dans les modules MIB sont représentés par deux tirets (--) consécutifs et le texte éventuel qui suit ce symbole, sur n'importe quelle ligne, peut être ignoré.

La conception modulaire, aisément extensible des MIB leur permet de prendre en charge toute nouvelle fonctionnalité ou tout nouvel appareil par l'ajout d'un module. Lorsqu'un module est écrit en tant qu'extension d'un autre module, il inclut une section IMPORTS qui se trouve sous la ligne DEFINITIONS. La section IMPORTS définit les objets requis par les modules de niveau supérieur dans la hiérarchie MIB et les modules dans lesquels ils sont, à leur tour, définis.

La définition suivante est tirée de RFC1157 et indique plusieurs objets qui sont importés à partir de RFC1155. Cette section peut être considérée comme analogue à l'instruction `include` dans un langage de programmation tel que C ou Perl ou même dans un fichier de règles Netcool. En outre, pour comprendre les objets dans le module MIB courant (RFC1157-SNMP), vous devez également tenir compte des objets dans le module MIB précédent (RFC1155-SMI).

```
RFC1157-SNMP DEFINITIONS ::= BEGIN IMPORTS ObjectName, ObjectSyntax,
NetworkAddress, IpAddress, TimeTicks FROM RFC1155-SMI;
```

Les erreurs typographiques sont courantes dans l'indication de noms de MIB importées. Par exemple, RFC1212 peut être référencé en tant que module MIB au lieu du nom correct, RFC-1212. Si des erreurs d'analyse syntaxique sont mises en évidence par MIB Manager, vous devez vérifier la section `IMPORTS` afin de confirmer que les modules MIB sont correctement nommés. Certains modules MIB contiennent également une section `EXPORTS` (qui se termine également par un point-virgule). Cette section informe le lecteur que l'auteur des MIB attend d'autres modules MIB pour utiliser les mêmes objets spécifiés. Pour nos objectifs, cette section est inutile et elle peut être ignorée.

Types de données définis

Les modules MIB SNMP sont définis dans un format appelé ASN.13 (Abstract Syntax Notation 1). Cependant, SNMP n'utilise qu'une partie d'ASN.14. ASN.1 est défini dans ITU-T X.208 et dans la norme ISO 8824. Les portions d'ASN.1 qui s'appliquent à SNMP sont définies dans RFC1155. RFC1155 définit les types de données SNMP valides suivants :

- Types primitifs : INTEGER, OCTET STRING, OBJECT IDENTIFIER, NULL
- Types constructeur : SEQUENCE et SEQUENCE OF
- Types définis : NetworkAddress, IpAddress, Counter, Gauge, TimeTicks, Opaque

Un type défini est le mécanisme utilisé pour spécifier un format particulier pour les types primitifs ou constructeur. Les auteurs MIB peuvent définir d'autres types à l'aide de la construction `TEXTUAL-CONVENTION`.

`DisplayString` est un bon exemple de type défini. Dans le module MIB `SNMPv2-SMI-v1`, la version v1 de `DisplayString` contient la définition suivante :

```
DisplayString ::= OCTET STRING (0..255)
```

Dans le module MIB `SNMPv2-TC`, la version v2 de `DisplayString` contient la définition suivante :

```
DisplayString ::= TEXTUAL-CONVENTION DISPLAY-HINT "255a" STATUS current
DESCRIPTION "Représente des informations textuelles extraites du jeu
de caractères ASCII NVT, tels qu'elles sont définies dans les pages
4, 10-11 de RFC 854. Pour récapituler RFC 854, le répertoire
NVT ASCII spécifie :
```

- l'utilisation des code caractères 0 à 127 (décimal)
- caractères graphiques (32 à 126) sont interprétés en tant que US ASCII
- NUL, LF, CR, BEL, BS, HT, VT et FF ont des significations spéciales dans RFC 854
- les 25 autres codes n'ont pas d'interprétation standard
- la séquence 'CR LF' signifie un retour à la ligne
- la séquence 'CR NUL' signifie un retour chariot
- le code 'LF' non précédé du code 'CR' signifie un déplacement dans la même colonne sur la ligne suivante.

- la séquence 'CR x' pour tout x autre que LF ou NUL n'est pas admise. (Il est à noter que cela signifie également qu'une chaîne peut se terminer par 'CR LF' ou 'CR NUL', mais pas par CR.) Tout objet défini à l'aide de cette syntaxe ne doit pas comporter plus de 255 caractères." SYNTAX OCTET STRING (SIZE (0..255))

L'exemple ci-dessus montre qu'une chaîne DisplayString est une chaîne OCTET STRING de 0 à 255 caractères de long. Notez que chaque DESCRIPTOR OBJECT qui correspond à un type d'objet dans une base MIB Internet standard doit être une chaîne unique, mnémonique et imprimable.

Définition d'objets

Une erreur courante faite lors de l'écriture des modules MIB consiste à créer un nom d'objet qui n'est pas unique. On prétend que l'instruction RFC1155 signifie que seuls les objets au sein d'un module MIB unique doivent être uniques. Comme traité précédemment, la base MIB est l'ensemble complet des modules qui, lorsqu'ils sont combinés, sont utilisés pour gérer une unité particulière. Par conséquent, tous les objets définis dans n'importe quel module MIB doit être unique, non seulement dans son propre module, mais aussi dans n'importe quel autre nom d'objet de tout module importé et de tout module que ces modules peuvent importer. Un mécanisme courant pour s'assurer que les noms d'objet sont uniques consiste à ajouter au début de tous les noms de module le symbole boursier de la société ou le nom abrégé de l'entreprise.

Lorsque les objets sont définis, ils sont mappés dans une hiérarchie numérique qui ressemble à une arborescence maximale. Chaque fois qu'un objet est défini, il est défini en tant que feuille d'un objet parent. Les trois objets racine suivants sont définis dans l'arbre MIB SNMP :

- ccitt (noeud racine zéro)
- iso (noeud racine 1)
- joint-iso-ccitt (noeud racine 2)

Tous les autres noeuds de l'arborescence MIB sont les enfants de l'un de ces trois noeuds racines. Par exemple, RFC1155-SMI définit les objets suivants :

```
internet OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 } directory
OBJECT IDENTIFIER ::=
  { internet 1 } mgmt OBJECT IDENTIFIER ::= { internet 2 }
experimental OBJECT IDENTIFIER
  ::= { internet 3 }
private OBJECT IDENTIFIER ::= { internet 4 } enterprises OBJECT
IDENTIFIER ::= { private 1 }
```

Ces définitions indiquent le nom de l'objet, les types d'objet associé, le nom parent de chaque objet (ou la liste ordonnée des parents) et le nombre de feuilles de cet enfant vers ce parent (ou parents). Graphiquement, ces éléments constituent une forme hiérarchique.

Vous pouvez naviguer dans la vue d'arborescence MIB en développant et en réduisant les noeuds dans l'arborescence MIB. Les branches du niveau supérieur de l'arborescence MIB contiennent les noms des modules MIB, et dans chaque branche de module MIB se trouvent les autres éléments qui composent la base MIB. Comme des modules supplémentaires sont ajoutés à la base MIB, des objets supplémentaires sont ajoutés à l'arborescence MIB. Chaque objet peut être désigné

par son nom d'objet ou par son identificateur d'objet. La méthode la plus précise consiste à se référer à son ID objet. Son ID objet (OID) est défini comme son numéro et chacun de ses numéros d'ancêtre remontant vers le noeud racine concaténé à un point (.) qui sépare chacun d'eux. L'ID objet de l'objet entreprise (noeud ou feuille) est 1.3.6.1.4.1.

De nombreux fournisseurs ne garantissent pas que leurs noms d'objet sont universellement uniques ; par conséquent, il est possible que deux fournisseurs disposent d'un objet partageant le même nom. Cela rend l'utilisation du nom d'objet pour identifier un objet un peu ambiguë.

Types d'objet MIB

Cette rubrique décrit les types d'objet définis dans SNMP v1 et v2.

Vous pouvez localiser les informations d'objet décrites dans les sections suivantes en sélectionnant un module dans la vue Modules MIB, puis en recherchant `ifIndex` dans la zone **Recherche** de la vue Arborescence d'ID objet. Cliquez sur l'objet `ifIndex` dans la hiérarchie de l'Arborescence d'ID objet pour afficher les informations sur l'objet et sur les conventions de texte dans la vue Détails.

OBJECT IDENTIFIER

OBJECT IDENTIFIER est défini par SNMP v1 et constitue le bloc de construction principal de l'arborescence MIB. Les identificateurs d'objet sont analogues à un en-tête de chapitre dans un manuel, ils ne contiennent pas de données réelles, mais ils indiquent quel type de contenu est relayé par leurs descendants.

OBJECT TYPE

OBJECT-TYPE est défini par SNMP v1 et est utilisé en tant que conteneur pour le stockage des informations sur le périphérique géré ou d'une valeur mesurée sur le périphérique.

TEXTUAL CONVENTION

TEXTUAL-CONVENTION (TC) est la définition d'un type d'objet et non un objet réel. Dans la vue Modules MIB, vous pouvez sélectionner **Conventions textuelles** dans la liste **Vue** afin d'afficher les conventions textuelles analysées, affichées dans l'arbre MIB. Sélectionnez un nom de TC dans l'arborescence MIB pour afficher sa définition dans la vue Détails.

SNMP v1 TRAP TYPE et SNMP v2 NOTIFICATION TYPE

SNMP v1 TRAP-TYPE et v2 NOTIFICATION-TYPE sont des mécanismes SNMP permettant de générer des événements autonomes vers le gestionnaire SNMP. Les interceptions SNMP dans v1 ne sont pas définies en tant qu'objets dans l'arborescence MIB. Un objet TRAP-TYPE ne possède pas de parent défini au format OBJECT IDENTIFIER. En revanche, une définition d'interception indique une entreprise pour laquelle une interception est définie. La phrase suivante est un objet TRAP-TYPE courant :

```
bgpEstablished TRAP-TYPE ENTERPRISE bgp VARIABLES
{ bgpPeerRemoteAddr, bgpPeerLastError,
  bgpPeerState }
DESCRIPTION "L'événement BGP est généré lorsque BGP FSM
passe à l'état ESTABLISHED." ::= 1
```

La section ENTERPRISE définit l'objet qui est le parent de l'interception. Toutefois, il est possible pour un objet de l'arborescence MIB d'être défini avec `bgp` en tant que parent, et d'être défini en tant qu'enfant numéro 1. En fait, `bgpVersion` est défini en tant que { `bgp 1` } dans le module RFC1269-MIB. Par conséquent, il est impossible d'ajouter une interception `v1` à l'arborescence MIB en tant que feuille à l'aide d'ENTERPRISE en tant que parent.

SNMP v2 remplace la définition de TRAP-TYPE par NOTIFICATION-TYPE et indique que cette nouvelle interception v2 doit être définie comme d'autres objets MIB, avec un numéro de parent et d'enfant, ce qui pose un problème uniquement pour les interceptions v1. L'article RFC1155 4,1 définit que l'utilisation du zéro (0) en tant que numéro enfant n'est pas valide et qu'elle est réservée à un usage ultérieur. SNMP v2 utilise ce zéro en permettant aux fournisseurs d'ajouter leurs interceptions v1 à une MIB v2, en ajoutant un zéro au nom de l'entreprise, puis en ajoutant le numéro d'interception après le zéro. Par conséquent, en v2, il est opportun de définir un identificateur d'objet avec un zéro en tant qu'enfant de l'entreprise, puis d'ajouter les interceptions v1 en tant qu'enfants de ce zéro.

Cette convention a entraîné une autre erreur commune faite par les auteurs MIB. La section 4 de la norme RFC1155 déclare :

«Une définition de type d'objet est composée de cinq zones : OBJECT : ----- Nom textuel, appelé OBJECT DESCRIPTOR, pour le type d'objet, ainsi que ses identificateurs OBJECT IDENTIFIER correspondants. Syntaxe : Syntaxe abstraite pour le type d'objet. Elle doit être remplacée par une instance de la syntaxe ObjectSyntax de type ASN.1 (définie ci-dessous). Définition : description textuelle de la sémantique du type d'objet. Les implémentations doivent garantir que leur instance de l'objet répond à cette définition car cette MIB est conçue pour être utilisée dans des environnements multifournisseur. De ce fait, il est essentiel que les objets aient une signification cohérente sur l'ensemble des machines. Accès : l'un des types suivants : lecture seule, lecture-écriture, écriture uniquement ou non accessible. Etat : l'un des types suivants : obligatoire, facultatif ou obsolète. Les mémos ultérieurs peuvent également indiquer d'autres zones pour les objets qu'ils définissent.»

Selon cette règle, tous les objets doivent avoir à la fois un nom d'objet et un numéro d'objet. Certains modules MIB du fournisseur, et même certains RFC, ont défini un type NOTIFICATION-TYPE avec un parent zéro, mais sans nom d'objet pour ce zéro. Dans l'exemple suivant, la définition d'objet n'est pas réellement correcte du point de vue syntaxique car il n'existe aucun nom d'objet défini pour le numéro d'enfant zéro de l'objet `adslAtucTraps`. MIB Manager reconnaît la préférence de certains auteurs MIB à utiliser des méthodes tel que des raccourcis et à autoriser l'ajout de l'objet sans nom d'objet. De plus, pour faciliter l'ajout de messages d'alerte v1 à l'arborescence MIB, MIB Manager ajoute automatiquement un objet zéro en tant qu'enfant de l'objet entreprise v1 (notez qu'une MIB v1 ne peut pas utiliser de zéro dans son OID), affecte cet objet zéro en tant que Traps où se trouve le nom de l'entreprise et ajoute le message d'alerte sous ce nouvel objet dans l'arborescence MIB. Par exemple, l'utilisation de `bgp` produirait les interceptions d'ancêtre suivantes : { `bgp bgpTraps(0) 1` }).

```
adslAtucPerfLofsThreshTrap NOTIFICATION-TYPE OBJECTS { adslAtucPerfCurr15MinLofs,
adslAtucThresh15MinLofs } STATUS current DESCRIPTION "Perte de trame, seuil
d'intervalle de 15 minutes atteint." ::= { adslAtucTraps 0 1 }
```

Varbinds

Les objets transmis avec l'interception v1 ou la notification v2 sont appelés varbind. Les objets varbind contiennent des informations supplémentaires sur l'événement signalé. Dans une interception v1, les objets varbind sont détaillés dans la section VARIABLES et dans une notification v2, les objets varbind sont répertoriés dans la section OBJECTS. Ils ont la même utilisation dans toutes les versions de SNMP. L'ordre dans lequel les objets varbind apparaissent dans la liste est important car le PDU (paquet SNMP) code les valeurs associées dans l'ordre où elles sont répertoriées dans la MIB.

Par exemple, dans la section OBJECTS, les trois objets varbind suivants ont été spécifiés : ifIndex, ifAdminStatus et ifOperStatus. Par conséquent, ifIndex est le premier objet varbind à être codé, ifAdminStatus est le second et ifOperStatus est le troisième. Dans l'IF-MIB, nous observons que le type d'objet ifIndex est défini en tant qu'InterfaceIndex. Etant donné qu'il ne s'agit pas d'un type ASN.1 primitif pour SNMP, il doit s'agir d'une convention textuelle. Dans les conventions textuelles, nous observons qu'InterfaceIndex a pour résultat un entier Integer32 (entier sur 32 bits). Par conséquent, lorsque le PDU arrive au niveau de MIB Manager, le premier objet varbind est un entier. Pour déterminer la signification de cet entier, MIB Manager doit faire référence au module IF-MIB, rechercher ifIndex et lire les informations d'objet associées. Lorsque l'on vérifie le second objet varbind, nous trouvons un type entier énuméré :

```
SYNTAX INTEGER { up(1), -- ready to pass packets down(2),  
testing(3) -- in some test mode }
```

Lorsque l'objet varbind est décodé à partir du paquet SNMP, sa valeur est un entier dont la valeur doit être interprétée, en fonction des éléments de cette liste énumérée. Lorsque MIB Manager est utilisé pour créer un fichier de règles, il crée une table de recherche pour lier automatiquement l'entier énuméré à sa représentation textuelle. Le troisième objet varbind est également un type énuméré avec les mêmes valeurs. Par conséquent, si l'état ifAdminStatus est 1 (actif) et l'état ifOperStatus est 2 (inactif), nous savons pourquoi l'événement a été généré et nous pouvons tenter de déterminer la cause de cette indisponibilité.

Les objets varbind sont présentés à l'utilisateur dans un fichier de règles sous la forme \$1, \$2, \$3, etc. dont chaque numéro représente un numéro d'objet varbind. MIB Manager crée des éléments basés sur les éléments varbind et utilise ces éléments pour définir des variables dans le tableau de détails. Par exemple, les éléments utilisés dans la table de détails peuvent être \$ifIndex = \$1, qui est un entier, \$ifAdminStatus = \$2, qui est un état de type actif (1) et \$ifOperStatus = \$3, qui est un état de type inactif (3). Toutes les modifications apportées aux paramètres de l'objet sont automatiquement définies dans le fichier de règles à l'aide des conventions définies par Netcool Knowledge Library (NCKL).

Tables

Les tables représentent l'équivalent d'un tableau multidimensionnel avec des lignes et des colonnes de données. L'objet table est défini par SEQUENCE OF d'un objet Entry. L'objet Entry est ensuite défini en tant que SEQUENCE d'objets OBJECT-TYPE. Parfois, un fournisseur conçoit un système inhabituel, par exemple le routeur Cisco 10k. Ce périphérique gère une table interne de conditions d'alarme et génère une interception ou une notification lorsque la table est modifiée. Vous devez alors émettre une requête SNMP GET sur le contenu de la table afin de déterminer l'état actuel de des alarmes actives sur le périphérique. Il est alors plus difficile pour le gestionnaire SNMP d'obtenir des alarmes mais cela reste possible si

l'administrateur dispose des outils adéquats.

OCTET STRING

Un octet est une construction de données composée de huit bits (communément appelé octet). Une chaîne OCTET STRING est donc un tableau d'octets (ou chaîne d'octets). L'expression OCTET STRING n'implique pas que tous les octets de la chaîne soient des caractères alphanumériques. Il peut également s'agir de caractères binaires et ils peuvent être utilisés en tant que masques de bit.

Formats d'objet MIB valides

Cette rubrique décrit les formats des objets MIB valides.

Les sections suivantes décrivent les formats d'objet MIB SNMP valides.

TEXTUAL-CONVENTION

Une convention TEXTUAL-CONVENTION (TC) v2 a la syntaxe suivante, où le nom d'objet est suivi de l'entrée ::=, puis de TEXTUAL-CONVENTION. Plusieurs sections sont ajoutées à la fin de la définition SYNTAX.

```
DisplayString ::= TEXTUAL-CONVENTION
DISPLAY-HINT "255a"
STATUS current
DESCRIPTION "Représente des informations textuelles extraites du jeu de
caractères NVT SCII, tel qu'il est défini dans les pages 4, 10-11 de la
norme RFC 854. Pour récapituler RFC 854, le répertoire NVT ASCII spécifie :
```

- l'utilisation des codes caractères 0 à 127 (décimal)
- les caractères graphiques (32 à 126) sont interprétés sous forme de US ASCII
- NUL, LF, CR, BEL, BS, HT, VT et FF ont les significations spéciales indiquées dans RFC 854
- les 25 autres codes n'ont pas d'interprétation standard
- la séquence 'CR LF' signifie un retour à la ligne
- la séquence 'CR NUL' signifie un retour chariot
- le code 'LF' non précédé du code 'CR' signifie un déplacement dans la même colonne sur la ligne suivante
- la séquence 'CR x' pour tout x autre que LF ou NUL n'est pas admise.

(Il est à noter que cela signifie également qu'une chaîne peut se terminer par 'CR LF' ou 'CR NUL', mais pas par CR.) Tout objet défini à l'aide de cette syntaxe ne doit pas comporter plus de 255 caractères."

```
SYNTAX OCTET STRING (SIZE (0..255))
```

Un TC v1 se compose du nom d'objet suivi de l'entrée ::= et il est ensuite ajouté à la fin d'une définition SYNTAX valide. Par exemple :

```
DisplayString ::= OCTET STRING
```

Ces deux objets existent en dehors de l'arborescence MIB et ne sont donc pas des objets de la MIB. Ils représentent un format pour lequel une syntaxe peut être définie. Ils ne possèdent pas d'ID objet. Vous pouvez les afficher dans la vue Modules MIB, dans un filtre séparé, en cliquant sur la liste déroulante **Vue** et en sélectionnant **Conventions textuelles**.

Objets TRAP-TYPE

L'objet TRAP-TYPE est valide pour les MIB v1. Ces objets n'ont pas été initialement définis pour s'adapter correctement dans l'arborescence MIB. Ils ne possèdent pas d'ID objet mais un ID entreprise et un numéro d'interception, par exemple :

```

newRoot TRAP-TYPE
ENTERPRISE dotIdBridge
DESCRIPTION "L'interception newRoot indique que l'agent expéditeur est devenu
la nouvelle racine de Spanning Tree ; l'interception est envoyée par un pont
aussitôt après son élection en tant que nouvelle racine, par exemple, à
l'expiration du délai de changement de topologie immédiatement après
son élection."
::= 1

```

Une interception v1 commence par un nom d'objet suivi par le mot clé TRAP-TYPE. Le tout est suivi d'un certain nombre de sections et se termine par l'entrée ::= et un nombre. Les accolades ({}) ne sont jamais utilisées avant ou après un nombre.

Objets MACRO

Un objet MACRO définit le format des autres objets MIB. Les définitions de MACRO commencent toujours par le type d'objet suivi du mot clé MACRO, puis de l'entrée ::= . Le reste de la définition de macro est placé entre les balises BEGIN et END, par exemple :

```

OBJECT-TYPE MACRO ::=
BEGIN
TYPE NOTATION ::= "SYNTAX"
type (TYPE ObjectSyntax) "ACCESS" Access "STATUS" Status VALUE NOTATION ::= value
(VALUE ObjectName) Access ::= "read-only" | "read-write" | "write-only" |
"not-accessible"
Status ::= "mandatory" | "optional" | "obsolete"
END

```

Autres objets

Tous les autres objets doivent respecter le format suivant :

```

snmpInPkts OBJECT-TYPE SYNTAX Counter ACCESS read-only STATUS mandatory DESCRIPTION
"Nombre total de messages distribués à l'entité SNMP à partir du service de
transport." ::= { snmp 1 }

```

Ces objets commencent par un nom d'objet qui doit commencer par un caractère alphabétique en minuscule. Le tout est suivi d'un mot clé qui indique le type d'objet. Les sections supplémentaires suivent le mot clé et l'entrée ::= ainsi qu'une liste des ancêtres entre accolades ({}) complète le format. Les ancêtres à l'intérieur des accolades ({}) peut avoir l'un des deux formats suivants. Dans le premier (illustré ci-dessus), le format est un nom d'objet suivi d'un nombre. Le nom d'objet est le nom du parent immédiat de cet objet et le nombre est le nombre de feuilles de cet objet vers le parent.

Le second format valide pour une liste d'ancêtres est une liste de tous les ancêtres allant jusqu'à un objet connu, par exemple :

```

internet OBJECT IDENTIFIER ::= { iso org(3) dod(6) 1 }

```

Dans l'exemple ci-dessus, la liste commence à un objet connu (iso étant la racine de l'arborescence) et continue de définir des noms d'objet et des numéros de feuille pour chaque génération successive, org(3) et dod(6), jusqu'à l'entier unique final (1) qui indique le numéro de l'objet. Notez que les espaces entre les noms d'objet et les numéros d'objet ne sont pas admis.

Chapitre 8. A propos des passerelles

Les passerelles Tivoli Netcool/OMNIBus vous permettent d'échanger des alertes entre les serveurs ObjectServer et des applications de tiers complémentaires, telles que les bases de données et le système de centre d'assistance ou Customer Relationship Management (CRM).

Vous pouvez utiliser des passerelles pour répliquer des alertes ou gérer un serveur ObjectServer de secours. Les passerelles d'application vous permettent d'intégrer différentes fonctions métier. Vous pouvez par exemple configurer une passerelle pour envoyer des informations d'alerte à un système de centre d'assistance. Vous pouvez également utiliser une passerelle pour archiver des alertes dans une base de données.

La figure suivante montre un exemple d'architecture de passerelle.

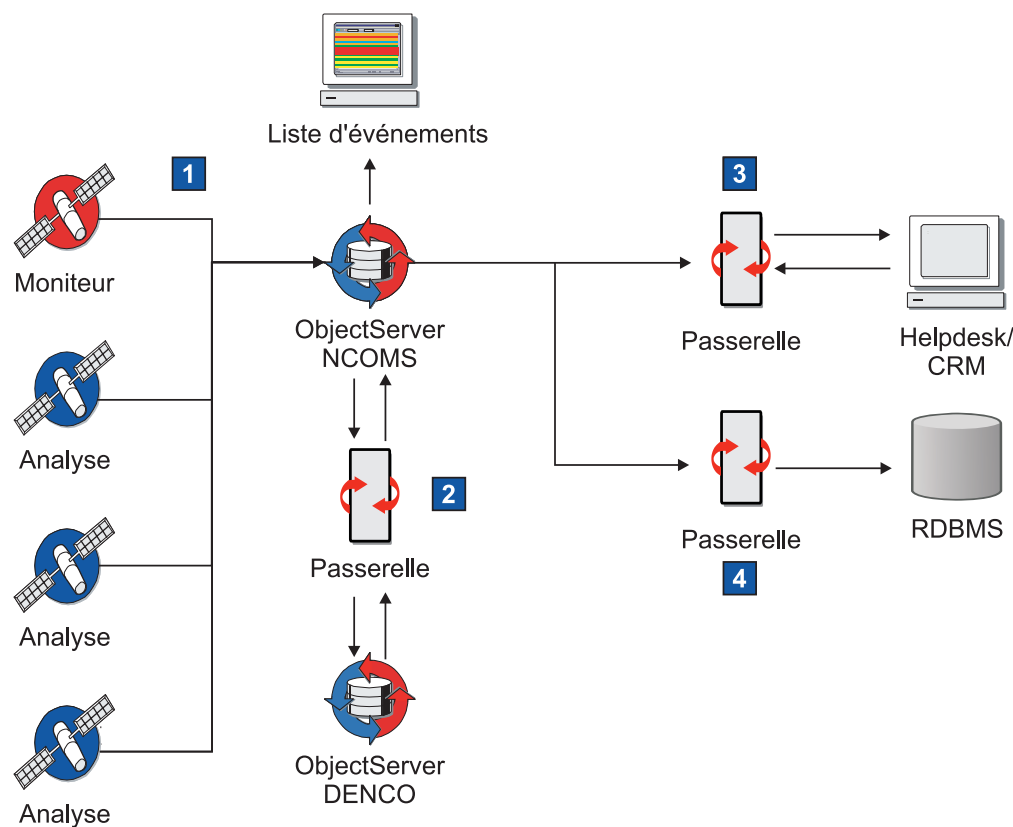


Figure 5. Passerelles dans l'architecture Tivoli Netcool/OMNIBus

La figure précédente montre comment utiliser les passerelles pour différents objectifs :

- 1** Les sondes envoient des alertes au serveur ObjectServer local.
- 2** La passerelle ObjectServer réplique des alertes entre les serveurs ObjectServer d'une configuration de reprise en ligne.
- 3** La passerelle Centre d'assistance intègre le Centre d'opérations réseau

(Network Operations Center, NOC) et le centre d'assistance en convertissant des tickets d'incidents en alertes et inversement.

- 4** La passerelle RDBMS stocke les alertes critiques dans un système de gestion de base de données relationnelle (RDBMS) pour vous permettre d'analyser les performances réseau.

Une fois qu'une passerelle est correctement installée et configurée, le transfert des alertes est transparent pour les opérateurs. Par exemple, les alertes sont automatiquement transmises d'un serveur ObjectServer à une base de données dans intervention de l'utilisateur.

Remarque : Les informations de cette publication sont générales et communes à toutes les passerelles. Pour obtenir des informations spécifiques aux passerelles, voir les publications sur les passerelles individuelles dans le centre de documentation IBM Tivoli Network Management à l'adresse :

<http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp>

Concepts associés:

«Types de passerelles»

Il existe deux types de passerelles principaux : les passerelles unidirectionnelles et bidirectionnelles.

Types de passerelles

Il existe deux types de passerelles principaux : les passerelles unidirectionnelles et bidirectionnelles.

Les passerelles *unidirectionnelles* permettent la transmission des alertes dans une seule direction. Les modifications apportées dans le serveur ObjectServer source sont répliquées dans un serveur ObjectServer ou une application de destination, mais les modifications apportées dans le serveur ObjectServer ou l'application de destination ne sont pas répliquées dans le serveur ObjectServer source. Les passerelles unidirectionnelles peuvent être considérées comme des outils *d'archivage*.

Les passerelles *bidirectionnelles* permettent de transmettre les alertes du serveur ObjectServer source au serveur ObjectServer ou à l'application cible ainsi que le retour au serveur ObjectServer cible. En configuration de passerelle bidirectionnelle, les modifications apportées au contenu d'un serveur ObjectServer source sont répliquées dans un serveur ObjectServer ou une application de destination, qui réplique à son tour ses alertes dans le serveur ObjectServer source. Les passerelles bidirectionnelles peuvent être considérées comme des outils de *synchronisation*.

Les passerelles peuvent envoyer des alertes à différentes cibles :

- Un autre ObjectServer
- Une base de données
- Une application de centre d'assistance
- D'autres applications ou unités

Les passerelles ObjectServer sont utilisées pour échanger des alertes entre serveurs ObjectServer. Cela est utile lorsque vous souhaitez créer une installation répartie ou lorsque vous souhaitez installer un serveur ObjectServer de secours.

Les passerelles de base de données sont utilisées pour stocker des alertes d'un serveur ObjectServer. Elles sont utiles si vous souhaitez conserver un enregistrement historique des alertes transmises au serveur ObjectServer.

Les passerelles de centre d'assistance sont utilisées pour intégrer Tivoli Netcool/OMNIbus avec une gamme de systèmes de centre d'assistance. Elles sont utiles si vous souhaitez corréler les tickets d'incident levés par vos clients avec les réseaux et systèmes que vous utilisez pour fournir leurs services.

D'autres passerelles sont des applications spécialisées qui transmettent des alertes ObjectServer à d'autres applications ou unités (par exemple, un fichier à plat ou un socket).

Remarque : Seules les passerelles qui envoient des alertes à certaines cibles peuvent être bidirectionnelles.

Concepts associés:

Chapitre 8, «A propos des passerelles», à la page 173

Les passerelles Tivoli Netcool/OMNIbus vous permettent d'échanger des alertes entre les serveurs ObjectServer et des applications de tiers complémentaires, telles que les bases de données et le système de centre d'assistance ou Customer Relationship Management (CRM).

Passerelles ObjectServer unidirectionnelles

Une passerelle ObjectServer unidirectionnelle permet aux alertes de passer d'un serveur ObjectServer source à un serveur ObjectServer cible. Les modifications apportées dans le serveur ObjectServer source sont répliquées dans le serveur ObjectServer de destination, mais les modifications apportées dans le serveur ObjectServer de destination ne sont pas répliquées dans le serveur ObjectServer source.

Pour plus d'informations sur les passerelles ObjectServer unidirectionnelles, voir le document *IBM Tivoli Netcool/OMNIbus ObjectServer Gateway Reference Guide*.

Concepts associés:

«Passerelles ObjectServer bidirectionnelles», à la page 176

Une passerelle ObjectServer bidirectionnelle permet aux alertes de passer d'un serveur ObjectServer source à un serveur ObjectServer cible. Les modifications apportées au contenu d'un serveur ObjectServer source sont répliquées dans un serveur ObjectServer de destination, qui réplique à son tour les modifications de ses alertes sur le serveur ObjectServer source. Vous pouvez, par exemple, gérer un système avec deux serveurs ObjectServer configurés comme paire de reprise en ligne.

«Passerelles ObjectServer unidirectionnelles», à la page 223

La Passerelle ObjectServer unidirectionnelle permet d'envoyer des alertes dans une direction, d'un ObjectServer source vers un ObjectServer de destination. Le nom de fichier exécutable d'une passerelle ObjectServer unidirectionnelle est `nco_g_objserv_uni`.

Passerelles ObjectServer bidirectionnelles

Une passerelle ObjectServer bidirectionnelle permet aux alertes de passer d'un serveur ObjectServer source à un serveur ObjectServer cible. Les modifications apportées au contenu d'un serveur ObjectServer source sont répliquées dans un serveur ObjectServer de destination, qui réplique à son tour les modifications de ses alertes sur le serveur ObjectServer source. Vous pouvez, par exemple, gérer un système avec deux serveurs ObjectServer configurés comme paire de reprise en ligne.

Pour plus d'informations sur les passerelles ObjectServer bidirectionnelles, voir le document *IBM Tivoli Netcool/OMNIBus ObjectServer Gateway Reference Guide*.

Concepts associés:

«Passerelles ObjectServer unidirectionnelles», à la page 175

Une passerelle ObjectServer unidirectionnelle permet aux alertes de passer d'un serveur ObjectServer source à un serveur ObjectServer cible. Les modifications apportées dans le serveur ObjectServer source sont répliquées dans le serveur ObjectServer de destination, mais les modifications apportées dans le serveur ObjectServer de destination ne sont pas répliquées dans le serveur ObjectServer source.

«Passerelles ObjectServer bidirectionnelles», à la page 221

La Passerelle ObjectServer bidirectionnelle permet d'envoyer des alertes dans les deux directions entre deux ObjectServer. Vous pouvez utiliser des passerelles bidirectionnelles pour créer une paire de reprise en ligne de serveurs ObjectServer. Le nom de fichier exécutable d'une passerelle ObjectServer bidirectionnelle est `nco_g_objserv_bi`.

Base de données, centre d'assistance et autres passerelles

La majorité des bases de données, centres d'assistance et autres passerelles utilise une architecture standard, mais chaque passerelle a son propre fichier binaire avec des modules supplémentaires pour gérer la communication avec les applications, unités ou fichiers cibles.

Pour plus d'informations sur les passerelles spécifiques et leurs architectures, voir les publications sur les passerelles individuelles.

Composants de passerelle

Les passerelles ont des composants *programme de lecture* et *programme d'écriture*. Les programmes de lecture extraient les alertes du serveur ObjectServer. Les programmes d'écriture transmettent les alertes à un autre ObjectServer ou à d'autres applications.

Il n'y a qu'un seul type de programme de lecture, mais différents types de programmes d'écriture, en fonction de l'application de destination.

Les *routes* indiquent la destination vers laquelle un programme de lecture transmet les alertes. Un programme de lecture peut avoir plusieurs routes vers différents programmes d'écriture et un programme d'écriture peut avoir plusieurs routes à partir de différents programmes de lecture.

Les *filtres* et *mappages* de passerelle configurent le flux d'alertes. Les filtres définissent les types d'alertes qui peuvent être transmis via une passerelle. Les mappages définissent le format de ces alertes.

Les programmes de lecture et d'écriture, les routes, les filtres et les mappages sont définis dans le fichier de configuration de passerelle.

Passerelles unidirectionnelles

Une passerelle de base de données, de centre d'assistance ou autre unidirectionnelle permet aux alertes de passer d'un serveur ObjectServer source à une application de destination. Les modifications apportées dans le serveur ObjectServer source sont répliquées dans l'application de destination, mais les modifications apportées dans l'application de destination ne sont pas répliquées dans le serveur ObjectServer source.

Un exemple simple de passerelle unidirectionnelle est la passerelle Fichier à plat, qui lit les alertes dans un serveur ObjectServer et les écrit dans un fichier à plat. Cet exemple d'architecture est représenté sur la figure suivante.

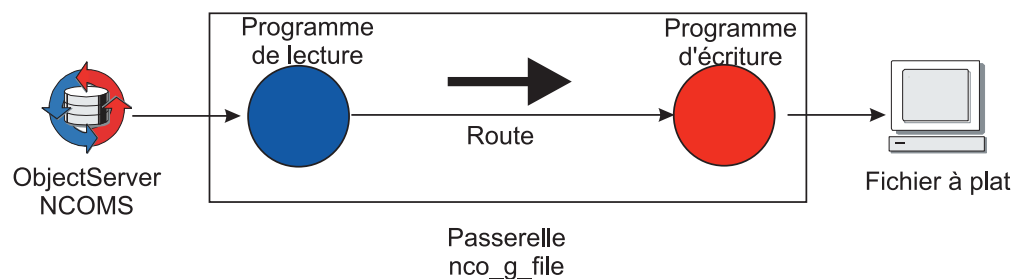


Figure 6. Exemple d'architecture de passerelles de fichier à plat

Passerelles bidirectionnelles

Dans une configuration de base de données, centre d'assistance ou passerelle bidirectionnelle, les modifications apportées à un serveur ObjectServer source sont répliquées dans une application de destination, qui réplique les modifications dans ses alertes sur le serveur ObjectServer source.

Cela vous permet, par exemple, d'élever des tickets d'incident dans un système de centre d'assistance pour certaines alertes. Les modifications apportées aux tickets du système de centre d'assistance peuvent ensuite être renvoyées au serveur ObjectServer.

Les passerelles bidirectionnelles ont une configuration similaire à celle des passerelles unidirectionnelles, avec un attribut COUNTERPART supplémentaire pour les programmes d'écriture. L'attribut COUNTERPART définit un lien entre le programme d'écriture et le programme de lecture d'une passerelle.

La figure suivante montre un exemple de configuration de passerelle bidirectionnelle.

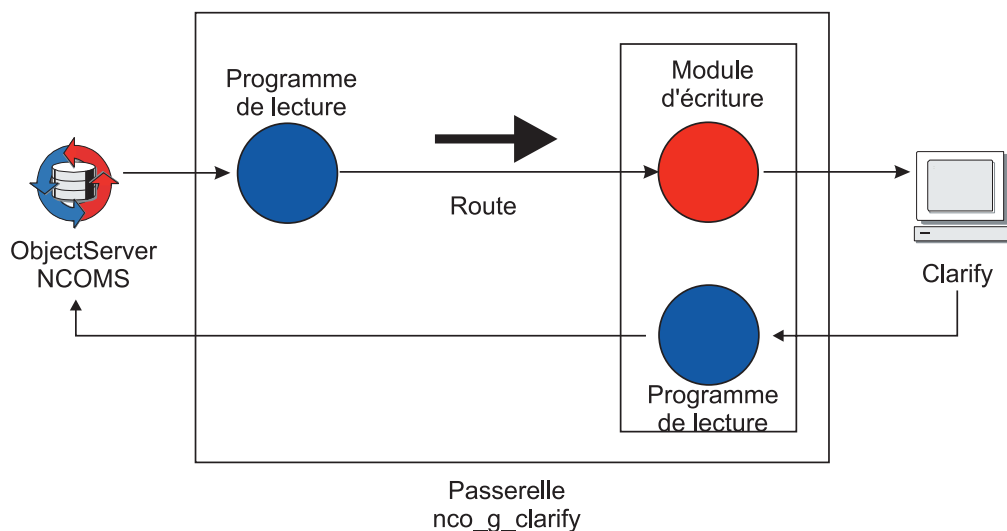


Figure 7. Passerelle Clarify bidirectionnelle

Composant programme de lecture

Un programme de lecture extrait les alertes d'un serveur ObjectServer. Il n'existe qu'un seul type de programme de lecture : le programme de lecture ObjectServer.

Au démarrage du programme de lecture, la passerelle tente d'ouvrir une connexion au serveur ObjectServer source. Si la passerelle parvient à ouvrir une connexion, elle commence immédiatement à lire les alertes du serveur ObjectServer.

Modules de programme d'écriture

Les modules de programme d'écriture gèrent les communications entre les passerelles et les applications de tiers et formatent l'alerte correctement pour l'entrer dans l'application.

Le module de programme d'écriture génère des fichiers journaux qui peuvent aider à déboguer la passerelle.

La communication entre le module de programme d'écriture et l'application de tiers utilise des applications auxiliaires, qui interagissent directement avec l'application via son API ou d'autres interfaces. Ces processus sont transparents pour l'utilisateur (bien qu'il soient visibles à l'aide de la commande **ps** ou d'un utilitaire similaire).

Le module de programme d'écriture utilise une mémoire cache de numéro de référence pour suivre les alertes et leur numéro de référence associé dans l'application cible. Pour chaque alerte, la mémoire cache stocke les informations suivantes :

- Le numéro de série de l'alerte
- Un numéro de référence de l'application cible, par exemple, un numéro de ticket de centre d'assistance

Lorsqu'un ticket est créé en réponse à une alerte, le module de programme d'écriture entre le numéro de référence dans la mémoire cache et le renvoie au serveur ObjectServer où l'alerte est mise à jour pour inclure ce numéro.

La figure suivante affiche un exemple simplifié de l'architecture du module de programme d'écriture.

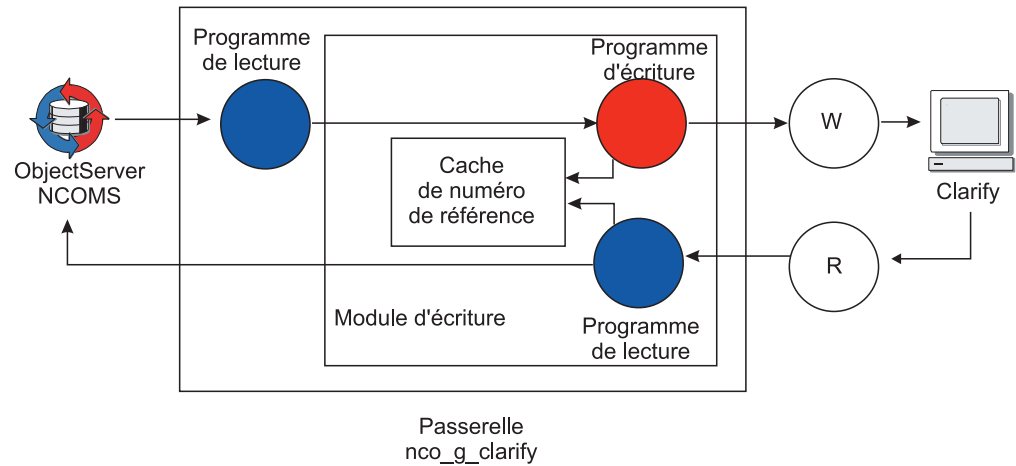


Figure 8. Architecture du module de lecture/écriture

Routes

Les routes créent la liaison entre le programme de lecture source et le programme d'écriture de destination.

Toutes les alertes reçues par le serveur ObjectServer source sont lues par le programme de lecture puis transmises au programme d'écriture via la route et écrites sur le serveur ObjectServer ou l'application de destination.

Mise à jour d'alertes à partir du centre d'assistance

Lorsqu'un opérateur de centre d'assistance apporte des modifications supplémentaires à un ticket, elles sont transmises à la passerelle qui exécute le fichier .sql d'action correspondant pour mettre à jour l'alerte sur le serveur ObjectServer.

En règle générale, les fichiers d'actions .sql suivants sont fournis :

- open.sql
- update.sql
- journal.sql
- close.sql

Pour obtenir des informations détaillées sur la configuration des mises à jour d'alertes à partir du centre d'assistance, voir les publications sur les passerelles individuelles.

Mode stocker-et-transmettre pour les passerelles

En cas de problème avec la cible de la passerelle, les programmes d'écriture du serveur ObjectServer et de la base de données peuvent continuer de fonctionner en utilisant le mode stocker-et-transmettre.

Lorsque le programme d'écriture détecte que le serveur ObjectServer ou la base de donnée cible est absent ou ne fonctionne pas (généralement car le programme

d'écriture ne peut pas écrire d'alerte), il passe en mode *stocker*. Dans ce mode, le programme d'écriture stocke toutes les données qu'il enverrait normalement à la base de données dans un fichier appelé :

```
$OMNIHOME/var/nom_prog_écriture.serveur_dest.store
```

Dans ce nom de fichier, *nom_prog_écriture* est le nom du programme d'écriture et *serveur_dest* est le nom du serveur auquel la passerelle tente d'envoyer des alertes.

Lorsque la passerelle détecte que le serveur de destination est à nouveau en ligne, elle passe en mode *transmettre* et envoie les informations d'alertes contenues dans le fichier *.store* au serveur de destination. Une fois que toutes les alertes du fichier *.store* ont été transmises, le programme d'écriture revient en fonctionnement normal.

Le mode stocker-et-transmettre fonctionne uniquement lorsqu'une connexion au serveur ObjectServer ou à la base de données de destination a été établie, utilisée puis perdue. Si le serveur de destination n'est pas en cours d'exécution au démarrage de la passerelle, le mode stocker-et-transmettre n'est pas déclenché et la passerelle s'arrête.

Si la passerelle se connecte au serveur ObjectServer de destination et qu'un fichier stocker-et-transmettre existe déjà, la passerelle relit le contenu du fichier avant d'envoyer de nouvelles alertes.

Le mode stocker-et-transmettre est configuré à l'aide des attributs `STORE_AND_FORWARD` et `STORE_FILE`.

Remarque : Voir les publications sur les passerelles individuelles pour déterminer si une passerelle prend en charge le mode stocker-et-transmettre. Ce mode ne fonctionne pas avec les configurations de passerelle bidirectionnelles, à l'exception de la passerelle ObjectServer bidirectionnelle.

Mode sécurisé pour les passerelles

Lorsque vous démarrez le serveur ObjectServer à l'aide de l'option de ligne de commande `-secure`, celui-ci authentifie les connexions de sonde, de passerelle et de serveur proxy en demandant un nom d'utilisateur et un mot de passe.

Lorsqu'une demande de connexion est envoyée, le serveur ObjectServer émet un message d'authentification. La sonde, la passerelle ou le serveur proxy doit répondre avec le nom d'utilisateur et le mot de passe corrects. Si la combinaison nom d'utilisateur-mot de passe est incorrecte, le serveur ObjectServer émet un message d'erreur et refuse la connexion.

Si le serveur ObjectServer n'est pas exécuté en mode sécurisé, les demandes de connexion de la sonde, de la passerelle et du serveur proxy ne sont pas authentifiées.

L'authentification sécurisée est configurée différemment pour les passerelles qui utilisent des fichiers de configuration uniques (fichiers *.conf*) et celles qui utilisent des fichiers de propriétés (fichiers *.props*). Reportez-vous aux guides de référence de chaque passerelle pour plus d'informations sur sa configuration.

Vous pouvez faire appel à l'utilitaire `nco_g_crypt` pour chiffrer les mots de passe à utiliser dans les fichiers de configuration et de propriétés. Si vous exécutez Tivoli

Netcool/OMNIbus en mode FIPS 140-2, utilisez l'utilitaire **nco_aes_crypt** pour chiffrer les mots de passe. Les mots de passes chiffrés sont déchiffrés par la passerelle avant d'être utilisés pour la connexion au système cible. Reportez-vous aux guides de référence de chaque passerelle pour plus d'informations sur le chiffrement des mots de passe pour celle-ci.

Pour plus d'informations sur l'exécution du serveur ObjectServer et de la passerelle ObjectServer en mode sécurisé, voir les documents *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus* et *IBM Tivoli Netcool/OMNIbus ObjectServer Gateway Reference Guide*.

Programmes d'écriture et reprise par restauration de la passerelle

La reprise en ligne se produit lorsqu'une passerelle perd sa connexion avec le serveur ObjectServer principal et se connecte à un serveur ObjectServer de secours. La fonction de reprise par restauration permet à la passerelle de se reconnecter au serveur ObjectServer principal lorsqu'il est à nouveau actif.

Le programme de lecture du serveur ObjectServer peut reprendre en ligne ou reprendre par restauration entre des serveurs ObjectServer source sans être arrêté. Cette fonction n'est pas prise en charge par tous les programmes d'écriture des passerelles.

Si un programme d'écriture ne prend pas en charge ce mode de reprise par restauration et reprise en ligne, il arrête la passerelle lors de la détection d'une reprise par restauration ou d'une reprise en ligne du programme de lecture et compte sur l'agent de processus pour redémarrer la passerelle.

Les programmes d'écriture suivants prennent en charge la reprise en ligne et la reprise par restauration du programme de lecture sans arrêt :

- Programme d'écriture ObjectServer
- Programme d'écriture de la base de données Sybase
- Programme d'écriture Sybase Reporter
- Programme d'écriture SNMP
- Programme d'écriture de Socket
- Programme d'écriture de Fichier à plat
- Programme d'écriture de base de données Informix
- Programme d'écriture de bus de messages (XML)
- Programme d'écriture JDBC
- Programme d'écriture Tivoli EIF
- Programme d'écriture TSRM
- Programme d'écriture JDBC

Les programmes d'écriture suivants prennent en charge la reprise en ligne et la reprise par restauration avec arrêt :

- Programme d'écriture Remedy ARS
- Programme d'écriture Siebel eCommunications
- Programme d'écriture de base de données Oracle
- Programme d'écriture Oracle Reporter
- Programme d'écriture Peregrine
- Programme d'écriture Clarify

- Programme d'écriture HP Service Desk
- Programme d'écriture de base de données ODBC
- Programme d'écriture JDBC

Etant donné que les passerelles ObjectServer bidirectionnelles sont utilisées pour resynchroniser les paires de reprise en ligne, la reprise par restauration est automatiquement désactivée. Cela est dû au fait qu'une moitié de la passerelle peut se connecter de manière légitime à un serveur de secours et ne doit pas être obligée d'effectuer la reprise par restauration sur le serveur ObjectServer principal. Toutefois, si une passerelle bidirectionnelle est utilisée pour partager des données entre deux sites distincts et que chaque site dispose d'une paire de reprise en ligne qui fonctionne, vous pouvez manuellement activer la reprise par restauration sur chaque serveur. Lorsqu'il est activé, le programme d'écriture active automatiquement la reprise par restauration sur le programme d'écriture homologue.

Tâches associées:

«Configuration des opérations de reprise par restauration pour les passerelles ObjectServer bidirectionnelles», à la page 230

Si les données sont modifiées sur un seul serveur ObjectServer d'une paire virtuelle, vous pouvez configurer une passerelle bidirectionnelle pour la reprise par restauration. L'opération est identique à celle d'une passerelle unidirectionnelle mais vous devez configurer des propriétés différentes.

«Configuration des opérations de reprise par restauration pour les passerelles ObjectServer unidirectionnelles», à la page 229

Deux serveurs ObjectServer peuvent être configurés en tant que paire, un serveur jouant le rôle de serveur principal et l'autre serveur étant le serveur de sauvegarde. Vous pouvez spécifier comment l'ObjectServer de sauvegarde effectue la reprise par restauration sur l'ObjectServer principal.

Chapitre 9. Configuration de passerelles

Les fichiers de configuration définissent l'environnement dans lequel les passerelles fonctionnent et la manière dont elles mappe les données entre les tables ObjectServer et les bases de données ou applications cible. La plupart des passerelles utilisent plusieurs fichiers de configuration. Certaines passerelles sont configurées à l'aide d'un seul fichier de configuration.

La plupart des passerelles, dont les passerelles ObjectServer, sont configurées à l'aide d'un fichier de propriétés (*nom_passerelle.props*), d'un fichier de définition de mappe (*nom_passerelle.map*), d'un fichier de commandes de démarrage (*nom_passerelle.startup.cmd*) et d'un fichier de définition de réplication de table (*nom_passerelle.tblrep.def*).

Les passerelles suivantes sont configurées à l'aide d'un seul fichier de configuration avec l'extension `.conf` :

- Gateway for HP ServiceCenter
- Gateway for Remedy ARS
- Gateway for Siebel

Utilisation de fichiers de configuration multiples

La plupart des passerelles sont configurées à l'aide d'un fichier de propriétés, d'un fichier de définition de mappe, d'un fichier de commandes de démarrage et d'un fichier de définition de réplication de table.

Avant d'exécuter une passerelle, vous devez éditer différents fichiers de configuration en fonction de votre environnement Tivoli Netcool/OMNIBus.

Le fichier des propriétés est un fichier texte qui contient un ensemble de propriétés et les valeurs correspondantes. Ces propriétés définissent l'environnement opérationnel de la passerelle, notamment les détails de connexion et l'emplacement des autres fichiers de configuration.

Au démarrage, la passerelle recherche le fichier de propriétés par défaut `$OMNIHOME/etc/NCO_GATE.props`. Reportez-vous à la documentation de référence de chaque passerelle pour plus d'informations sur la configuration de son fichier de propriétés.

Le fichier de définition de mappe et le fichier de définition de réplication de table définissent la manière dont la passerelle transfère les données entre les serveurs ObjectServer et les bases de données ou applications cible. Le fichier de commandes de démarrage vous permet de configurer l'exécution automatique d'opérations sur les données au démarrage de la passerelle.

Référence associée:

«Propriétés et options de ligne de commande de passerelle communes», à la page 192

Les passerelles disposent de plusieurs propriétés communes et des options de ligne de commande associées. Ces propriétés définissent des paramètres pour des fonctions génériques (telles que la consignation de messages), la communication interprocessus (IPC) et des paramètres de passerelle communes, par exemple, des fichiers de définition de mappe. Vous pouvez remplacer les valeurs de propriété par défaut en éditant le fichier de propriétés ou à l'aide d'options de ligne de commande lorsque vous démarrez la passerelle.

Fichier de définition de mappe

Le fichier de définition de mappe définit la manière dont la passerelle mappe les zones d'alerte (colonnes) d'une table ObjectServer à une base de données ou une application cible. Le fichier de définition de mappe est référencé par le fichier de définition de réplication de table, le fichier de commandes de démarrage et l'interface interactive SQL.

Le fichier de définition de mappe par défaut est `$OMNIHOME/gates/nom_passerelle/nom_passerelle.map`.

Par exemple, le fichier de définition de mappe de la passerelle Gateway for JDBC est `$OMNIHOME/gates/jdbc/jdbc.map`.

Le nom et l'emplacement du fichier sont spécifiés dans le fichier de propriétés de la passerelle par la propriété **Gate.MapFile**.

Le fichier de définition de mappe peut contenir une ou plusieurs commandes CREATE MAPPING qui utilisent la syntaxe suivante : Les différents mappages de zones sont séparés par une virgule (,).

```
CREATE MAPPING nom_mappe
(
  'zone_destination' = {'@zone_source'|expression_simple|attribut}
  [ON INSERT ONLY]
  [CONVERT TO {INTEGER|STRING|DATE}]
  [NOT NULL '@zone_source'],
  ...
);
```

La commande CREATE MAPPING est structurée comme suit :

- *nom_mappe* est le nom de la définition de mappe.
- *zone_destination* est le nom de la zone de destination dans la base de données ou l'application cible.
- *zone_source* est le nom d'une zone d'alerte (colonne) dans la table ObjectServer.
- *expression_simple* représente des données de type chaîne, entier ou booléen, ou une séquence de chaînes ou d'entiers joints à l'aide des opérateurs suivants : : +, -, *, /. Tous les opérateurs concatènent des chaînes. Tous les opérateurs fonctionnent de gauche à droite lorsqu'ils opèrent sur des entiers.
- *attribut* est un attribut de mappage.

Le tableau suivant répertorie les clauses de la commande CREATE MAPPING.

Tableau 38. Clauses de la commande CREATE MAPPING

Clause	Description
ON INSERT ONLY	Cette clause contrôle la manière dont la zone est mise à jour pendant la durée de vie d'une alerte. Si cette clause est omise, la zone est mise à jour chaque fois que l'état d'une alerte change. Si elle est incluse, la zone est créée une seule fois pour une alerte et n'est jamais mise à jour.
CONVERT TO {INTEGER STRING DATE}	Cette clause définit une conversion forcée si les données d'une zone source ne correspondent pas au type de données de la zone de destination. Les paramètres disponibles sont INTEGER, STRING et DATE.
NOT NULL '@zone_source'	Cette clause fournit une valeur <i>zone_source</i> de remplacement à utiliser lorsque la valeur source d'origine est zéro ou une chaîne vide. Cette autre valeur peut être une zone (colonne) ou une constante, mais pas une expression.

Fonctions de conversion

Vous pouvez utiliser des fonctions de conversion dans votre définition de mappe, pour convertir des types de données source en types de données STRING (chaîne), INTEGER (entier) ou TIME (date et heure).

Les fonctions de conversion utilisent la syntaxe suivante :

- TO_STRING('@zone_source')
- TO_INTEGER('@zone_source')
- TO_TIME('@zone_source')

Attributs de mappe

Vous pouvez utiliser des noms d'attribut pour inclure des données supplémentaires dans les définitions de mappage. Vous pouvez indiquer deux types d'attributs : des attributs d'accès aux valeurs du cache ou des attributs dynamiques.

La passerelle utilise des attributs de valeurs du cache pour accéder aux valeurs qui sont stockées dans le cache des références croisées. Le tableau suivant décrit les attributs de valeurs de cache utilisables dans les définitions de mappe.

Tableau 39. Attributs d'accès aux valeurs du cache

Nom d'attribut	Description
STATUS.SERIAL	Numéro de série mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
STATUS.SERVER_SERIAL	Numéro de série du serveur mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.

Tableau 39. Attributs d'accès aux valeurs du cache (suite)

Nom d'attribut	Description
STATUS.SERVER_NAME	Nom du serveur mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
STATUS.IDENTIFIER	Identificateur mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
JOURNAL.SERIAL	Numéro de série de la ligne de la table des journaux mis en cache.
DETAILS.IDENTIFIER	Numéro de série de la ligne de la table des détails mis en cache.

Les attributs dynamiques permettent à la passerelle d'accéder aux valeurs dynamiques qu'elle génère automatiquement. Le tableau suivant décrit les attributs dynamiques utilisables dans les définitions de mappe.

Tableau 40. Attributs dynamiques

Nom d'attribut	Description
ACTION_CODE	Cet attribut affiche une chaîne d'un seul caractère spécifiant le type d'opération réalisé. Les valeurs admises sont les suivantes : <ul style="list-style-type: none"> • I : Insérer • U : Mettre à jour • D : Supprimer
ACTION_TIME	Cet attribut affiche l'heure UTC (temps universel coordonné) à laquelle l'action s'est produite.
DELETEDAT	Cet attribut affiche la date à laquelle la ligne a été effacée, le cas échéant.

Exemple de fichier de mappe

Voici un exemple d'extrait de fichier de définition de mappe qui mappe les données entre deux serveurs ObjectServer :

```
CREATE MAPPING StatusMap
(
'Identifiant' = '@Identifiant' ON INSERT ONLY'
'Node' = '@Node' ON INSERT ONLY,
'NodeAlias' = '@NodeAlias' ON INSERT ONLY NOT NULL '@Node',
'Manager' = '@Manager' ON INSERT ONLY,
'Agent' = '@Agent' ON INSERT ONLY,
'AlertGroup' = '@AlertGroup' ON INSERT ONLY,
'AlertKey' = '@AlertKey' ON INSERT ONLY,
'Severity' = '@Severity',
'Summary' = '@Summary',
'StateChange' = '@StateChange'
);

CREATE MAPPING JournalMap
(
'KeyField' = TO_STRING(STATUS.SERIAL) + ":" +
TO_STRING('@UID') + ":" +
TO_STRING('@Chrono') ON INSERT ONLY,
'Serial' = STATUS.SERIAL,
'Chrono' = '@Chrono',
```

```

'UID' = TO_INTEGER('@UID'),
'Text1' = '@Text1',
'Text2' = '@Text2'
);

CREATE MAPPING DetailsMap
(
'KeyField' = '@Identifiant' + '###' +
TO_STRING('@Sequence') ON INSERT ONLY,
'Identifiant' = '@Identifiant',
'AttrVal' = '@AttrVal',
'Sequence' = '@Sequence',
'Name' = '@Name',
'Detail' = '@Detail'
);

```

Fichier de définition de réplication de table

Le fichier de définition de réplication de table définit la façon dont les données sont répliquées entre un serveur ObjectServer et une base de données ou une application cible.

Le fichier de définition de réplication de table par défaut est \$OMNIHOME/gates/*nom_passerelle/nom_passerelle.tblrep.def*. Le nom et l'emplacement du fichier sont spécifiés dans le fichier de propriétés de la passerelle. Le nom de la propriété est différent selon la passerelle.

Par exemple, le fichier de définition de réplication de table de la passerelle Gateway for JDBC est spécifié par la propriété **Gate.RdrWtr.TblReplicateDefFile**. Le fichier par défaut est \$OMNIHOME/gates/jdbc/jdbc.rdrwtr.tblrep.def.

Le fichier de définition de réplication de table peut contenir une ou plusieurs commandes REPLICATE qui utilisent la syntaxe suivante :

```

REPLICATE {ALL|INSERTS, UPDATES, DELETES, FT_INSERTS, FT_UPDATES, FT_DELETES}
FROM TABLE 'table_source'
USING MAP 'nom_mappe'
[FILTER WITH 'filtre']
[INTO 'table_cible']
[ORDER BY 'chaîne_ordre']
[WITH NORESYNC]
[RESYNC DELETES FILTER 'filtre']
[SET UPDTOINS CHECK TO {ENABLED|DISABLED|FORCED}]
[AFTER IDUC DO 'commande_mise_à_jour']
[CACHE FILTER 'condition'];

```

Utilisez les clauses facultatives dans l'ordre dans lequel elles sont répertoriées dans la description de la syntaxe. Par exemple, lorsque vous utilisez les clauses FILTER WITH et AFTER IDUC DO, la clause FILTER WITH doit être placée avant la clause AFTER IDUC DO. Si vous utilisez la clause CACHE FILTER, il doit s'agir de la dernière clause de la commande REPLICATE.

Les clauses disponibles pour la commande REPLICATE sont décrites dans le tableau suivant.

Tableau 41. Clauses de la commande REPLICATE

Clause	Description
ALL	<p>Utilisez cette clause pour demander que toutes les insertions, mises à jour et suppressions sur la table source soient répliquées dans la table cible. Vous pouvez restreindre la réplication à l'aide d'une ou plusieurs clauses INSERTS, UPDATES, DELETES, FT_INSERTS, FT_UPDATES et FT_DELETES au lieu de la clause ALL.</p> <p>Par exemple, la commande suivante réplique les insertions et les mises à jour, mais pas les suppressions. En ne répliquant pas les suppressions, vous pouvez réduire la charge des ressources mémoire sur la base de données cible.</p> <p>REPLICATE INSERTS, UPDATES</p>
FROM TABLE ' <i>table_source</i> '	Utilisez cette clause pour indiquer la source de données, <i>table_source</i> étant la table à répliquer dans la base de données cible.
USING MAP ' <i>nom_mappe</i> '	Utilisez cette clause pour indiquer le mappage à utiliser, <i>nom_mappe</i> étant la définition de mappe qui définit la table source.
FILTER WITH ' <i>filtre</i> '	<p>Facultative. Utilisez cette clause pour filtrer les lignes de base de données qui sont sélectionnées pour la réplication, <i>filtre</i> définissant le filtre utilisé par la passerelle.</p> <p>Conseil : Pour répliquer des événements qui ne sont pas égaux à une valeur particulière, placez un point d'exclamation (!) avant le signe égal (=) dans la clause de filtrage. Par exemple, la clause suivante va répliquer tous les événements dont le niveau de gravité n'est pas égal à 5 :</p> <p>FILTER WITH 'Severity !=5'</p>
INTO ' <i>table_cible</i> '	<p>Facultative. Utilisez cette clause pour indiquer la table de la base de données cible vers laquelle les données sont répliquées, <i>table_cible</i> étant le nom de la table cible.</p> <p>Si cette clause est omise, la commande REPLICATE utilise le nom de la table source comme valeur de table cible. Par conséquent, cette option n'est pas obligatoire lorsque vous répliquez les tables ObjectServer alerts.staus, alerts.details et alerts.journal principales. Cette clause est prévue pour la réplication d'autres tables ObjectServer vers la base de données cible, mais ce type de transfert de données est normalement effectué à l'aide de la commande TRANSFER.</p>

Tableau 41. Clauses de la commande REPLICATE (suite)

Clause	Description
ORDER BY ' <i>chaîne_ordre</i> '	<p>Facultative. Utilisez cette clause pour spécifier l'ordre dans lequel les lignes sont renvoyées à la passerelle depuis le serveur ObjectServer, <i>chaîne_ordre</i> étant une liste de noms de colonne séparés par des virgules. Par exemple :</p> <p>ORDER BY 'Serial DESC, StateChange ACS'</p> <p>Vous pouvez utiliser les clauses facultatives ASC et DESC pour demander que les lignes soient triées par ordre croissant (ASC) ou décroissant (DESC). Si vous ne spécifiez d'ordre de tri, les lignes sont triées par ordre croissant.</p>
WITH NORESYNC	Facultative. Utilisez cette clause pour indiquer les tables à ne pas resynchroniser.
RESYNC DELETES FILTER ' <i>filtre</i> '	Facultative. Utilisez cette clause pour indiquer lignes répliquées à supprimer avant l'insertion dans la table cible, <i>filtre</i> définissant le filtre de suppression de resynchronisation. Ce filtre est utilisé lorsque les lignes des tables cible et source ne sont pas une correspondance exacte.
SET UPDTOINS CHECK TO {ENABLED DISABLED FORCED}	<p>Facultative. Utilisez cette clause pour configurer le comportement de mise à jour en insertion. Elle comporte les paramètres suivants :</p> <ul style="list-style-type: none"> • ENABLED : Il s'agit du paramètre par défaut. La passerelle exécute des conversions de mise à jour en insertion normales. Si une mise à jour reçue de la table source contient une ligne qui n'existe pas dans la table cible, la mise à jour est convertie en insertion et la ligne est écrite dans la table cible. • DISABLED : Pour chaque mise à jour reçue de la table source, la passerelle envoie une mise à jour à la table cible. Si la mise à jour contient des lignes inexistantes dans la table cible, ces lignes sont supprimées. • FORCED : La passerelle convertit toutes les mises à jour reçues de la table source en insertion sur la table cible. Si la ligne existe déjà dans la table cible, elle est dédoublée. Ce comportement est identique au comportement des sondes.
AFTER IDUC DO ' <i>commande_mise_à_jour</i> '	Facultative. Utilisez cette clause pour modifier des lignes sur le serveur ObjectServer source immédiatement après qu'elles ont été extraites par la passerelle. <i>commande_mise_à_jour</i> est une liste d'affectations de colonne séparées par des virgules. La mise à jour est appliquée à toutes les lignes qui sont insérées, mises à jour ou supprimées.

Tableau 41. Clauses de la commande REPLICATE (suite)

Clause	Description
CACHE FILTER ' <i>condition</i> '	<p>Facultative. Utilisez cette clause pour filtrer les entrées de cache qui sont extraites lors d'une actualisation du cache. <i>condition</i> définit la condition SQL que la passerelle ajoute à l'instruction SELECT qui permet d'extraire les entrées de cache.</p> <p>Utilisez cette clause pour réduire la quantité de données extraite par la passerelle durant une actualisation de cache de la passerelle unidirectionnelle.</p>

Voici un exemple de fichier de définition de réplication de table :

```

REPLICATE INSERTS, DELETES FROM TABLE 'alerts.status'
  USING MAP 'StatusMap'
  ORDER BY 'Serial' ASC
  FILTER WITH 'Severity !=5'
  SET UPDPOINTS CHECK TO FORCED
  AFTER IDUC DO 'Location=\PASSED BY GW\'
  CACHE FILTER 'ServerName IN (\'NCOMBS_P\',\'NCOMBS_B\')';

REPLICATE ALL FROM TABLE 'alerts.journal' USING MAP 'JournalMap';

REPLICATE ALL FROM TABLE 'alerts.details' USING MAP 'DetailsMap';

```

Fichier de commandes de démarrage

Le fichier de commandes de démarrage contient un ensemble de commandes exécutées automatiquement par la passerelle lors de chaque démarrage. Vous pouvez utiliser ces commandes pour transférer des données d'une table ObjectServer vers une base de données cible.

Le fichier de commandes de démarrage par défaut est \$OMNIHOME/gates/*nom_passerelle/nom_passerelle.startup.cmd*. Par exemple, le fichier de commandes de démarrage de la passerelle Gateway for JDBC est \$OMNIHOME/gates/jdbc/jdbc.startup.cmd.

Le nom et l'emplacement du fichier sont spécifiés dans le fichier de propriétés de la passerelle par la propriété **Gate.StartupCmdFile**.

Les commandes de démarrage suivantes sont disponibles :

- SET PROPERTY : Cette commande permet de définir la valeur d'une propriété dans le fichier de propriétés de passerelle.

Syntaxe	SET PROPERTY ' <i>nom_propriété</i> ' TO <i>valeur</i> ;
Exemple	SET PROPERTY 'Gate.NGtkDebug' TO FALSE;

- GET PROPERTY : Cette commande permet de renvoyer la valeur d'une propriété spécifiée depuis le fichier de propriétés de passerelle.

Syntaxe	GET PROPERTY ' <i>nom_propriété</i> ';
Exemple	GET PROPERTY 'Gate.NGtkDebug';

- SHOW PROPS : Cette commande permet d'afficher la configuration en cours de la passerelle en répertoriant toutes les propriétés et leurs valeurs.

Syntaxe	SHOW PROPS;
Exemple	SHOW PROPS;

- GET CONFIG : Cette commande permet d'afficher la configuration en cours de la passerelle en répertoriant toutes les propriétés et leurs valeurs. GET CONFIG fournit la même fonction que la commande SHOW PROPS.

Syntaxe	GET CONFIG;
Exemple	GET CONFIG;

- SET LOG : Cette commande permet de définir le niveau de consignation de messages pour la passerelle. Les niveaux disponibles sont debug, info, warn, error et fatal. Le niveau par défaut est warn.

Syntaxe	SET LOG LEVEL TO <i>niveau_message</i> ;
Exemple	SET LOG LEVEL TO debug;

- TRANSFER : Cette commande permet de transférer des données entre une table ObjectServer et une table de base de données cible. Vous pouvez ajouter plusieurs commandes TRANSFER dans le fichier de commandes de démarrage. Vous pouvez utiliser une condition de filtre pour transférer une table partielle. Vous pouvez également indiquer une mappe dans le cadre de l'opération de transfert. Cette mappe doit être définie dans le fichier de définition de mappe (*nom_passerelle.map*).

Syntaxe	TRANSFER FROM ' <i>table_source</i> ' TO ' <i>table_cible</i> ' [VIA FILTER ' <i>filtre</i> '] [WITH DELETE VIA ' <i>filtre_suppression</i> '] [USING TRANSFER_MAP ' <i>nom_mappe</i> '];
Exemple	TRANSFER FROM 'master.names' TO 'resync.names' VIA FILTER 'Name != \'personne\'' DELETE; Dans cet exemple, toutes les lignes qui contiennent le nom <i>personne</i> sont supprimées de la table <i>resync.names</i> et remplacées par les lignes correspondantes de la table <i>master.names</i> . Des barres obliques inversées (\) doivent être utilisées comme caractères d'échappement pour les apostrophes (') entourant <i>personne</i> .

- FAILOVER_SYNC : Cette commande permet de synchroniser les données entre les serveurs ObjectServer principal et de secours. La commande spécifie les tables maîtres transférées durant l'opération de transfert de données. Vous pouvez ajouter plusieurs commandes FAILOVER_SYNC dans le fichier de commandes de démarrage.

Syntaxe	FAILOVER_SYNC ADD REMOVE ' <i>nom_table</i> ' TO <i>nom_programme_écriture</i> ;
Exemple	FAILOVER_SYNC ADD 'master.names' TO ObjectServerA; FAILOVER_SYNC ADD 'master.groups' TO ObjectServerA; FAILOVER_SYNC ADD 'master.members' TO ObjectServerA; FAILOVER_SYNC ADD 'master.permissions' TO ObjectServerA; FAILOVER_SYNC ADD 'master.profiles' TO ObjectServerA;

Voici un exemple de fichier de commandes de démarrage :

```

GET CONFIG;
SET LOG LEVEL TO debug;
TRANSFER FROM 'master.names' TO 'resync.names' VIA FILTER
'Name != \'personne\' ' DELETE;

```

Propriétés et options de ligne de commande de passerelle communes

Les passerelles disposent de plusieurs propriétés communes et des options de ligne de commande associées. Ces propriétés définissent des paramètres pour des fonctions génériques (telles que la consignation de messages), la communication interprocessus (IPC) et des paramètres de passerelle communs, par exemple, des fichiers de définition de mappe. Vous pouvez remplacer les valeurs de propriété par défaut en éditant le fichier de propriétés ou à l'aide d'options de ligne de commande lorsque vous démarrez la passerelle.

Le tableau ci-dessous répertorie les propriétés et les options de ligne de commande de passerelle communes. Toutes les passerelles n'utilisent pas toutes les propriétés qui y sont indiquées. Reportez-vous à la documentation de référence de chaque passerelle pour plus d'informations sur ses propriétés.

Tableau 42. Propriétés et options de ligne de commande de passerelle communes

Nom de propriété	Option de ligne de commande	Description
ConfigCryptoAlg chaîne	-configcryptoalg chaîne	Cette propriété permet de spécifier l'algorithme de chiffrement utilisé par la passerelle. Utilisez cette propriété avec la propriété ConfigKeyFile et l'utilitaire nco_aes_crypt fourni avec Netcool/OMNIbus. La valeur par défaut est AES.
ConfigKeyFile chaîne	-configkeyfile chaîne	Cette propriété permet de spécifier la clé de chiffrement utilisée pour chiffrer les données. Utilisez cette propriété avec la propriété ConfigCryptoAlg et l'utilitaire nco_aes_crypt fourni avec Netcool/OMNIbus. La valeur par défaut est "".
Connections entier	-connections entier	Cette propriété permet de spécifier le nombre maximal de connexions client qui peuvent être établies avec le serveur de passerelle. La valeur par défaut est 30.

Tableau 42. Propriétés et options de ligne de commande de passerelle communes (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.CacheHashTblSize entier	-cachehtblsize entier	Cette propriété permet de spécifier le nombre d'éléments que la passerelle attribue pour la mémoire cache de table de hachage. La valeur par défaut est 5023.
Gate.MapFile chaîne	-mapfile chaîne	Cette propriété permet de spécifier le fichier de mappage utilisé par la passerelle. Le fichier par défaut est \$OMNIHOME/gates/nom_passerelle/nom_passerelle.map.
Gate.NGtkDebug booléen	-ngtkdebug booléen	Cette propriété permet d'activer la consignation des messages de débogage de bibliothèque NGTK. La valeur par défaut est TRUE.
Gate.PAAware entier	-paaware entier	Cette propriété indique si la passerelle est activée pour l'agent de processus (PA). Cette propriété est tenue à jour par le serveur d'agent de processus et est incluse dans le fichier de propriétés pour information uniquement. La valeur par défaut est 0 (non activée pour l'agent de processus).
Gate.PAAwareName chaîne	-paname chaîne	Cette propriété indique le nom de l'agent de processus qui contrôle la passerelle. Cette propriété est tenue à jour par le serveur d'agent de processus et est incluse dans le fichier de propriétés pour information uniquement. La valeur par défaut est "".

Tableau 42. Propriétés et options de ligne de commande de passerelle communes (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.StartupCmdFile chaîne	-startupcmdfile chaîne	Cette propriété permet de spécifier l'emplacement du fichier de commandes de démarrage. L'emplacement par défaut est \$OMNIHOME/gates/ <i>nom_passerelle</i> / <i>nom_passerelle</i> .startup.cmd.
Gate.Transfer.FailoverSyncRate entier	-fsyncrate entier	Cette propriété permet de spécifier le débit (en secondes) de la synchronisation de reprise en ligne. La valeur par défaut est 60.
Gate.UnixAdminGrp chaîne	-unixadmingrp chaîne	Cette propriété permet de spécifier le groupe d'administration auquel la passerelle appartient si l'authentification UNIX standard est utilisée. Valeur par défaut : ncoadmin.
Gate.UsePamAuth booléen	-usepamauth booléen	Cette propriété permet de spécifier si l'authentification PAM est utilisée. Pour exécuter la passerelle en mode FIPS 140-2, paramétrez cette propriété sur TRUE. La valeur par défaut est FALSE.
Help booléen	-help booléen	Cette propriété permet de spécifier que la passerelle doit afficher une aide lorsqu'elle démarre et s'arrête. La valeur par défaut est FALSE.
Ipc.Timeout entier	-ipctimeout entier	Cette propriété permet de spécifier la durée (en secondes) pendant laquelle le client attend une réponse du serveur. Si ce temps est dépassé, une erreur est consignée. La valeur par défaut est 60.

Tableau 42. Propriétés et options de ligne de commande de passerelle communes (suite)

Nom de propriété	Option de ligne de commande	Description
MaxLogFileSize <i>entier</i>	<code>-maxlogfilesize</code> <i>entier</i>	Cette propriété permet de spécifier la taille maximale (en octets) que la passerelle attribue au fichier journal. La valeur par défaut est 1024.
MessageLevel <i>chaîne</i>	<code>-messagelevel</code> <i>chaîne</i>	Cette propriété permet de spécifier le niveau de génération de rapports pour les messages de fichier journal de passerelle. La valeur par défaut est warn.
MessageLog <i>chaîne</i>	<code>-messagelog</code> <i>chaîne</i>	Cette propriété permet de spécifier l'emplacement du fichier journal des messages de passerelle. La valeur par défaut est \$OMNIHOME/log/NC0_GATE.log.
Name <i>chaîne</i>	<code>-name</code> <i>chaîne</i>	Cette propriété permet de spécifier le nom de l'instance de passerelle en cours. Si vous souhaitez exécuter plusieurs passerelles sur un seul ordinateur, vous devez utiliser un nom différent pour chaque instance. La valeur par défaut est NC0_GATE.
Props.CheckNames <i>booléen</i>	N/D	Cette propriété permet d'ordonner à la passerelle de se fermer si une propriété du fichier de propriétés est définie sur une valeur non valide. La valeur par défaut est TRUE.
PropsFile <i>chaîne</i>	<code>-propsfile</code> <i>chaîne</i>	Spécifie l'emplacement du fichier de propriétés de la passerelle. La valeur par défaut est \$OMNIHOME/etc/NC0_GATE.props.

Tableau 42. Propriétés et options de ligne de commande de passerelle communes (suite)

Nom de propriété	Option de ligne de commande	Description
UniqueLog <i>booléen</i>	-uniquelog <i>booléen</i>	Cette propriété permet de spécifier que les noms des fichiers journaux sont rendus uniques par l'ajout du PID de la passerelle au nom de fichier. La valeur par défaut est FALSE.
Version <i>booléen</i>	-version <i>booléen</i>	Cette propriété permet de spécifier que la passerelle doit afficher les informations de version lorsqu'elle démarre et s'arrête. La valeur par défaut est FALSE.

Concepts associés:

«Utilisation de fichiers de configuration multiples», à la page 183

La plupart des passerelles sont configurées à l'aide d'un fichier de propriétés, d'un fichier de définition de mappe, d'un fichier de commandes de démarrage et d'un fichier de définition de réplication de table.

Utilisation de commandes sur des passerelles en cours d'exécution

Vous pouvez utiliser l'interface interactive SQL pour émettre des commandes vers une passerelle en cours d'exécution.

Remarque : Si vous exécutez Tivoli Netcool/OMNIbus en mode FIPS 140-2, vous ne pouvez pas utiliser l'interface interactive SQL pour exécuter des commandes sur des passerelles.

Vous pouvez exécuter les commandes qui sont prises en charge par le fichier de commandes de démarrage de passerelle, par exemple, SET PROPERTY et TRANSFER.

Utilisez la commande suivante pour vous connecter à une passerelle :

- Sous UNIX et Linux : `$OMNIHOME/bin/nco_sql -server nom_serveur -user nom_utilisateur`
- Sous Windows : `%OMNIHOME%\bin\isql -S nom_serveur -U nom_utilisateur`

Où *nom_serveur* correspond au nom de la passerelle configuré dans l'éditeur de serveur et *nom_utilisateur* représente un nom d'utilisateur valide.

Sur les systèmes d'exploitation UNIX et Linux, les exigences suivantes s'appliquent lorsque vous exécutez des commandes vers des passerelles avec l'interface interactive SQL :

- Si vous définissez la propriété **Gate.UsePamAuth** sur FALSE, l'utilisateur exécutant le processus de passerelle doit disposer du droit de lecture sur la base de données utilisateur. En fonction de la configuration du système d'exploitation, l'utilisateur doit disposer du droit de lecture sur les fichiers suivants :
etc/passwd, etc/shadow et etc/group.

Vous devez également définir la propriété **Gate.UnixAdminGrp** sur le groupe d'administration des utilisateurs UNIX auquel le compte utilisateur de la passerelle appartient.

- Si vous définissez la propriété **Gate.UsePamAuth** sur TRUE, vous devez configurer le service PAM de passerelle pour les modules auth et account.

La passerelle peut également avoir besoin du droit de lecture sur les bases de données utilisées par la configuration PAM. Par exemple, si la configuration PAM utilise l'authentification d'utilisateur UNIX, la passerelle peut avoir besoin d'un accès en lecture au répertoire `/etc/shadow`.

Pour plus d'informations sur l'interface interactive SQL, voir le *guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Outil de ligne de commande interactif pour les passerelles

Pour les passerelles Gateway for Oracle et Gateway for ODBC, vous pouvez utiliser l'outil de ligne de commande interactif pour les passerelles (**nco_g_icmd**) pour exécuter des commandes sur les passerelles lorsqu'elles sont en cours d'exécution. Pour plus d'informations, voir les guides de référence Gateway for Oracle et Gateway for ODBC.

Utilisation de fichiers de configuration uniques

Les passerelles Gateway for HP ServiceCenter, Gateway for Remedy ARS, et Gateway for Siebel utilisent un fichier de configuration unique pour définir les mappages de données, et contrôler les composants de programme de lecture et de programme d'écriture.

Avant d'exécuter l'une de ces passerelles, vous devez éditer le fichier de configuration pour spécifier le programme de lecture, le programme d'écriture, la route, le mappage et les définitions de filtre.

Le fichier de configuration par défaut est `$OMNIHOME/etc/G_nom_passerelle.conf`. Par exemple, la passerelle Gateway for Siebel utilise le fichier de configuration suivant : `$OMNIHOME/etc/G_SIEBEL.conf`

Référence associée:

«Commandes de filtrage», à la page 212

Des commandes de filtrage sont disponibles pour les passerelles.

«Commandes de mappage», à la page 210

Des commandes de mappage sont disponibles pour les passerelles.

«Commandes de programme de lecture», à la page 206

Des commandes de programme de lecture sont disponibles pour les passerelles.

«Commandes de route», à la page 213

Des commandes de route sont disponibles pour les passerelles.

«Commandes de programme d'écriture», à la page 208

Des commandes de programme d'écriture sont disponibles pour les passerelles.

Configuration du programme de lecture

Un programme de lecture extrait les alertes d'un serveur ObjectServer. Les programmes de lecture sont démarrés à l'aide de la commande `START READER`, qui définit le nom du programme de lecture ainsi que celui du serveur ObjectServer à partir duquel la lecture doit être effectuée.

Par exemple, pour démarrer un programme de lecture à partir d'un serveur ObjectServer NCOMS, ajoutez la commande suivante au fichier de configuration :

```
START READER NCOMS_READ CONNECT TO NCOMS;
```

Une fois cette commande émise, le programme de lecture démarre et la passerelle tente d'ouvrir une connexion avec le serveur ObjectServer source. Si la passerelle parvient à ouvrir une connexion, elle commence immédiatement à lire les alertes du serveur ObjectServer. Pour permettre au programme de lecture de transmettre ces alertes à leur destination, vous devez définir une route et un programme d'écriture associés.

Référence associée:

«Commandes de programme de lecture», à la page 206

Des commandes de programme de lecture sont disponibles pour les passerelles.

Configuration du programme d'écriture

Les programmes d'écriture envoient les alertes acquises par un programme de lecture à l'application ou au serveur ObjectServer de destination. Ils sont créés à l'aide de la commande `START WRITER`, qui définit le nom du programme et les informations qui lui permettent de se connecter à sa destination.

Par exemple, pour créer le programme d'écriture pour une passerelle Fichier à plat, ajoutez la commande suivante au fichier de configuration :

```
START WRITER FILE_WRITER
(
  TYPE = FILE,
  REVISION = 1,
  FILE = '/tmp/omnibus/log/NCOMS_alert.log',
  MAP = FILE_MAP,
  INSERT_HEADER = 'INSERT: ',
  UPDATE_HEADER = 'UPDATE: ',
  DELETE_HEADER = 'DELETE: ',
  START_STRING = '',
  END_STRING = '',
  INSERT_TRAILER = '\n',
  UPDATE_TRAILER = '\n',
  DELETE_TRAILER = '\n'
);
```

Après l'émission de la commande `START WRITER`, la passerelle tente d'établir une connexion à la destination de l'alerte (une application ou un autre ObjectServer). Le programme d'écriture envoie les alertes reçues du serveur ObjectServer source jusqu'à l'émission de la commande `STOP WRITER`.

Référence associée:

«Commandes de programme d'écriture», à la page 208

Des commandes de programme d'écriture sont disponibles pour les passerelles.

Configuration de route

Les routes créent la liaison entre les programmes de lecture et d'écriture. Elles sont créées à l'aide de la commande ADD ROUTE. Cette commande définit le nom de la route, le programme de lecture source et le programme d'écriture de la destination.

Par exemple, pour créer la route entre le programme de lecture du serveur ObjectServer NCOMS et le programme d'écriture d'une passerelle Fichier à plat, ajoutez la commande suivante au fichier de configuration :

```
ADD ROUTE FROM NCOMS_READ TO FILE_WRITER;
```

Une fois cette commande émise, la connexion entre un programme de lecture et un programme d'écriture est établie. Toutes les alertes reçues par le serveur ObjectServer source sont lues par le programme de lecture puis transmises au programme d'écriture via la route et écrites sur le serveur ObjectServer ou l'application de destination.

Référence associée:

«Commandes de route», à la page 213

Des commandes de route sont disponibles pour les passerelles.

Configuration de mappage

Les mappages définissent comment les alertes reçues du serveur ObjectServer source doivent être écrites dans le serveur ObjectServer ou l'application de destination. Chaque programme d'écriture a un mappage différent, défini à l'aide de la commande CREATE MAPPING.

Par exemple, pour créer le mappage entre le programme de lecture du serveur ObjectServer et le programme d'écriture d'une passerelle Fichier à plat, ajoutez la commande suivante au fichier de configuration :

```
CREATE MAPPING FILE_MAP
(
  '@Identif'ier',
  '@Serial',
  '@Node',
  '@Manager',
  '@FirstOccurrence' CONVERT TO DATE,
  '@LastOccurrence' CONVERT TO DATE,
  '@InternalLast' CONVERT TO DATE,
  '@Tally',
  '@Class',
  '@Grade',
  '@Location',
  '@ServerName',
  '@ServerSerial'
);
```

Dans cet exemple, le nom du mappage est FILE_MAP.

Chaque ligne comprise entre les parenthèses définit comment la passerelle écrit les alertes dans le fichier. Pour la passerelle Fichier à plat, la commande CREATE MAPPING définit les zones à partir desquelles les données sont écrites dans chaque alerte du fichier de sortie. Les zones d'alerte du serveur ObjectServer source sont représentées par le symbole @.

L'exemple suivant illustre les commandes INSERT et UPDATE à l'aide du mappage FILE_MAP présenté dans l'exemple précédent :

```

INSERT: "Downlink6LinkMon4Link",127,"sfo4397","Netcool Probe",12/05/03 15:39:23,
12/05/03 15:39:23,12/05/03 15:30:53,1,3300,0,"","NCOMS",127
UPDATE: "muppetMachineMon2Systems",104,"sfo4397","Netcool Probe",12/05/03 12:29:34,
12/05/03 15:40:06,12/05/03 15:31:36,11,3300,0,"","NCOMS",104
UPDATE: "muppetMachineMon4Systems",93,"sfo4397","Netcool Probe",12/05/03 12:29:11,
12/05/03 15:40:35,12/05/03 15:32:05,12,3300,0,"","NCOMS",93

```

D'autres passerelles peuvent nécessiter la spécification d'une zone dans la cible pour chaque zone du serveur ObjectServer source. Par exemple, dans la passerelle Gateway for Remedy ARS, les zones du serveur ObjectServer source sont mappées sur des zones Remedy ARS, identifiées par des valeurs d'entier longues au lieu de noms de zones. Dans l'exemple suivant, la zone ARS 536870913 mappe sur la zone Serial du serveur ObjectServer :

```
536870913 = '@Serial' ON INSERT ONLY
```

La clause ON INSERT ONLY contrôle à quel moment la zone est mise à jour. Les zones ayant la clause ON INSERT ONLY ne sont transmises qu'une seule fois lorsque l'alerte est créée pour la première fois dans le serveur ObjectServer. Les zones qui n'ont pas la clause ON INSERT ONLY sont mises à jour à chaque modification de l'alerte.

Référence associée:

«Commandes de mappage», à la page 210

Des commandes de mappage sont disponibles pour les passerelles.

Configuration de filtre

Vous ne souhaitez peut-être pas toujours envoyer l'ensemble des alertes lues par un programme de lecture à l'application de destination. Les filtres définissent les alertes lues par le programme de lecture du serveur ObjectServer qui doivent être transmises à la destination.

Vous pouvez par exemple décider d'envoyer uniquement les alertes ayant le niveau de gravité Critique.

Vous créez des filtres à l'aide de la commande CREATE FILTER et vous les appliquez en utilisant la commande START READER. Par exemple, pour créer un filtre qui transmet uniquement les alertes critiques à l'application de destination ou au serveur ObjectServer, ajoutez la commande suivante au fichier de configuration :

```
CREATE FILTER CRITONLY AS 'Severity = 5';
```

Cette commande crée un filtre appelé CRITONLY, qui transmet uniquement les alertes ayant le niveau de gravité Critique (5).

Pour appliquer le filtre à un programme de lecture d'un serveur ObjectServer, ajoutez la commande suivante au fichier de configuration :

```
START READER NCOMS_READ CONNECT TO NCOMS USING FILTER CRITONLY;
```

Remarque : Pour effectuer des comparaisons de chaînes avec des filtres, vous devez utiliser des barres obliques inversées comme caractère d'échappement pour les guillemets de la commande CREATE FILTER. Par exemple, pour créer un filtre qui ne transmet que les alertes d'un noeud appelé fred, la commande CREATE FILTER est :

```
CREATE FILTER FREDONLY AS 'NODE = \'fred\'';
```

Création de plusieurs filtres et de plusieurs programmes de lecture

Si vous avez besoin de plusieurs filtres pour le même ObjectServer, vous pouvez créer plusieurs programmes de lecture pour ces filtres.

Par exemple, pour créer un programme de lecture qui transmet toutes les alertes critiques et un autre qui transmet toutes les autres alertes, utilisez les commandes suivantes :

```
CREATE FILTER CRITONLY AS 'Severity = 5';
CREATE FILTER NONCRIT AS 'Severity < 5';
START READER CRIT_NCOMS CONNECT TO NCOMS USING FILTER CRITONLY;
START READER NONCRIT_NCOMS CONNECT TO NCOMS USING FILTER NONCRIT;
```

Chargement de filtres créés à l'aide du Générateur de filtres

Vous pouvez charger et utiliser des filtres créés à l'aide du Générateur de filtre.

Par exemple :

```
LOAD FILTER FROM '/usr/filters/myfilt.elf';
```

Cette commande charge le fichier `/usr/filters/myfilt.elf` en tant que filtre. Ce nom de filtre est défini par la zone **Nom** du Générateur de filtre.

Remarque : La zone **Nom** doit être alphabétique et ne contenir aucun espace.

Référence associée:

«Commandes de filtrage», à la page 212

Des commandes de filtrage sont disponibles pour les passerelles.

«START READER», à la page 206

La commande `START READER` vous permet de démarrer un programme de lecture appelé *nom_programme_lecture* qui se connecte à un serveur ObjectServer appelé *nom_serveur*.

Options de ligne de commande de passerelle communes

Les passerelles qui utilisent un fichier de configuration unique disposent de plusieurs options de ligne de commande communes. Ces passerelles n'utilisent pas de fichiers de propriétés.

Les options de ligne de commande communes décrites ici sont utilisées par les passerelles suivantes :

- Gateway for HP ServiceCenter
- Gateway for Remedy ARS
- Gateway for Siebel

Le tableau suivant répertorie les options de ligne de commande communes disponibles que vous pouvez utiliser au démarrage d'une passerelle. Reportez-vous à la documentation de référence de chaque passerelle pour plus d'informations sur ses options de ligne de commande spécifiques.

Tableau 43. Options de ligne de commande de passerelle communes

Option de ligne de commande	Description
-admingroup <i>chaîne</i>	Spécifie le nom du groupe d'utilisateurs UNIX disposant des droits d'administrateur. Les membres de ce groupe peuvent se connecter à la passerelle. Le nom de groupe par défaut est ncoadmin.
-authenticate UNIX PAM HPTCB	<p>Spécifie le mode d'authentification à utiliser pour vérifier les données d'identification des utilisateurs. Les options sont UNIX, PAM et HPTCB.</p> <p>Le mode d'authentification par défaut est UNIX, ce qui signifie que la fonction Posix getpwnam ou getsnam est utilisée pour vérifier les données d'authentification des utilisateurs sur les systèmes d'exploitation UNIX. En fonction de la configuration du système, les mots de passe sont vérifiés à l'aide du fichier /etc/passwd, du fichier de mots de passe shadow /etc/shadow, de NIS ou de NIS+.</p> <p>Si PAM est spécifié comme mode d'authentification, des modules PAM (Pluggable Authentication Modules) sont utilisés pour vérifier les données d'identification. Le nom de service utilisé par la passerelle lors de l'initialisation de l'interface PAM est netcool.</p> <p>L'authentification PAM est uniquement disponible sur les systèmes d'exploitation Linux, Solaris et HP-UX 11.</p> <p>Si HPTCB est spécifié comme mode d'authentification, ce système de protection des mots de passe HP-UX est utilisé. Cette option est uniquement disponible sur les systèmes HP de confiance (sécurisés).</p>
-config <i>chaîne</i>	Spécifie le nom du fichier de configuration à lire au démarrage de la passerelle. La valeur par défaut est \$OMNIHOME/etc/ <i>nom_passerelle.conf</i> .
-connections	Nombre de connexions autorisées. La valeur par défaut est 30.
-debug	Lorsque cette option est spécifiée, le mode débogage est activé.
-help	Affiche des informations d'aide sur les options de ligne de commande et quitte.
-ipctimeout	Délai d'attente de session IPC. La valeur par défaut est 60 secondes.
-logfile <i>chaîne</i>	Spécifie le nom du fichier journal. Si cette option est omise, la valeur par défaut est \$NCHOME/omnibus/log/ <i>nom_passerelle.log</i> .
-logsize <i>entier</i>	Spécifie la taille maximale du fichier journal en Ko. La valeur minimale est 16 Ko. La valeur par défaut est 1 Mo.
-messagelevel	<p>Niveau de messages à consigner. Les niveaux disponibles sont : debug, info, warn, error et fatal.</p> <p>La valeur par défaut est warn.</p>
-messagelog	<p>Chemin du fichier journal des messages.</p> <p>La valeur par défaut est : \$NCHOME/omnibus/log/NCO_GATE.log</p>
-name <i>chaîne</i>	<p>Spécifie le nom de la passerelle. Indiquez ce nom à la suite de l'option de ligne de commande -server pour vous connecter à la passerelle à l'aide de nco_sql.</p> <p>Si cette option est omise, la valeur par défaut est <i>NOM_PASSERELLE</i>.</p>
-notruncate	Spécifie que le fichier journal n'est pas tronqué.
-oldtimestamp	Indique le format d'horodatage à utiliser dans le fichier journal.

Tableau 43. Options de ligne de commande de passerelle communes (suite)

Option de ligne de commande	Description
-propsfile	Chemin d'accès complet au fichier de propriétés de passerelle.
-queue <i>entier</i>	Spécifie la taille des files d'attente internes. La valeur par défaut est 1024. Ne modifiez pas cette option, à moins d'y être invité par le service de support logiciel IBM.
-stacksize <i>entier</i>	Spécifie la taille des unités d'exécution internes. La valeur par défaut est 256 Ko. Ne modifiez pas cette option, à moins d'y être invité par le service de support logiciel IBM.
-unique log	Si -logfile n'est pas défini, cette option force le fichier journal à avoir un nom unique en ajoutant l'ID processus de la passerelle à la fin du nom de fichier journal par défaut. Si -logfile est défini, cette option est sans effet.
-version	Affiche les informations de version et quitte.

Configuration de passerelles en cours d'exécution

Vous pouvez utiliser l'interface interactive SQL pour modifier la configuration d'une passerelle en cours d'exécution.

Pourquoi et quand exécuter cette tâche

Remarque : Pour se connecter à une passerelle sous UNIX à l'aide de `nco_sql`, vous devez indiquer les nom d'utilisateur et mot de passe d'un membre du groupe d'utilisateurs UNIX autorisé à se connecter à une passerelle. Ce groupe d'utilisateurs est indiqué à l'aide de l'option de ligne de commande `-admin group`. Par défaut, il s'agit du groupe d'utilisateurs `ncoadmin`. Il se peut que vous deviez demander à votre administrateur système de créer ce groupe. De plus, l'utilisateur qui exécute la passerelle doit disposer d'un accès au fichier approprié permettant de vérifier des mots de passe, de sorte que les membres du groupe `ncoadmin` puissent être authentifiés lors de la connexion à la passerelle, à l'aide de `nco_sql`.

Utilisez l'interface interactive SQL pour vous connecter à une passerelle en tant qu'utilisateur spécifique, comme le montre le tableau suivant.

Tableau 44. Connexion à la passerelle à l'aide de l'interface interactive SQL

Sous	Entrez la commande suivante
UNIX	<code>\$OMNIHOME/bin/nco_sql -server <i>nom_serveur</i> -user <i>nom_utilisateur</i></code>
Windows	<code>%OMNIHOME%\bin\redist\isql -S <i>nom_serveur</i> -U <i>nom_utilisateur</i></code>

Dans ces commandes, `nom_serveur` correspond au nom de la passerelle et `nom_utilisateur` représente un nom d'utilisateur valide. Si vous n'indiquez pas de nom d'utilisateur, la valeur par défaut correspond à l'utilisateur qui exécute la commande.

Vous êtes invité à entrer un mot de passe. Sous UNIX, la valeur par défaut consiste à entrer votre mot de passe UNIX. Pour authentifier les utilisateurs qui emploient d'autres méthodes, utilisez l'option de ligne de commande `-authenticate`.

Après la connexion à l'aide d'un nom d'utilisateur et d'un mot de passe, une invite numérotée s'affiche.

1>

Vous pouvez entrer des commandes pour configurer la passerelle dynamiquement. L'exemple suivant montre une session dans laquelle de nouvelles routes sont ajoutées à une passerelle avec un fichier de configuration .conf unique :

```
$ nco_sql -server REMEDY
Mot de passe :
Utilisateur 'admin' connecté.
1> ADD ROUTE FROM DENCO_READ TO ARS_WRITER ;
2> ADD ROUTE FROM DENCO_READ TO OS_WRITER ;
3> go
```

1>

Remarque : Si vous souhaitez désactiver une configuration interactive, ajoutez la ligne suivante à la fin du fichier de configuration de la passerelle :

```
SET CONNECTIONS FALSE ;
```

Chargement et enregistrement de configurations

Vous pouvez utiliser l'interface interactive SQL pour charger et enregistrer des configurations de passerelle lorsque la passerelle est en cours d'exécution.

Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser le fichier de configuration enregistré pour d'autres passerelles.

Procédure

1. Pour arrêter les programmes de lecture et d'écriture en cours d'exécution :

```
>1 STOP;
>2 go
```

2. Pour supprimer la configuration en cours :

```
>1 DUMP CONFIG;
>2 go
```

La commande DUMP CONFIG ne fonctionne pas si des programmes de lecture et d'écriture sont en cours d'exécution, ou si la configuration a été modifiée de manière interactive. Pour déterminer si la configuration a été modifiée de manière interactive, utilisez la commande SHOW SYSTEM. Vous pouvez utiliser l'option FORCE pour forcer la suppression de la configuration en cours.

3. Pour charger une nouvelle configuration :

```
>1 LOAD CONFIG FROM 'nom_fichier';
>2 go
```

Où *nom_fichier* est le chemin d'accès à un autre fichier de configuration.

4. Pour enregistrer la configuration en cours :

```
>1 SAVE CONFIG TO 'nom_fichier';
>2 go
```

Où *nom_fichier* est le nom et le chemin d'accès pour le nouveau fichier de configuration.

Commandes de passerelle

Les rubriques suivantes décrivent les commandes que vous pouvez utiliser avec des passerelles faisant appel à un fichier de configuration unique.

Les commandes décrites ici sont utilisées par les passerelles suivantes :

- Gateway for HP ServiceCenter
- Gateway for Remedy ARS
- Gateway for Siebel

Les commandes de passerelle peuvent être utilisées dans le fichier de configuration (*G_nom_passerelle.conf*) ou avec l'interface interactive SQL. Le tableau suivant répertorie les commandes disponibles :

Tableau 45. Commandes de passerelle

Groupe de commandes	Commande	Description
Commandes de programme de lecture	START READER	Démarre un programme de lecture.
	STOP READER	Arrête un programme de lecture.
	SHOW READERS	Répertorie tous les programmes de lecture en cours.
Commandes de programme d'écriture	START WRITER	Démarre un programme d'écriture.
	STOP WRITER	Arrête un programme d'écriture.
	SHOW WRITERS	Répertorie tous les programmes d'écriture en cours.
	SHOW WRITER TYPES	Répertorie tous les types de programme d'écriture pris en charge par une passerelle.
Commandes de mappage	SHOW WRITER ATTRIBUTES	Répertorie tous les attributs d'un programme d'écriture.
	CREATE MAPPING	Crée un fichier de mappage.
	DROP MAPPING	Supprime un fichier de mappage.
	SHOW MAPPINGS	Répertorie tous les mappages utilisés par une passerelle.
Commandes de filtrage	SHOW MAPPING ATTRIBUTES	Répertorie tous les attributs d'un mappage.
	CREATE FILTER	Crée un filtre.
	LOAD FILTER	Charge un filtre.
Commandes de route	DROP FILTER	Supprime un filtre.
	ADD ROUTE	Ajoute une route entre un programme de lecture et un programme d'écriture.
	REMOVE ROUTE	Supprime une route.
Commandes de configuration	SHOW ROUTES	Répertorie tous les routes configurées pour une passerelle.
	LOAD CONFIG	Charge un fichier de configuration.
	SAVE CONFIG	Enregistre un fichier de configuration.
	DUMP CONFIG	Efface la configuration en cours.

Tableau 45. Commandes de passerelle (suite)

Groupe de commandes	Commande	Description
Commandes générales	SHUTDOWN	Arrête la passerelle.
	SET CONNECTIONS	Active ou désactive les connexions à la passerelle à partir de l'interface interactive SQL.
	SHOW SYSTEM	Affiche des informations sur les paramètres de la passerelle.
	SET DEBUG MODE	Définit le mode débogage de la passerelle.
	TRANSFER	Transfère le contenu d'une table de base de données vers une autre table de base de données.

Commandes de programme de lecture

Des commandes de programme de lecture sont disponibles pour les passerelles.

Concepts associés:

«Configuration du programme de lecture», à la page 198

Un programme de lecture extrait les alertes d'un serveur ObjectServer. Les programmes de lecture sont démarrés à l'aide de la commande START READER, qui définit le nom du programme de lecture ainsi que celui du serveur ObjectServer à partir duquel la lecture doit être effectuée.

START READER :

La commande START READER vous permet de démarrer un programme de lecture appelé *nom_programme_lecture* qui se connecte à un serveur ObjectServer appelé *nom_serveur*.

Syntaxe

```
START READER nom_programme_lecture CONNECT TO nom_serveur
[ USING FILTER nom_filtre ]
[ ORDER BY 'column, ... [ ASC | DESC ]' ]
[ AFTER IDUC DO 'commande_mise_à_jour' ]
[ IDUC = entier] [ JOURNAL_FLUSH = entier] [ IDUC_ORDER ];
```

La clause facultative USING FILTER, suivie du nom d'un filtre créé à l'aide de la commande CREATE FILTER, vous permet de restreindre le nombre de lignes affectées par les mises à jour de passerelle. Le filtre remplace une clause SQL WHERE pour que la passerelle mette uniquement à jour les lignes sélectionnées par le filtre.

La clause facultative ORDER BY indique à la passerelle d'afficher les résultats par ordre séquentiel, en fonction des valeurs d'un ou plusieurs nom de colonne, par ordre décroissant (DESC) ou croissant (ASC). Si la clause ORDER BY n'est pas spécifiée, aucun tri n'est utilisé.

La clause facultative AFTER IDUC indique à la passerelle d'effectuer la mise à jour spécifiée dans la commande *commande_mise_à_jour* sur le serveur ObjectServer lorsqu'elle place des alertes dans la file d'attente du programme d'écriture. Cette clause est utilisée pour fournir un retour lorsque les alertes passent par une passerelle.

Remarque : La commande de mise à jour qui suit une instruction AFTER IDUC DO dans la commande START READER doit être une instruction UPDATE simple. Elle ne doit pas utiliser de conditions (par exemple WHERE ou HAVING) ; elles ne sont pas prises en charge dans ce contexte.

La valeur spécifiée dans la clause facultative IDUC indique un intervalle IDUC pour les passerelles qui est plus fréquent que la valeur de la propriété **Granularity** définie sur le serveur ObjectServer source. Les mises à jour de passerelles peuvent ainsi être transmises plus rapidement à la source, sans provoquer de détérioration des performances système générales.

La valeur spécifiée dans la clause facultative JOURNAL_FLUSH indique un délai en secondes entre la mise à jour IDUC dans le serveur ObjectServer (chaque *Granularité* secondes) et le moment où les entrées de journal sont extraites par la passerelle. Normalement, seules les entrées de journal effectuées pendant les dernières *Granularité* secondes sont extraites. En cas de charge importante du système, définissez cette clause pour que les entrées de journal soient extraites pour les dernières *entier + Granularité* secondes. Cela empêche la perte des entrées de journal créés après la mise à jour IDUC mais avant l'extraction des entrées par la passerelle. Toutes les entrées de journal doubles extraites sont éliminées par le dédoublement.

La clause facultative IDUC_ORDER indique l'ordre dans lequel les données IDUC sont traitées. Le mode de traitement par défaut pour les passerelles est de traiter les instructions DELETE suivies des instructions UPDATE puis des instructions INSERT. Ne modifiez pas cette clause, sauf si le service de support logiciel IBM vous l'a conseillé.

Exemple

L'exemple suivant utilise la zone Grade comme zone d'état. Initialement, toutes les sondes définissent Grade sur 0. La passerelle filtre toutes les alertes qui ont un Grade de 1. Une fois que les alertes sont passées par la passerelle, la mise à jour AFTER IDUC fournit un retour sur l'état de l'alerte en modifiant la valeur de la zone Grade sur 2.

```
START READER NCOMS_READER CONNECT TO NCOMS USING FILTER CRIT_ONLY  
ORDER BY 'SERIAL ASC' AFTER IDUC DO 'update alerts.status set Grade=2';
```

Concepts associés:

«Configuration de filtre», à la page 200

Vous ne souhaitez peut-être pas toujours envoyer l'ensemble des alertes lues par un programme de lecture à l'application de destination. Les filtres définissent les alertes lues par le programme de lecture du serveur ObjectServer qui doivent être transmises à la destination.

STOP READER :

La commande STOP READER vous permet d'arrêter le programme de lecture *nom_programme_lecture*.

Syntaxe

```
STOP READER nom_programme_lecture;
```

Cette commande n'arrête pas le programme de lecture s'il est utilisé avec des routes.

Exemple

```
STOP READER NCOMS_READ;
```

SHOW READERS :

La commande SHOW READERS vous permet de répertorier tous les programmes de lecture actuels qui ont été démarrés et qui sont en cours d'exécution sur la passerelle.

Syntaxe

```
SHOW READERS;
```

Cette commande peut uniquement être utilisée de manière interactive.

Exemple

```
SHOW READERS;
```

Commandes de programme d'écriture

Des commandes de programme d'écriture sont disponibles pour les passerelles.

Concepts associés:

«Configuration du programme d'écriture», à la page 198

Les programmes d'écriture envoient les alertes acquises par un programme de lecture à l'application ou au serveur ObjectServer de destination. Ils sont créés à l'aide de la commande START WRITER, qui définit le nom du programme et les informations qui lui permettent de se connecter à sa destination.

START WRITER :

La commande START WRITER vous permet de démarrer un programme d'écriture appelé *nom_programme_écriture*.

Syntaxe

```
START WRITER nom_programme_écriture  
( TYPE=type_programme_écriture , REVISION=numéro  
[ , paramètre_mot_clé [ , paramètre_mot_clé ] ... ] );
```

La commande START WRITER est suivie d'une liste de paramètres de mot clé séparés par des virgules entre parenthèses. Le premier paramètre doit être un paramètre TYPE qui indique le *type_programme_écriture*. Le paramètre suivant doit être un paramètre REVISION. Il est actuellement défini sur 1 pour tous les programmes d'écriture. Les mots clés restants ainsi que leurs paramètres dépendent du type de programme d'écriture.

Exemple

Cet exemple démarre le programme d'écriture pour une passerelle Socket Writer.

```
START WRITER SOCKET_WRITER  
(  
    TYPE = SOCKET,  
    REVISION = 1,  
    HOST = 'sfo768',  
    PORT = 4010,  
    MAP = SOCKET_MAP,  
    INSERT_HEADER = 'INSERT: ',  
    UPDATE_HEADER = 'UPDATE: ',  
    DELETE_HEADER = 'DELETE: ',  
    START_STRING = ''',
```

```

        END_STRING = '',
        INSERT_TRAILER = '\n',
        UPDATE_TRAILER = '\n',
        DELETE_TRAILER = '\n'
);

```

STOP WRITER :

La commande STOP WRITER vous permet d'arrêter le programme d'écriture appelé *nom_programme_écriture*.

Syntaxe

```
STOP WRITER nom_programme_écriture;
```

Si une route utilise ce programme d'écriture, il n'est pas arrêté.

Exemple

```
STOP WRITER ARS_WRITER;
```

SHOW WRITERS :

La commande SHOW WRITERS vous permet de répertorier tous les programmes d'écriture actuels dans la passerelle.

Syntaxe

```
SHOW WRITERS;
```

Cette commande peut uniquement être utilisée de manière interactive.

Exemple

```

1>SHOW WRITERS;
2>GO
Name          Type Routes Msgq Id Mutex Id Thread
-----
SNMP_WRITER  SNMP  1      15    0      0  0x001b8cd0
1>

```

SHOW WRITER TYPES :

La commande SHOW WRITER TYPES vous permet de répertorier tous les types de programme d'écriture actuellement connus et pris en charge par la passerelle.

Syntaxe

```
SHOW WRITER TYPES;
```

Cette commande peut uniquement être utilisée de manière interactive.

Exemple

```

1> SHOW WRITER TYPES;
2> GO
Type          Revision      Description
-----
ARS           1             Action Request System V3.0
OBJECT_SERVER 1             Netcool/OMNIBUS ObjectServer V7
SYBASE        1             Sybase SQL Server 10.0 RDBMS
SNMP          1             SNMP Trap forwarder
SERVICE_VIEW 1             Service View

```

SHOW WRITER ATTRIBUTES :

La commande SHOW WRITER ATTRIBUTES vous permet d'afficher toutes les paramètres (ou attributs) du programme d'écriture *nom_programme_écriture*.

Syntaxe

```
SHOW WRITER { ATTRIBUTES | ATTR } FOR nom_programme_écriture;
```

Le mot clé ATTRIBUTES peut être échangé avec le mot clé abrégé ATTR.

Cette commande peut uniquement être utilisée de manière interactive.

Exemple

```
1> SHOW WRITER ATTR FOR SNMP_WRITER;
```

```
2> GO
```

Attribute	Value
MAP	SNMP_MAP
TYPE	SNMP
REVISION	1
GATEWAY	penelope

```
1>
```

Commandes de mappage

Des commandes de mappage sont disponibles pour les passerelles.

Concepts associés:

«Configuration de mappage», à la page 199

Les mappages définissent comment les alertes reçues du serveur ObjectServer source doivent être écrites dans le serveur ObjectServer ou l'application de destination. Chaque programme d'écriture a un mappage différent, défini à l'aide de la commande CREATE MAPPING.

CREATE MAPPING :

La commande CREATE MAPPING vous permet de créer un fichier de mappage appelé *nom_mappage* à utiliser par un programme d'écriture.

Syntaxe

```
CREATE MAPPING nom_mappage ( mapping [ , mapping ] );
```

Les lignes de mappage ont la syntaxe suivante :

```
{ chaîne | entier } = { chaîne | entier | nom | réel | booléen }  
[ ON INSERT ONLY ] [ CONVERT TO { INT | STRING | DATE } ]
```

Le premier argument est un identificateur pour la zone de destination et le deuxième est un identificateur pour la zone source (ou une valeur prédéfinie).

Le côté droit du mappage dépend du programme d'écriture avec lequel le mappage doit être utilisé. (Pour obtenir des détails spécifiques à la passerelle, voir la section programme d'écriture de la documentation sur la passerelle individuelle.)

La clause facultative ON INSERT ONLY détermine le comportement de mise à jour du mappage. Sans clause ON INSERT ONLY, la zone est mise à jour chaque fois qu'une modification est apportée à une alerte. Avec la clause ON INSERT ONLY, la zone est insérée au moment de la création (c'est-à-dire lorsque l'alerte apparaît

pour la première fois) mais n'est pas mise à jour lors des mises à jour ultérieures de l'alerte, même si la valeur de la zone est modifiée.

La clause facultative de type CONVERT TO permet au mappage de définir une conversion forcée pour les situations dans lesquelles une zone source ne correspond pas au type de la zone de destination. Le type peut être INT, STRING ou DATE. La conversion de la zone source dans le type de données spécifié est ainsi forcée.

Exemple

```
CREATE MAPPING SYBASE_MAP
(
  'Node'=@Node ON INSERT ONLY,
  'Summary'=@Summary ON INSERT ONLY,
  'Severity'=@Severity );
```

DROP MAPPING :

La commande DROP MAPPING vous permet de supprimer un mappage appelé *nom_mappage* de la passerelle.

Syntaxe

```
DROP MAPPING nom_mappage;
```

Cette commande ne supprime pas la mappe si elle est utilisée par un programme d'écriture.

Exemple

```
DROP MAPPING SYBASE_MAP;
```

SHOW MAPPINGS :

La commande SHOW MAPPINGS vous permet de répertorier tous les mappages actuellement créés dans la passerelle.

Syntaxe

```
SHOW MAPPINGS;
```

Cette commande peut uniquement être utilisée de manière interactive.

Exemple

```
1> SHOW MAPPINGS;
2> GO
  Name                               Writers
-----
SNMP_MAP                             1
1>
```

SHOW MAPPING ATTRIBUTES :

La commande SHOW MAPPING ATTRIBUTES vous permet d'afficher les mappages (ou les attributs) du mappage *nom_mappage*.

Syntaxe

```
SHOW MAPPING { ATTRIBUTES | ATTR } FOR nom_mappage;
```

Le mot clé ATTRIBUTES peut être échangé avec le mot clé abrégé ATTR. Cette commande peut uniquement être utilisée de manière interactive.

Exemple

```
SHOW MAPPING ATTR FOR SYBASE_MAP;
```

Commandes de filtrage

Des commandes de filtrage sont disponibles pour les passerelles.

Concepts associés:

«Configuration de filtre», à la page 200

Vous ne souhaitez peut-être pas toujours envoyer l'ensemble des alertes lues par un programme de lecture à l'application de destination. Les filtres définissent les alertes lues par le programme de lecture du serveur ObjectServer qui doivent être transmises à la destination.

CREATE FILTER :

La commande CREATE FILTER vous permet de créer un filtre appelé *nom_filtre* à utiliser par un programme de lecture.

Syntaxe

```
CREATE FILTER nom_filtre AS condition_filtre;
```

La spécification de filtre *condition_filtre* est une condition SQL.

Exemple

```
CREATE FILTER HIGH_TALLY_LOG AS 'Tally > 100';  
CREATE FILTER NCOMS_FILTER AS 'Agent = \'NNM\'';
```

LOAD FILTER :

La commande LOAD FILTER vous permet de charger un filtre à partir d'un fichier.

Syntaxe

```
LOAD FILTER FROM 'nom_fichier';
```

Indiquez le chemin d'accès au fichier. Les fichiers ont une extension *.elf*.

Exemple

```
LOAD FILTER FROM '/disk/filters/newfilter.elf';
```

DROP FILTER :

La commande DROP FILTER vous permet de supprimer un filtre appelé *nom_filtre* de la passerelle.

Syntaxe

```
DROP FILTER nom_filtre;
```

Le filtre n'est pas supprimé s'il est utilisé par un programme de lecture.

Exemple

```
DROP FILTER HIGH_TALLY_LOG;
```

Commandes de route

Des commandes de route sont disponibles pour les passerelles.

Concepts associés:

«Configuration de route», à la page 199

Les routes créent la liaison entre les programmes de lecture et d'écriture. Elles sont créées à l'aide de la commande ADD ROUTE. Cette commande définit le nom de la route, le programme de lecture source et le programme d'écriture de la destination.

ADD ROUTE :

La commande ADD ROUTE vous permet d'ajouter une route entre un programme de lecture appelé *nom_programme_lecture* et un programme d'écriture appelé *nom_programme_écriture*, pour permettre aux alertes de passer par la passerelle.

Syntaxe

```
ADD ROUTE FROM nom_programme_lecture TO nom_programme_écriture;
```

Exemple

```
ADD ROUTE FROM NCOMS_READER TO ARS_WRITER;
```

REMOVE ROUTE :

La commande REMOVE ROUTE vous permet de supprimer une route existante entre un programme de lecture appelé *nom_programme_lecture* et un programme d'écriture appelé *nom_programme_écriture*.

Syntaxe

```
REMOVE ROUTE FROM nom_programme_lecture TO nom_programme_écriture;
```

Exemple

```
REMOVE ROUTE FROM NCOMS_READER TO ARS_WRITER;
```

SHOW ROUTES :

La commande SHOW ROUTES vous permet d'afficher toutes les routes actuellement configurées dans la passerelle.

Syntaxe

```
SHOW ROUTES;
```

Cette commande peut uniquement être utilisée de manière interactive.

Exemple

```
1> SHOW ROUTES;
```

```
2> GO
```

```
Reader                               Writer
-----
NCOMS_READER                         SNMP_WRITER
```

```
1>
```

Commandes de configuration

Des commandes de configuration sont disponibles pour les passerelles.

LOAD CONFIG :

La commande LOAD CONFIG vous permet de charger un fichier de configuration de passerelle appelé *nom_fichier*.

Syntaxe

```
LOAD CONFIG FROM 'nom_fichier';
```

Exemple

```
LOAD CONFIG FROM '/disk/config/gateconf.conf';
```

SAVE CONFIG :

La commande SAVE CONFIG vous permet de sauvegarder la configuration actuelle de la passerelle dans un fichier appelé *nom_fichier*.

Syntaxe

```
SAVE CONFIG TO 'nom_fichier';
```

Exemple

```
SAVE CONFIG TO '/disk/config/newgate.conf';
```

DUMP CONFIG :

La commande DUMP CONFIG vous permet d'effacer la configuration actuelle.

Syntaxe

```
DUMP CONFIG [ FORCE ];
```

Si la passerelle est active et transmet des alertes, cette commande n'efface pas la configuration, à moins d'utiliser le mot clé facultatif FORCE.

Exemple

```
DUMP CONFIG;
```

Commandes générales

Des commandes générales sont disponibles pour les passerelles.

SHUTDOWN :

La commande SHUTDOWN vous permet d'ordonner à la passerelle de s'arrêter ; tous les programmes de lecture et d'écriture sont arrêtés.

Syntaxe

```
SHUTDOWN [ FORCE ];
```

Par défaut, la passerelle n'est pas arrêtée si des modifications interactives de la configuration n'ont pas été sauvegardées.

Si le mot clé facultatif FORCE est utilisé, la passerelle est arrêtée, même si la configuration a été modifiée de manière interactive.

Exemple

```
SHUTDOWN;
```

SET CONNECTIONS :

La commande SET CONNECTIONS vous permet d'activer ou de désactiver des connexions à la passerelle à l'aide de l'interface SQL interactive.

Syntaxe

```
SET CONNECTIONS { TRUE | FALSE | YES | NO };
```

Lorsque la commande est définie sur FALSE ou NO, la connexion à la passerelle est impossible avec `nco_sql`. Lorsque la commande est définie sur TRUE ou YES, la connexion à la passerelle avec `nco_sql` est possible. Cette commande détermine si une reconfiguration interactive est autorisée.

Exemple

```
SET CONNECTIONS TRUE;
```

SHOW SYSTEM :

La commande SHOW SYSTEM vous permet d'afficher des informations sur les paramètres de passerelle actuels.

Syntaxe

```
SHOW SYSTEM;
```

Les paramètres renvoyés sont indiqués dans le tableau suivant.

Tableau 46. Paramètres de la commande Show system

Paramètre système	Description
Version	Numéro de version de la passerelle.
Server Type	Type de serveur. Défini sur Gateway.
Connections	Statut de l'indicateur SET CONNECTIONS.

Tableau 46. Paramètres de la commande Show system (suite)

Paramètre système	Description
Debug Mode	Statut de l'indicateur SET DEBUG MODE.
Multi User	Mode multi-utilisateurs de la passerelle. Défini sur YES.
Configuration Changed	Si la configuration a été modifiée de manière interactive, ce paramètre est défini sur YES.

Des paramètres supplémentaires peuvent être renvoyés en mode débogage. Cette commande peut uniquement être utilisée de manière interactive.

Exemple

```
1> SHOW SYSTEM;
2> GO
System Parameter      Value
-----
Version                7.0
Server Type            Gateway
Connections            ENABLED
Debug Mode             NO
Multi User             YES
```

SET DEBUG MODE :

La commande SET DEBUG MODE vous permet de définir le mode de débogage de la passerelle.

Syntaxe

```
SET DEBUG MODE { TRUE | FALSE | YES | NO };
```

Lorsqu'elle est définie sur TRUE ou YES, les messages de débogage sont envoyés au fichier journal. La définition par défaut est NO ou FALSE. N'utilisez cette commande que si le service de support logiciel IBM vous l'a conseillé.

Exemple

```
SET DEBUG MODE NO;
```

TRANSFER :

La commande TRANSFER vous permet de transférer le contenu d'une table de base de données vers une autre table de base de données.

Syntaxe

```
TRANSFER 'nom_table' FROM nom_programme_lecture TO nom_programme_écriture
[ AS 'format_table' ]
{ DELETE | DELETE condition | DO NOT DELETE }
[ USE TRANSFER_MAP ] [ USING FILTER clause_filtre ];
```

Vous pouvez utiliser cette commande pour transférer des tables entre des passerelles Sybase, Oracle, Informix, ODBC, CORBA et Socket Writer.

La clause AS *format_table* spécifie le format de la table de destination s'il est différent du format de table source.

Les clauses DELETE et DO NOT DELETE définissent le mode de traitement de la table de destination. Par défaut, le contenu de la table de destination est supprimé

avant le transfert du contenu de la table source. Vous pouvez facultativement spécifier une condition qui détermine si la suppression est effectuée. Si vous spécifiez la clause DO NOT DELETE, le contenu de la table de destination n'est pas supprimé avant le transfert du contenu de la table source.

Remarque : La clause DELETE ne fonctionne pas avec les passerelles Socket Writer et CORBA.

La clause USE TRANSFER_MAP indique à la passerelle d'utiliser la définition de mappage affectée comme mappe au programme d'écriture utilisé dans la commande TRANSFER. Cette clause est uniquement disponible pour la passerelle Oracle.

Une clause de filtrage facultative peut être appliquée en spécifiant USING FILTER suivi du filtre. Entrez un filtre valide.

Exemple

```
TRANSFER 'alerts.conversions' FROM NCO_READER TO SYBASE_WRITER AS
'alerts.conversions' DELETE;
TRANSFER 'alerts.status' FROM NCOMS_READ TO DENCO_WRITE AS 'ncoms.status'
USING FILTER 'ServerName = \'NCOMS\'T' DELETE USE TRANSFER_MAP;
```

Création de tables de conversion

Vous pouvez créer des tables de conversion pour permettre la conversion de certaines données entre les zones.

Par exemple, si vous utilisez la passerelle pour Remedy ARS, vous pouvez créer une table dans le serveur ObjectServer pour insérer des valeurs pour la zone Severity de Remedy ARS.

Pour ce faire, vous devez utiliser des commandes SQL ObjectServer. Vous pouvez exécuter ces commandes à l'aide des outils suivants :

- Interface SQL interactive UNIX (**nco_sql**)
- Interface SQL interactive Windows (**isql**)
- Administrateur Netcool/OMNIBus

L'exemple de commandes SQL ObjectServer suivant crée une table remedy dans un serveur ObjectServer et insère six valeurs ainsi que les descriptions correspondantes pour la zone Severity :

```
create database conversions;
use database conversions;
create table conversions.remedy persistent (
  KeyField varchar(255) primary key,
  ColName varchar(255),
  OSValue varchar(255),
  Conversion varchar(255)
);
go

insert into conversions.remedy values ('Severity0','Severity','0','Clear');
insert into conversions.remedy values ('Severity1','Severity','1','Indeterminate');
insert into conversions.remedy values ('Severity2','Severity','2','Warning');
insert into conversions.remedy values ('Severity3','Severity','3','Minor');
insert into conversions.remedy values ('Severity4','Severity','4','Major');
insert into conversions.remedy values ('Severity5','Severity','5','Critical');
```

Chapitre 10. Exécution des passerelles

Pour pouvoir exécuter une passerelle, vous devez créer une entrée pour celle-ci dans l'éditeur de serveur. Vous devez également configurer la passerelle en fonction de votre environnement opérationnel.

Pour plus d'informations sur l'ajout d'une entrée dans l'éditeur de serveur, voir *IBM Tivoli Netcool/OMNIbus - Guide d'installation et de déploiement*.

Concepts associés:

Chapitre 9, «Configuration de passerelles», à la page 183

Les fichiers de configuration définissent l'environnement dans lequel les passerelles fonctionnent et la manière dont elles mappe les données entre les tables ObjectServer et les bases de données ou applications cible. La plupart des passerelles utilisent plusieurs fichiers de configuration. Certaines passerelles sont configurées à l'aide d'un seul fichier de configuration.

Utilisation des variables d'environnement OMNIHOME et NCHOME

Tivoli Netcool/OMNIbus version 7.0 (et les versions antérieures) utilise la variable d'environnement OMNIHOME dans de nombreux fichiers de configuration. Pour utiliser ces fichiers sur Tivoli Netcool/OMNIbus V7.1 (et ultérieures), remplacez les occurrences de la variable d'environnement OMNIHOME par NCHOME/omnibus.

Sur les systèmes d'exploitation UNIX et Linux, remplacez \$OMNIHOME par \$NCHOME/omnibus.

Sur les systèmes d'exploitation Windows, remplacez %OMNIHOME% par %NCHOME%\omnibus.

Exécution des passerelles

Sur les systèmes d'exploitation UNIX et Linux, vous pouvez démarrer des passerelles à partir de la ligne de commande. Sur les systèmes d'exploitation Windows, vous pouvez démarrer des passerelles à partir de la ligne de commande ou en tant que services Windows.

Pour démarrer une passerelle qui utilise les fichiers de configuration par défaut, exécutez la commande suivante :

- Sous UNIX et Linux : `$OMNIHOME/bin/nco_g_nom_passerelle`
- Sous Windows : `%OMNIHOME%\bin\nco_g_nom_passerelle`

Pour démarrer une deuxième instance d'une passerelle sous un autre nom, utilisez l'une des commandes suivantes :

- Sous UNIX et Linux :
 - Utilisez la commande suivante pour démarrer une passerelle qui fait appel à plusieurs fichiers de configuration :
`$OMNIHOME/bin/nco_g_nom_passerelle -name GATE2 -propsfile $OMNIHOME/etc/GATE2.props`
 - Utilisez la commande suivante pour démarrer une passerelle qui fait appel à un fichier de configuration unique :

```
$OMNIHOME/bin/nco_g_nom_passerelle -name GATE2 -config  
$OMNIHOME/etc/GATE2.conf
```

- Sous Windows :
 - Utilisez la commande suivante pour démarrer une passerelle qui fait appel à plusieurs fichiers de configuration :

```
%OMNIHOME%\bin\nco_g_nom_passerelle.exe -name GATE2 -propsfile  
%OMNIHOME%\etc\GATE2.props
```
 - Utilisez la commande suivante pour démarrer une passerelle qui fait appel à un fichier de configuration unique :

```
%OMNIHOME%\bin\nco_g_nom_passerelle.exe -name GATE2 -config  
%OMNIHOME%\etc\GATE2.conf
```

Sur les systèmes d'exploitation UNIX et Linux, vous devez exécuter les passerelles sous le contrôle de processus.

Pour plus d'informations sur l'utilisation du contrôle de processus, voir le *guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Exécution de passerelles en tant que services Windows

Pour exécuter la passerelle en tant que service Windows, procédez comme suit :

1. Pour exécuter la passerelle sur le même hôte que le serveur ObjectServer, servez-vous de la commande suivante pour l'enregistrer comme un service.

```
%OMNIHOME%\bin\nco_g_nom_passerelle.exe -install -depend NCOObjectServer
```
2. Pour exécuter la passerelle sur un hôte différent du serveur ObjectServer, utilisez la commande suivante pour l'enregistrer comme un service.

```
%OMNIHOME%\bin\nco_g_nom_passerelle.exe -install
```
3. Démarrez la passerelle à l'aide de la console de gestion des services Microsoft.

Traitement des incidents de passerelle

Lors du traitement des incidents de passerelle, commencez par vérifier le fichier journal de la passerelle.

Le fichier journal de passerelle par défaut est \$OMNIHOME/log/
NCO_nom_passerelle.log.

Vous pouvez recevoir un message d'erreur similaire au suivant :
error in srv_select () - file descriptor x is no longer active!

Ce type de message d'erreur indique que la passerelle a annulé son opération en raison de l'échec d'un module de programme de lecture ou d'écriture. Dans ce cas, vérifiez les fichiers journaux suivants :

- NCO_nom_passerelle_XRWY_WRITE.log
- NCO_nom_passerelle_XRWY_READ.log

Où X identifie le nom de la passerelle et Y sa version.

Chapitre 11. Configuration des passerelles ObjectServer

Une fois que vous avez installé ou mis à niveau les passerelles de serveur ObjectServer dans votre environnement, définissez les propriétés de la passerelle pour définir la façon dont elle fonctionne, puis définissez le mappage entre les zones des serveurs ObjectServer source et cible et définissez les tables qui sont répliquées entre les serveurs ObjectServer.

Dans une configuration à plusieurs niveaux, les passerelles ObjectServer fonctionnent de la manière suivante :

- Chaque ObjectServer de la couche de collecte possède sa propre passerelle unidirectionnelle ObjectServer dédiée qui relie le serveur ObjectServer à la couche d'agrégation.
- La couche d'agrégation inclut une paire de serveurs ObjectServer connectée par une passerelle bidirectionnelle ObjectServer pour que les serveurs restent synchronisés. La passerelle ObjectServer bidirectionnelle s'exécute sur l'hôte de secours.
- Chaque ObjectServer de la couche d'affichage possède sa propre passerelle unidirectionnelle ObjectServer dédiée qui relie le serveur ObjectServer à la couche d'agrégation. Chaque programme de lecture de passerelle d'affichage se connecte à la paire d'agrégation virtuelle alors que chaque programme d'écriture de passerelle se connecte, et est fixé, à son ObjectServer d'affichage dédié. Par conséquent, même si les programmes de lecture peuvent exécuter des opérations de reprise en ligne et de reprise par restauration entre les serveurs ObjectServer principal et de sauvegarde de la couche d'agrégation, le programme d'écriture reste connecté uniquement à son serveur ObjectServer d'affichage dédié. (Ces connexions de passerelle sont l'opposé des connexions de passerelle dans la couche de collecte.)

Pour plus d'informations sur la configuration de l'architecture à plusieurs niveaux, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Information associée:



Guide des meilleures pratiques Tivoli Netcool/OMNIbus V7.4

Passerelles ObjectServer bidirectionnelles

La Passerelle ObjectServer bidirectionnelle permet d'envoyer des alertes dans les deux directions entre deux ObjectServer. Vous pouvez utiliser des passerelles bidirectionnelles pour créer une paire de reprise en ligne de serveurs ObjectServer. Le nom de fichier exécutable d'une passerelle ObjectServer bidirectionnelle est `nco_g_objserv_bi`.

Les modifications apportées à un ObjectServer sont répliquées dans l'autre ObjectServer. Cette réplication garantit que les deux ObjectServers contiennent les mêmes alertes et vous permet de conserver un ObjectServer de sauvegarde.

Important : Assurez-vous que les données de table sont modifiées dans un seul des ObjectServers. Si elles sont modifiées dans les deux ObjectServers, il se peut que la passerelle ne synchronise pas les modifications et que les données de ligne restent dans un état indéterminé. La définition de la configuration d'une paire de reprise en ligne de serveurs ObjectServer garantit que seules les données de

l'ObjectServer principal sont modifiées, sauf lorsque le système bascule sur l'ObjectServer de sauvegarde. Pour plus d'information sur le basculement, voir le *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIBus*.

Flux d'informations

La figure suivante présente la structure d'une Passerelle ObjectServer bidirectionnelle.

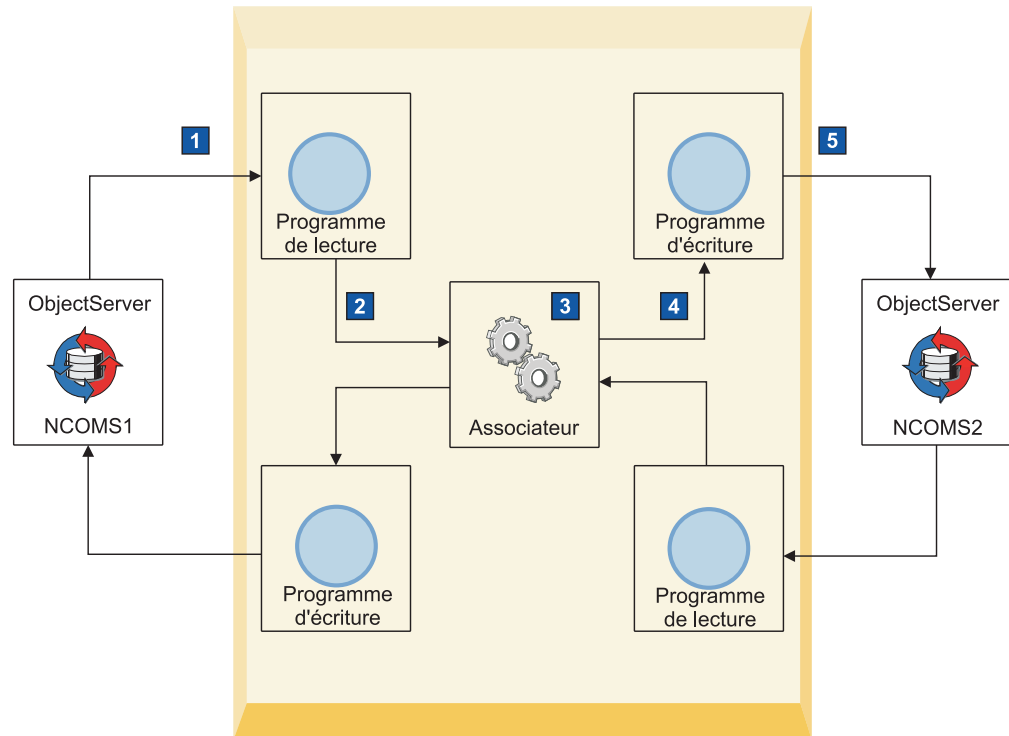


Figure 9. Passerelle ObjectServer bidirectionnelle

La passerelle bidirectionnelle est composée d'un associateur et de deux composants de lecture/écriture (un pour chaque ObjectServer). Le flux d'informations entre les composants est le suivant :

- 1** : Le programme de lecture lit les données à partir d'un seul serveur ObjectServer.
- 2** : Les données sont transmises à l'associateur de passerelle.
- 3** : L'associateur transforme les données dans un format approprié pour le second ObjectServer. Pour transformer les données, l'associateur utilise les paramètres du fichier de définition de mappe.
- 4** : L'associateur transmet les données au programme de lecture.
- 5** : Le programme d'écriture écrit les données dans le second ObjectServer.

Concepts associés:

«Passerelles ObjectServer bidirectionnelles», à la page 176

Une passerelle ObjectServer bidirectionnelle permet aux alertes de passer d'un serveur ObjectServer source à un serveur ObjectServer cible. Les modifications apportées au contenu d'un serveur ObjectServer source sont répliquées dans un serveur ObjectServer de destination, qui réplique à son tour les modifications de ses alertes sur le serveur ObjectServer source. Vous pouvez, par exemple, gérer un système avec deux serveurs ObjectServer configurés comme paire de reprise en

ligne.

Tâches associées:

«Configuration des opérations de reprise par restauration», à la page 229

Utilisez les propriétés de passerelle pour configurer la façon dont un ObjectServer de sauvegarde rebascule sur un ObjectServer principal une fois que ce dernier est restauré. Si la fonction de reprise par restauration est activée, la passerelle se connecte à l'ObjectServer de sauvegarde lorsque la connexion à l'ObjectServer principal est perdue. Lorsque l'ObjectServer principal devient actif, la passerelle s'y reconnecte.

Référence associée:

«Programmes d'écriture et reprise par restauration de la passerelle», à la page 181

La reprise en ligne se produit lorsqu'une passerelle perd sa connexion avec le serveur ObjectServer principal et se connecte à un serveur ObjectServer de secours. La fonction de reprise par restauration permet à la passerelle de se reconnecter au serveur ObjectServer principal lorsqu'il est à nouveau actif.

Passerelles ObjectServer unidirectionnelles

La Passerelle ObjectServer unidirectionnelle permet d'envoyer des alertes dans une direction, d'un ObjectServer source vers un ObjectServer de destination. Le nom de fichier exécutable d'une passerelle ObjectServer unidirectionnelle est `nco_g_objserv_uni`.

Les changements apportés à l'ObjectServer source sont reflétés dans l'ObjectServer de destination, et les changements apportés à l'ObjectServer de destination sont reflétés dans l'ObjectServer source.

Flux d'informations

La figure ci-dessous présente la structure d'une Passerelle ObjectServer unidirectionnelle.

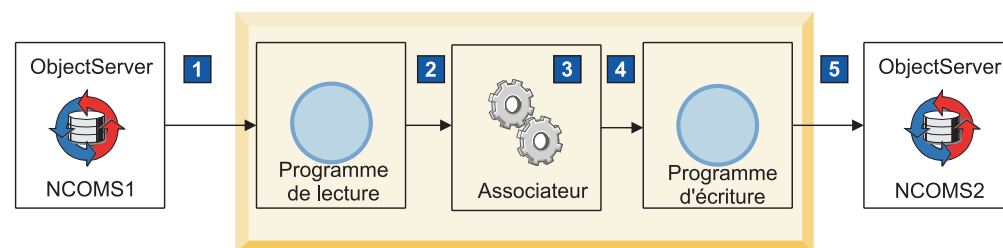


Figure 10. Passerelle ObjectServer unidirectionnelle

La passerelle unidirectionnelle est composée d'un composant associateur, d'un composant de lecture et d'un composant d'écriture. Le flux d'informations entre les composants est le suivant :

- 1** : Le programme de lecture de la passerelle lit les données de l'ObjectServer source, NCOMS1.
- 2** : Les données sont transmises à l'associateur de passerelle.
- 3** : Les données sont mappées des tables et colonnes de l'ObjectServer source vers celles de l'ObjectServer cible, en fonction des paramètres du fichier de définition de mappe.
- 4** : L'associateur transmet les données au programme d'écriture de la passerelle.

5 : Le programme d'écriture écrit les données dans l'ObjectServer cible, NCOMS2.

Concepts associés:

«Passerelles ObjectServer unidirectionnelles», à la page 175

Une passerelle ObjectServer unidirectionnelle permet aux alertes de passer d'un serveur ObjectServer source à un serveur ObjectServer cible. Les modifications apportées dans le serveur ObjectServer source sont répliquées dans le serveur ObjectServer de destination, mais les modifications apportées dans le serveur ObjectServer de destination ne sont pas répliquées dans le serveur ObjectServer source.

Configuration des propriétés de passerelles ObjectServer

Pour définir l'environnement opérationnel de la passerelle, notamment les détails de connexion et l'emplacement des autres fichiers de configuration, précisez les propriétés dans le fichier de propriétés. Il s'agit d'un fichier texte qui contient un ensemble de propriétés et leurs valeurs correspondantes.

Par défaut, la passerelle ObjectServer lit le fichier de propriétés dans le fichier `$OMNIHOME/etc/NCO_GATE.props`. Vous pouvez utiliser l'option de ligne de commande `-name` pour renommer ce fichier ou l'option `-propsfile` pour demander à la passerelle de lire un autre fichier de propriétés. Des exemples de fichier de propriétés accessibles en écriture sont fournis dans `$NCHOME/omnibus/gates/objserv_uni/objserv_uni.props` pour les passerelles bidirectionnelles et dans `$NCHOME/omnibus/gates/objserv_bi/objserv_bi.props` pour les passerelles bidirectionnelles.

Pour plus d'informations sur les propriétés communes et les propriétés Interprocess Control (IPC), reportez-vous au *guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

La plupart des tâches de configuration sont identiques pour les passerelles unidirectionnelles et les passerelles bidirectionnelles. Cependant, si une tâche de configuration exige la définition de propriétés différentes, selon si la passerelle est unidirectionnelle ou bidirectionnelle, les tâches sont décrites séparément, par exemple la configuration des opérations de reprise par restauration.

Après avoir édité le fichier de propriétés, redémarrez la passerelle pour que les modifications prennent effet.

L'architecture standard à plusieurs niveaux comprend des exemples de configuration, y compris le fichier de propriétés, pour les passerelles de la couche de collecte, de la couche d'agrégation et de la couche d'affichage. Ces configurations se trouvent dans `$NCHOME/omnibus/extensions/multitier/gateway`. Par exemple, `$NCHOME/omnibus/extensions/multitier/gateway/AGG_GATE` contient la configuration d'un serveur ObjectServer de la couche d'agrégation qui peut être déployé lorsque vous exécutez un serveur ObjectServer principal unique sur un serveur ObjectServer de sauvegarde.

Contrôle de la taille de la mémoire cache

La passerelle utilise une mémoire cache pour stocker les informations des tables qui requièrent le transfert d'un ObjectServer vers un autre. Modifiez la taille de la mémoire cache si l'ObjectServer auquel la passerelle est liée gère un grand nombre d'événements.

La principale fonction de la mémoire cache est de faciliter les opérations d'insertion de journal et de table d'informations. Lorsqu'un journal ou une information est acheminé pour l'insertion dans un ObjectServer cible, le programme d'écriture de la passerelle a besoin de connaître la série de l'état correspondant dans l'ObjectServer cible. Ces informations sont contenues dans la mémoire cache. La mémoire cache est également utilisée pour toute autre table spécifiée dans la table de définition de réplification de table.

La mémoire cache augmente les performances en fournissant à la passerelle une vue résumée en mémoire du contenu des ObjectServer auxquels elle est liée. En conséquence, la passerelle n'a pas besoin d'effectuer une requête sur un ObjectServer pour vérifier l'existence d'un événement ou la valeur Serial ou Tally d'un événement. En revanche, elle peut vérifier la mémoire cache de l'ObjectServer.

Procédure

La mémoire cache est implémentée par une table de hachage. Les performances diminuent si le nombre de lignes dans la table d'états d'ObjectServer est considérablement supérieur au nombre d'entrées de la table de hachage.

Pour contrôler la taille de la mémoire cache de table de hachage, modifiez la valeur de la propriété **Gate.CacheHashTblSize**. La valeur par défaut est 5023 éléments ou lignes. Si une table d'états possède un grand nombre de lignes, par exemple, plus de 20000, augmentez le nombre. Pour plus d'efficacité, indiquez une valeur correspondant à un nombre premier.

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Définition du niveau des messages de débogage

Vous pouvez résoudre les problèmes liés à la passerelle en consultant les messages d'erreur. La passerelle possède une fonction de gestion des erreurs, qui est fournie par la bibliothèque NGTK (Netcool/OMNIBus Gateway Toolkit). Vous pouvez indiquer quels messages sont inclus dans les fichiers de débogage.

Procédure

1. Pour spécifier que la bibliothèque NGTK consigne les messages de débogage, paramétrez la propriété **Gate.NGtkDebug** sur TRUE.

2. Pour spécifier les messages de débogage qui sont inclus dans les fichiers de débogage, paramétrez les propriétés suivantes sur TRUE ou FALSE, selon les besoins :

- **Gate.Mapper.Debug**
- **Gate.ObjectServerB.Debug**
- **Gate.ObjectServerA.Debug**

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Vérifier si la passerelle s'exécute sous le contrôle de processus

La passerelle peut être placée sous le contrôle de Process Agent (PA), auquel cas PA indique comment la passerelle s'exécute. Vous pouvez déterminer si une passerelle est sous le contrôle de PA en vérifiant les propriétés de la passerelle.

Procédure

Pour vérifier si une passerelle est sous le contrôle de PA, procédez comme suit :

- Pour vérifier si la passerelle est activée pour PA, examinez la valeur de la propriété **Gate.PAAware**. Si la propriété a pour valeur 0, cela signifie que la passerelle n'est pas activée pour PA.
- Pour le nom de l'agent PA exécutant la passerelle, vérifiez la valeur de la propriété **Gate.PAAwareName**

Important : Ne modifiez pas ces propriétés dans le fichier de propriétés de la passerelle. Ces propriétés sont tenues à jour automatiquement par le serveur d'agent de processus et fournissent des informations uniquement.

Pour plus d'informations sur le contrôle PA, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Modification de la taille de mémoire tampon

La taille de mémoire tampon contrôle le nombre d'entrées que la passerelle stocke dans sa mémoire tampon avant de les transmettre à l'ObjectServer. La valeur optimale pour la taille de mémoire tampon dépend de la taille moyenne des événements et de la vitesse du réseau. La taille de mémoire tampon par défaut est souvent suffisante. Pour réduire le temps d'attente, vous pouvez ajuster la taille de mémoire tampon.

La passerelle utilise des mémoires tampon séparées pour les serveurs ObjectServer source et de destination.

Procédure

Vous pouvez ajuster la taille de la mémoire tampon, en procédant comme suit :

1. Pour déterminer le paramètre le plus efficace pour votre système, notez la durée des opérations de resynchronisation. Modifiez ensuite les propriétés de resynchronisation, puis relancez une resynchronisation et comparez les chiffres de durée. Renouvelez cette opération, selon vos besoins.
2. Pour modifier la taille de mémoire tampon pour l'ObjectServer source et l'ObjectServer de destination, apportez les modifications appropriées aux propriétés suivantes :
 - **Gate.ObjectServerA.BufferSize**
 - **Gate.ObjectServerB.BufferSize**

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Tâches associées:

«Configuration de la resynchronisation», à la page 233

Les propriétés de passerelle permettent d'indiquer comment la passerelle est resynchronisée avec ses serveurs ObjectServer cible. Vous pouvez configurer la resynchronisation pour les passerelles ObjectServer bidirectionnelles et unidirectionnelles.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Modification du mécanisme d'authentification

La passerelle prend en charge l'authentification UNIX standard ou l'authentification PAM (Pluggable Authentication Modules). L'authentification UNIX standard est l'authentification par défaut. L'authentification PAM est requis pour exécuter la passerelle en mode FIPS 140-2.

Pour plus d'information sur l'authentification, voir le *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIBus*.

Procédure

Pour modifier le mécanisme d'authentification en PAM, paramétrez la propriété **Gate.UsePamAuth** sur TRUE.

Que faire ensuite

1. Configurez Tivoli Netcool/OMNIbus de sorte qu'il utilise PAM pour l'authentification externe. Les noms de service sont `nco_g_objserv_uni` et `nco_g_objserv_bi`. Les deux comportent le type de module `account`. Pour plus d'informations sur la configuration de PAM, voir le *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.
2. Redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Connexion à des serveurs ObjectServer sécurisés

Lorsqu'un ObjectServer s'exécute en mode sécurisé, la passerelle doit établir sa connexion soit en tant qu'utilisateur ObjectServer connu, soit en tant que superutilisateur. L'utilisateur doit disposer de droits d'accès aux tables à répliquer, ainsi qu'à certaines tables système. La configuration d'ObjectServer par défaut inclut un groupe appelé Gateway, qui dispose des droits d'accès requis.

Pour plus d'informations sur l'exécution de l'ObjectServer en mode sécurisé, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Resynchronisation des données de sécurité avec des serveurs ObjectServer sécurisés

Si vous souhaitez resynchroniser les données de sécurité lorsque les serveurs ObjectServer s'exécutent en mode sécurisé, exécutez la passerelle en tant que superutilisateur. Si vous ne procédez pas ainsi, lorsque vous tentez la resynchronisation, la passerelle quitte et aucune donnée de sécurité n'est transférée vers l'ObjectServer de destination.

Aucune donnée de sécurité n'est transférée car la passerelle supprime les autorisations de destination et ne peut pas insérer de lignes copiées à partir de la table source. Si vous exécutez la passerelle en tant que superutilisateur, ce problème est ignoré car aucune autorisation n'a besoin d'être définie explicitement. Pour plus d'informations sur l'ObjectServer, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249
Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242
Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Configuration des opérations de reprise par restauration

Utilisez les propriétés de passerelle pour configurer la façon dont un ObjectServer de sauvegarde rebascule sur un ObjectServer principal une fois que ce dernier est restauré. Si la fonction de reprise par restauration est activée, la passerelle se connecte à l'ObjectServer de sauvegarde lorsque la connexion à l'ObjectServer principal est perdue. Lorsque l'ObjectServer principal devient actif, la passerelle s'y reconnecte.

Important : Bien que vous puissiez utiliser des passerelles ObjectServer pour contrôler la reprise par restauration des clients, pour réduire le risque de perte de données, le comportement de reprise par restauration doit être géré par une paire de serveurs ObjectServer de reprise en ligne et non pas par les clients eux-mêmes. Le risque de perte de données peut avoir lieu si les clients sont repris par restauration sur un serveur ObjectServer principal avant que la synchronisation soit terminée.

Configuration des opérations de reprise par restauration pour les passerelles ObjectServer unidirectionnelles

Deux serveurs ObjectServer peuvent être configurés en tant que paire, un serveur jouant le rôle de serveur principal et l'autre serveur étant le serveur de sauvegarde. Vous pouvez spécifier comment l'ObjectServer de sauvegarde effectue la reprise par restauration sur l'ObjectServer principal.

Procédure

Ces instructions s'appliquent uniquement aux programmes de lecture ou d'écriture de passerelle qui se connectent à une paire ObjectServer virtuelle.

1. Dans l'ObjectServer de sauvegarde, paramétrez la propriété **BackupObjectServer** sur TRUE, puis redémarrez l'ObjectServer, si nécessaire.
2. Dans le fichier de propriétés de la passerelle, définissez les propriétés **Gate.Reader.Failback** et **Gate.Writer.Failback** sur TRUE.
3. Pour déterminer la fréquence à laquelle les parties de programme de lecture et d'écriture de la passerelle interrogent l'ObjectServer principal ayant échoué, définissez les propriétés **Gate.Reader.FailbackTimeout** et/ou **Gate.Writer.FailbackTimeout**.
4. Pour indiquer les paires ObjectServer, définissez les propriétés **Gate.ObjectServerA.Server** et **Gate.ObjectServerB.Server** sur le nom de serveur ObjectServer virtuel.

Résultats

Lorsque l'ObjectServer principal échoue, le programme de lecture et le programme d'écriture basculent sur l'ObjectServer de sauvegarde sans s'arrêter. Lorsque le programme de lecture ou d'écriture détecte qu'il est désormais connecté à un ObjectServer de sauvegarde, il interroge régulièrement le retour de l'ObjectServer principal. Lorsque l'ObjectServer principal est de nouveau détecté, le programme de lecture ou d'écriture rebascule automatiquement sur l'ObjectServer principal.

Exemple

La figure suivante montre un exemple de configuration de reprise par restauration avec passerelle unidirectionnelle.

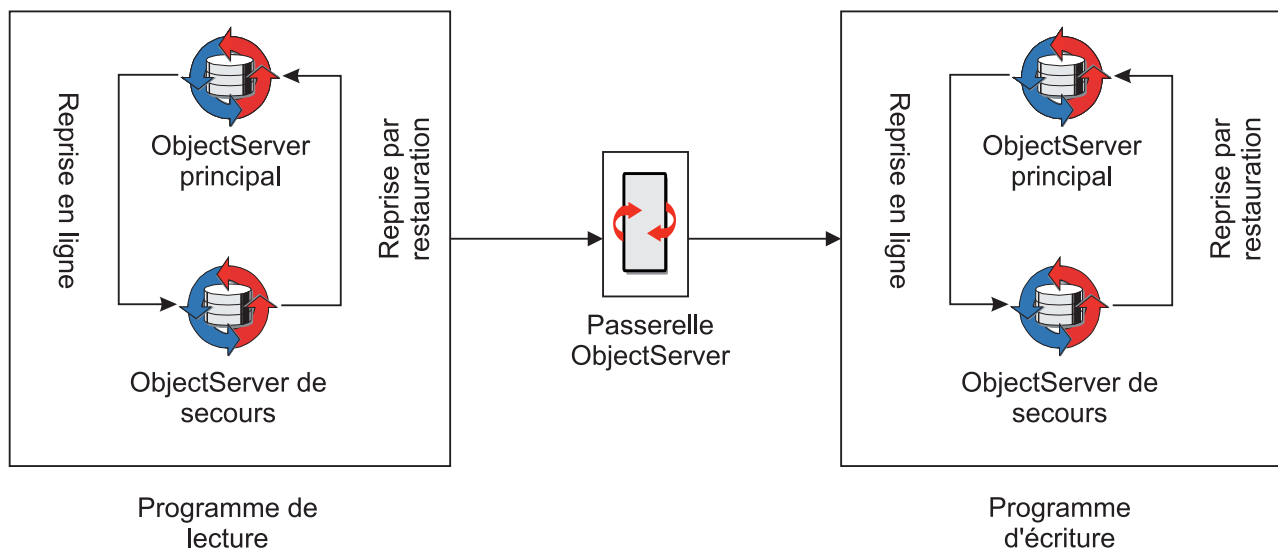


Figure 11. Configuration de la reprise par restauration pour les passerelles ObjectServer unidirectionnelles

Généralement, dans une architecture à plusieurs niveaux, la couche de collecte possède un serveur ObjectServer unique comme programme de lecture et une paire de serveurs ObjectServer comme programme d'écriture. La couche d'agrégation possède une paire de serveurs ObjectServer comme programmes de lecture et d'écriture. La couche d'affichage possède un serveur ObjectServer unique comme programme de lecture et une paire de serveurs ObjectServer comme programme d'écriture.

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Programmes d'écriture et reprise par restauration de la passerelle», à la page 181
La reprise en ligne se produit lorsqu'une passerelle perd sa connexion avec le serveur ObjectServer principal et se connecte à un serveur ObjectServer de secours. La fonction de reprise par restauration permet à la passerelle de se reconnecter au serveur ObjectServer principal lorsqu'il est à nouveau actif.

Configuration des opérations de reprise par restauration pour les passerelles ObjectServer bidirectionnelles

Si les données sont modifiées sur un seul serveur ObjectServer d'une paire virtuelle, vous pouvez configurer une passerelle bidirectionnelle pour la reprise par restauration. L'opération est identique à celle d'une passerelle unidirectionnelle mais vous devez configurer des propriétés différentes.

Restriction : La reprise par restauration n'a pas lieu lorsque la passerelle bidirectionnelle se connecte à un ObjectServer qui ne se trouve pas dans une paire virtuelle. Par exemple, si vous utilisez une passerelle bidirectionnelle pour gérer un ObjectServer de sauvegarde, la reprise par restauration n'a pas lieu.

Procédure

Pour configurer des opérations de reprise par restauration, procédez comme suit :

1. Dans l'ObjectServer de sauvegarde, paramétrez la propriété **BackupObjectServer** sur TRUE, puis redémarrez l'ObjectServer, si nécessaire.
2. Définissez les propriétés suivantes dans le fichier de propriétés de la passerelle.
 - Si l'ObjectServer A possède un ObjectServer de sauvegarde, paramétrez la propriété **Gate.ObjectServerA.Failback** sur TRUE.
 - Si l'ObjectServer B possède un ObjectServer de sauvegarde, paramétrez la propriété **ObjectServerGate.ObjectServerB.Failback** sur TRUE.
3. Pour spécifier la fréquence à laquelle la passerelle cherche à se reconnecter à l'ObjectServer défaillant, définissez la valeur des propriétés **Gate.ObjectServerA.FailbackTimeout** et **Gate.ObjectServerB.FailbackTimeout**.
4. Pour indiquer les paires ObjectServer, définissez les propriétés **Gate.ObjectServerA.Server** et **Gate.ObjectServerB.Server** sur le nom de serveur ObjectServer virtuel.

Résultats

Lorsque l'ObjectServer principal échoue, la passerelle bascule sur l'ObjectServer de sauvegarde sans s'arrêter. Lorsque la passerelle est connectée à un ObjectServer de sauvegarde, elle cherche régulièrement à se connecter à nouveau à l'ObjectServer principal. Lorsque l'ObjectServer principal est de nouveau détecté, la passerelle rebasculé automatiquement sur l'ObjectServer principal.

Exemple

La figure ci-dessous présente un exemple d'opération de reprise par restauration pour une passerelle bidirectionnelle.

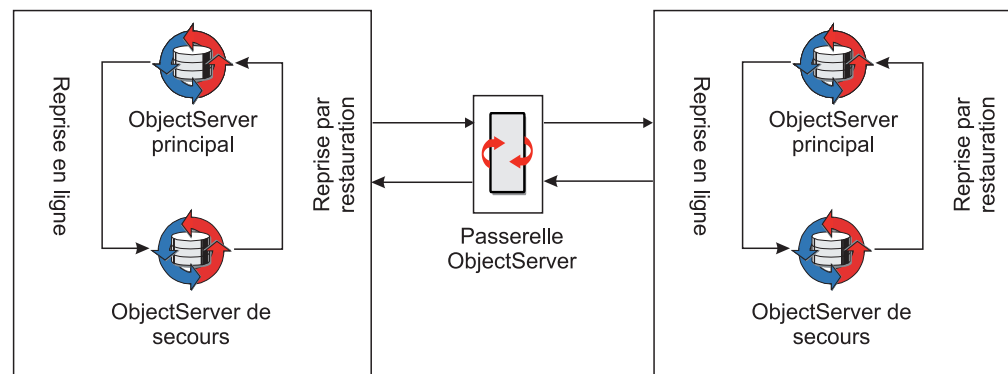


Figure 12. Passerelle ObjectServer bidirectionnelle

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Programmes d'écriture et reprise par restauration de la passerelle», à la page 181
La reprise en ligne se produit lorsqu'une passerelle perd sa connexion avec le serveur ObjectServer principal et se connecte à un serveur ObjectServer de secours. La fonction de reprise par restauration permet à la passerelle de se reconnecter au serveur ObjectServer principal lorsqu'il est à nouveau actif.

Configuration des opérations de stockage et de retransmission

Vous pouvez configurer les passerelles bidirectionnelles et unidirectionnelles de sorte qu'elles stockent et retransmettent les données des tables répliquées si l'ObjectServer de destination passe hors connexion.

Configuration des opérations de stockage et de retransmission pour les passerelles ObjectServer unidirectionnelles

Les passerelles unidirectionnelles prennent en charge le stockage et la retransmission sur n'importe quelle table lorsque l'ObjectServer cible passe hors connexion. Vous pouvez configurer la passerelle de sorte qu'elle stocke et retransmette toutes ou aucune des tables répliquées.

Procédure

Pour configurer la fonction de stockage et de retransmission :

1. Paramétrez la propriété **Gate.Writer.SAF** sur TRUE.
2. Pour spécifier le fichier dans lequel sont consignées les alertes lorsque l'ObjectServer de destination est hors ligne, paramétrez la propriété **Gate.Writer.SAFFile**.
3. Redémarrez la passerelle.

Que faire ensuite

Pour désactiver la fonction de stockage et de transmission, définissez la propriété **Gate.Writer.SAF** sur FALSE et redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Configuration des opérations de stockage et de retransmission pour les passerelles ObjectServer bidirectionnelles

La passerelle prend en charge le stockage et la retransmission sur n'importe quelle table lorsque l'ObjectServer cible passe hors connexion. Vous pouvez configurer la passerelle de sorte qu'elle stocke et retransmette toutes ou aucune des tables répliquées.

Procédure

1. Pour configurer la fonction de stockage et de retransmission pour l'ObjectServer A :
 - a. Paramétrez la propriété **Gate.ObjectServerA.SAF** sur TRUE.
 - b. Pour spécifier le fichier dans lequel sont consignées les alertes lorsque l'ObjectServer de destination est hors ligne, paramétrez la propriété **Gate.ObjectServerA.SAFFile**.
2. Pour configurer la fonction de stockage et de retransmission pour l'ObjectServer B :
 - a. Paramétrez la propriété **Gate.ObjectServerB.SAF** sur TRUE.

- b. Pour spécifier le fichier dans lequel sont consignées les alertes lorsque l'ObjectServer de destination est hors ligne, paramétrez la propriété **Gate.ObjectServerB.SAFFile**.
3. Après avoir redéfini les propriétés, redémarrez la passerelle.

Que faire ensuite

Pour désactiver la fonction de stockage et de retransmission, paramétrez la propriété **Gate.ObjectServerA.SAF** sur FALSE et la propriété **Gate.ObjectServerB.SAF** sur FALSE, puis redémarrez la passerelle.

Référence associée:

«Programmes d'écriture et reprise par restauration de la passerelle», à la page 181
La reprise en ligne se produit lorsqu'une passerelle perd sa connexion avec le serveur ObjectServer principal et se connecte à un serveur ObjectServer de secours. La fonction de reprise par restauration permet à la passerelle de se reconnecter au serveur ObjectServer principal lorsqu'il est à nouveau actif.

«Propriétés des passerelles bidirectionnelles», à la page 249
Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242
Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Configuration de la resynchronisation

Les propriétés de passerelle permettent d'indiquer comment la passerelle est resynchronisée avec ses serveurs ObjectServer cible. Vous pouvez configurer la resynchronisation pour les passerelles ObjectServer bidirectionnelles et unidirectionnelles.

Tâches associées:

«Modification de la taille de mémoire tampon», à la page 227

La taille de mémoire tampon contrôle le nombre d'entrées que la passerelle stocke dans sa mémoire tampon avant de les transmettre à l'ObjectServer. La valeur optimale pour la taille de mémoire tampon dépend de la taille moyenne des événements et de la vitesse du réseau. La taille de mémoire tampon par défaut est souvent suffisante. Pour réduire le temps d'attente, vous pouvez ajuster la taille de mémoire tampon.

Configuration de la resynchronisation pour les passerelles ObjectServer unidirectionnelles

Pour indiquer comment les passerelles unidirectionnelles sont resynchronisées avec les serveurs ObjectServer, définissez les propriétés de resynchronisation qui se trouvent dans le fichier de propriétés. Les passerelles ObjectServer unidirectionnelles disposent de moins de propriétés de resynchronisation que les passerelles ObjectServer bidirectionnelles.

Procédure

Pour configurer la resynchronisation, définissez les propriétés décrites dans le tableau ci-dessous.

Tableau 47. Propriétés de resynchronisation et options de ligne de commande des Passerelles ObjectServer unidirectionnelles

Nom de propriété	Option de ligne de commande	Description
Gate.Resync.Enable <i>booléen</i>	<i>-resyncenable booléen</i>	Cette propriété permet de spécifier que la passerelle doit utiliser la resynchronisation. La valeur par défaut est TRUE.
Gate.Resync.Type <i>chaîne</i>	<i>-resynctype chaîne</i>	Cette propriété permet d'indiquer comment la passerelle doit resynchroniser les données de table entre les serveurs ObjectServer lorsqu'elle démarre ou restaure une connexion perdue. La passerelle resynchronise les tables définies dans le fichier de définition de réplication de table. Voir les informations situées à la suite de tableau pour obtenir les valeurs sur lesquelles vous pouvez définir cette propriété. La valeur par défaut est NORMAL.
Gate.Resync.LockType <i>chaîne</i>	<i>-resynclocktype chaîne</i>	Cette propriété permet de spécifier les options de verrouillage des serveurs ObjectServer source et de destination pendant la resynchronisation des événements. Choisissez l'une des options suivantes : <ul style="list-style-type: none"> • FULL : la passerelle verrouille les serveurs ObjectServer source et cible. • PARTIAL : la passerelle ne bloque que l'ObjectServer de destination. • NONE : la passerelle ne bloque ni l'ObjectServer source, ni l'ObjectServer cible. <div style="background-color: #e0e0e0; padding: 2px; display: inline-block; margin-top: 5px;">Fix Pack 1</div> La valeur par défaut est PARTIAL.

Vous pouvez définir la propriété **Gate.Resync.Type** sur l'une des valeurs suivantes :

- **NORMAL** : Pour chaque table, la passerelle supprime toutes les données de l'ObjectServer esclave. La passerelle re-transfère ensuite l'ensemble complet des tables du maître à l'esclave. Avec ce type de resynchronisation, le maître et l'esclave sont totalement synchronisés. Cependant, les lignes de table figurant sur l'esclave et non sur le maître sont perdues. En outre, si des lignes de table figurent sur le maître et l'esclave, la copie de la ligne figurant sur le maître est conservée sur le maître et sur l'esclave. Toutes les mises à jour précédentes de la ligne sur l'esclave sont perdues.
- **UPDATE** : Pour chaque table, la passerelle génère un cache contenant toutes les lignes des ObjectServer maître et esclave. La passerelle examine ensuite le contenu du cache pour chaque table et compare les données de ligne du maître à celles de l'esclave. Les données sont resynchronisées comme suit :
 - Les lignes de l'esclave qui figurent également sur le maître sont mises à jour avec les données issues du maître, si les données du maître sont différentes de celles de l'esclave.
 - Les lignes figurant sur le maître et non sur l'esclave sont copiées sur l'esclave.
 - Les lignes de l'esclave qui ne figurent pas sur le maître sont conservées.

Avec ce type de resynchronisation, aucun événement n'est perdu, mais il se peut que les serveurs ObjectServer maître et esclave ne soient pas totalement synchronisés.
- **MINIMUM** : Cette option fonctionne de la même manière que UPDATE. Suite à l'intervention de la passerelle :
 1. Les mémoires caches des événements des ObjectServers maître et esclave sont comparées.

2. Toute ligne apparaissant sur l'esclave mais pas sur le maître est écrite sur le maître.

Une fois la resynchronisation TWOWAYUPDATE effectuée, le maître et l'esclave comportent des lignes identiques.

- MINIMAL : Cette option fonctionne de la même manière que UPDATE. En outre, les événements (c'est-à-dire les lignes de la table alerts.status) qui figurent sur l'esclave et non sur le maître sont marqués pour suppression. Pour marquer ces événements à des fins de suppression, la passerelle se comporte comme suit :
 1. Pour chaque ligne de la table alerts.status sur l'ObjectServer esclave qui ne figure pas sur le maître, la zone OldRow est paramétrée sur 1.
 2. Le déclencheur pass_deletes s'exécute sur l'ObjectServer esclave et supprime toutes les lignes dont la zone OldRow est paramétrée sur 1.

L'avantage d'une resynchronisation MINIMAL est que les ObjectServers maître et esclave sont totalement synchronisés mais moins de données sont envoyées pendant le processus de resynchronisation. La resynchronisation MINIMAL consomme moins de données car les lignes ne sont pas supprimées puis recopiées, contrairement à ce qui se passe avec une resynchronisation NORMAL.

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Configuration de la resynchronisation pour les passerelles ObjectServer bidirectionnelles

Pour indiquer comment les passerelles bidirectionnelles sont resynchronisées avec les serveurs ObjectServer, définissez les propriétés de resynchronisation qui se trouvent dans le fichier de propriétés. Les passerelles ObjectServer bidirectionnelles disposent de plus de propriétés de resynchronisation que les passerelles ObjectServer unidirectionnelles.

Procédure

Pour configurer la resynchronisation, définissez les propriétés décrites dans le tableau ci-dessous.

Tableau 48. Propriétés de resynchronisation et options de ligne de commande des Passerelles ObjectServer bidirectionnelles

Nom de propriété	Option de ligne de commande	Description
Gate.Resync.Enable <i>booléen</i>	-resyncenable <i>booléen</i>	Cette propriété permet de spécifier que la passerelle doit utiliser la resynchronisation. La valeur par défaut est TRUE.

Tableau 48. Propriétés de resynchronisation et options de ligne de commande des Passerelles ObjectServer bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Resync.Master chaîne	-resyncmaster chaîne	<p>Cette propriété permet d'indiquer à la passerelle quel ObjectServer elle doit toujours considérer comme étant le maître lors d'une resynchronisation.</p> <p>Les valeurs valides sont ObjectServerA et ObjectServerB.</p> <p>La valeur par défaut est "", soit "aucun maître". Si aucun maître n'est défini, la passerelle exécute une procédure visant à identifier le maître.¹</p>
Gate.Resync.Preferred chaîne	-resyncpreferred chaîne	<p>Cette propriété permet d'indiquer à la passerelle quel ObjectServer elle doit considérer comme étant le maître lors d'une resynchronisation si Gate.Resynch.Master est omis, si aucune sauvegarde n'est définie et si le temps de disponibilité est le même pour chaque ObjectServer.¹</p> <p>Les valeurs valides sont ObjectServerA et ObjectServerB.</p> <p>La valeur par défaut est "".</p>
Gate.Resync.Type chaîne	-resynctype chaîne	<p>Cette propriété permet d'indiquer comment la passerelle doit resynchroniser les données de table entre les serveurs ObjectServer lorsqu'elle démarre ou restaure une connexion perdue. La passerelle resynchronise les tables définies dans le fichier de définition de réplication de table.²</p> <p>La valeur par défaut est NORMAL.</p>
Gate.Resync.LockType chaîne	-resynclocktype chaîne	<p>Cette propriété permet de spécifier les options de verrouillage des serveurs ObjectServer source et de destination pendant la resynchronisation des événements.</p> <p>Vous avez ensuite plusieurs options :</p> <ul style="list-style-type: none"> • FULL : la passerelle verrouille les serveurs ObjectServer source et cible. • PARTIAL : la passerelle ne bloque que l'ObjectServer de destination. • NONE : la passerelle ne bloque ni l'ObjectServer source, ni l'ObjectServer cible. <p>Fix Pack 1 La valeur par défaut est PARTIAL.</p>

Tableau 48. Propriétés de resynchronisation et options de ligne de commande des Passerelles ObjectServer bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Notes du tableau :		
<p>1. Si la propriété Gate.Resync.Master n'est pas définie, la passerelle agit comme décrit ci-après pour identifier le maître. Les propriétés listées dans le fichier de propriétés de la passerelle et dans les fichiers de propriétés de chaque ObjectServer sont utilisées.</p> <p>a. La passerelle vérifie la propriété Gate.Resync.Master dans son fichier de propriétés. Si la propriété est définie, l'ObjectServer indiqué pour cette propriété est utilisé.</p> <p>b. Dans le cas contraire, la passerelle vérifie la propriété BackupObjectServer dans le fichier de propriétés de chaque ObjectServer et opère comme suit :</p> <ul style="list-style-type: none"> • Si la propriété prend la valeur TRUE pour l'un des ObjectServers (autrement dit, si celui-ci est défini en tant que serveur de secours), la passerelle vérifie la propriété ActingPrimary de cet ObjectServer. Si la propriété prend la valeur TRUE, l'ObjectServer de secours est utilisé. • Si la propriété ne prend la valeur TRUE pour aucun des ObjectServers, la passerelle passe à l'étape 1c. <p>c. Si aucun ObjectServer de secours n'est défini, ou si la propriété ActingPrimary de l'ObjectServer de secours prend la valeur FALSE, la passerelle vérifie le temps de disponibilité de chaque ObjectServer et opère comme suit :</p> <ul style="list-style-type: none"> • L'ObjectServer avec le temps de disponibilité le plus long est utilisé. • Si les temps de disponibilité sont identiques, la passerelle vérifie la propriété Gate.Resync.Preferred dans son fichier de propriétés. Si un ObjectServer est spécifié pour la propriété, cet ObjectServer est utilisé. • Si la propriété n'est pas définie, l'ObjectServer utilisé est celui qui est spécifié en tant que ObjectServerA dans le fichier de propriétés de la passerelle. <p>2. Vous pouvez indiquer l'une des valeurs suivantes pour la propriété Gate.Resync.Type :</p> <ul style="list-style-type: none"> • NORMAL : Pour chaque table, la passerelle supprime toutes les données de l'ObjectServer esclave. La passerelle re-transfère ensuite l'ensemble complet des tables du maître à l'esclave. Avec ce type de resynchronisation, le maître et l'esclave sont totalement synchronisés. Cependant, les lignes de table figurant sur l'esclave et non sur le maître sont perdues. En outre, si des lignes de table figurent sur le maître et l'esclave, la copie de la ligne figurant sur le maître est conservée sur le maître et sur l'esclave. Toutes les mises à jour précédentes de la ligne sur l'esclave sont perdues. • UPDATE : Pour chaque table, la passerelle génère un cache contenant toutes les lignes des ObjectServer maître et esclave. La passerelle examine ensuite le contenu du cache pour chaque table et compare les données de ligne du maître à celles de l'esclave. Les données sont resynchronisées comme suit : <ul style="list-style-type: none"> – Les lignes de l'esclave qui figurent également sur le maître sont mises à jour avec les données issues du maître, si les données du maître sont différentes de celles de l'esclave. – Les lignes figurant sur le maître et non sur l'esclave sont copiées sur l'esclave. – Les lignes de l'esclave qui ne figurent pas sur le maître sont conservées. <p>Avec ce type de resynchronisation, aucun événement n'est perdu, mais il se peut que les serveurs ObjectServer maître et esclave ne soient pas totalement synchronisés.</p> • MINIMUM : Cette option fonctionne de la même manière que UPDATE. Suite à l'intervention de la passerelle : <ul style="list-style-type: none"> a. Les mémoires caches des événements des ObjectServers maître et esclave sont comparées. b. Toute ligne apparaissant sur l'esclave mais pas sur le maître est écrite sur le maître. <p>Une fois la resynchronisation TWOWAYUPDATE effectuée, le maître et l'esclave comportent des lignes identiques.</p> • MINIMAL : Cette option fonctionne de la même manière que UPDATE. En outre, les événements (c'est-à-dire les lignes de la table alerts.status) qui figurent sur l'esclave et non sur le maître sont marqués pour suppression. Pour marquer ces événements à des fins de suppression, la passerelle se comporte comme suit : <ul style="list-style-type: none"> a. Pour chaque ligne de la table alerts.status sur l'ObjectServer esclave qui ne figure pas sur le maître, la zone OldRow est paramétrée sur 1. b. Le déclencheur pass_deletes s'exécute sur l'ObjectServer esclave et supprime toutes les lignes dont la zone OldRow est paramétrée sur 1. <p>L'avantage d'une resynchronisation MINIMAL est que les ObjectServers maître et esclave sont totalement synchronisés mais moins de données sont envoyées pendant le processus de resynchronisation. La resynchronisation MINIMAL consomme moins de données car les lignes ne sont pas supprimées puis recopiées, contrairement à ce qui se passe avec une resynchronisation NORMAL.</p> 		

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249
 Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242
 Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Propriétés de passerelles ObjectServer génériques

Certaines propriétés sont partagées par des passerelles ObjectServer unidirectionnelles et bidirectionnelles.

Pour plus d'informations sur les propriétés de passerelles génériques et les propriétés Interprocess Communication (IPC), voir le *Guide des sondes et des passerelles IBM Netcool/OMNIbus*. Le tableau ci-dessous décrit les propriétés partagées par les passerelles ObjectServer unidirectionnelles et bidirectionnelles.

Tableau 49. Propriétés et options de ligne de commande de passerelles ObjectServer génériques

Nom de propriété	Option de ligne de commande	Description
Gate.CacheHashTblSize entier	-chashtblsize entier	Cette propriété permet de spécifier la taille (en éléments) que la passerelle alloue à la mémoire cache de la table de hachage. La valeur par défaut est 5023.
Gate.MapFile chaîne	-mapfile chaîne	Cette propriété permet de spécifier l'emplacement du fichier de définition de mappe. L'emplacement par défaut est \$OMNIHOME/gates/objserv_uni/objserv_uni.map pour les passerelles unidirectionnelles et \$OMNIHOME/gates/objserv_bi/objserv_bi.map pour les passerelles bidirectionnelles.
Gate.StartupCmdFile chaîne	-startupcmdfile chaîne	Cette propriété permet de spécifier l'emplacement du fichier de commandes de démarrage. L'emplacement par défaut est \$OMNIHOME/objserv_uni/objserv_uni.startup.cmd pour les passerelles unidirectionnelles et \$OMNIHOME/objserv_bi/objserv_bi.startup.cmd pour les passerelles bidirectionnelles.

Tableau 49. Propriétés et options de ligne de commande de passerelles ObjectServer génériques (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Transfer.FailoverSyncRate entier	-fsyncrate entier	<p>Cette propriété permet de spécifier la durée (en secondes) de la synchronisation de reprise en ligne.</p> <p>La valeur par défaut est 60, c'est-à-dire que toutes les 60 secondes, la passerelle transfère le contenu de toute table définie pour la synchronisation de reprise en ligne depuis le serveur ObjectServer source vers le serveur ObjectServer cible.</p>
Gate.NGtkDebug booléen	-ngtkdebug booléen	<p>Cette propriété permet de spécifier si la bibliothèque NGTK consigne les messages de débogage.</p> <p>La valeur par défaut est TRUE.</p> <p>Conseil : Utilisez les propriétés suivantes pour spécifier les messages de débogage inclus dans le fichier journal de débogage :</p> <ul style="list-style-type: none"> • Gate.Mapper.Debug • Gate.Reader.Debug • Gate.Writer.Debug
Gate.PAAware entier	-paaware entier	<p>Cette propriété est réservée à l'agent de processus (PA) et indique si la passerelle est activée pour PA. Elle est incluse dans le fichier de propriétés pour information uniquement. Ne modifiez pas cette propriété.</p> <p>La valeur par défaut est " ", ce qui signifie que la passerelle n'est pas activée pour PA.</p>

Tableau 49. Propriétés et options de ligne de commande de passerelles ObjectServer génériques (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.PAAwareName chaîne	-paname chaîne	Cette propriété est réservée à l'agent de processus (PA) et indique le nom du PA qui contrôle la passerelle. Elle est incluse dans le fichier de propriétés pour information uniquement. Ne modifiez pas cette propriété. La valeur par défaut est " ", ce qui signifie qu'aucun PA ne contrôle la passerelle.
Gate.UsePamAuth booléen	-usepamauth booléen	Systèmes d'exploitation UNIX uniquement : cette propriété permet de spécifier si l'authentification PAM est utilisée. La valeur par défaut est FALSE. Remarque : Pour exécuter la passerelle en mode FIPS 140-2, paramétrez cette propriété sur TRUE.
Gate.UnixAdminGrp chaîne	-unixadmingroup chaîne	Systèmes d'exploitation UNIX uniquement : cette propriété permet de spécifier le groupe d'administration auquel la passerelle doit appartenir si l'authentification UNIX standard est utilisée. Valeur par défaut : ncoadmin.
MaxLogFileSize entier	-maxlogfilesize entier	Cette propriété permet de spécifier la taille, en ko, que la passerelle attribue au fichier journal. Lorsque le fichier journal atteint cette taille, la passerelle renomme le fichier journal en ajoutant au nom les caractères .old et crée un nouveau fichier journal. La valeur par défaut est 1024.

Tableau 49. Propriétés et options de ligne de commande de passerelles ObjectServer génériques (suite)

Nom de propriété	Option de ligne de commande	Description
OldTimeStamp <i>booléen</i>	<code>-oldtimestamp</code> <i>booléen</i>	<p>Cette propriété permet de spécifier le format d'obsolescence de l'horodatage utilisé par la passerelle dans le fichier journal. Les options sont les suivantes :</p> <ul style="list-style-type: none"> • TRUE : Indique le format d'horodatage spécifique aux paramètres régionaux, qui est utilisé par Tivoli Netcool/OMNibus V7.2.1 ou version antérieure. Par exemple, <code>jj/MM/AAAA hh:mm:ss AM</code> ou <code>jj/MM/AAAA hh:mm:ss PM</code> lorsque les paramètres régionaux ont pour valeur <code>en_GB</code>. • FALSE : Affiche l'horodatage au format ISO 8601, qui est <code>AAAA-MM-JJThh:mm:ss</code>, où T sépare la date et l'heure, et hh représente l'heure au format 24 heures. <p>La valeur par défaut est FALSE.</p> <p>Important : Ne paramétrez pas la propriété OldTimeStamp sur TRUE lors d'une exécution en mode UTF-8.</p>
N/D	<code>-utf8enabled</code> <i>booléen</i>	
Gate.Mapper.Debug <i>booléen</i>	<code>-mapperdebug</code> <i>booléen</i>	<p>Cette propriété permet de spécifier si la passerelle inclut des messages de débogage de l'associateur dans le journal de débogage.</p> <p>La valeur par défaut est TRUE.</p>
Gate.Mapper.ForwardHistoricDetails <i>booléen</i>	<code>-mapperforhistdtls</code> <i>booléen</i>	<p>Cette propriété permet de spécifier si la passerelle achemine toutes les informations d'historique sur la mise à jour convertie.</p> <p>La valeur par défaut est FALSE.</p>

Tableau 49. Propriétés et options de ligne de commande de passerelles ObjectServer génériques (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Mapper.ForwardHistoricJournals <i>booléen</i>	<code>-mapperforhistjrn1</code> <i>booléen</i>	Cette propriété permet de spécifier si la passerelle achemine tous les journaux historiques sur la mise à jour convertie. La valeur par défaut est FALSE.

Concepts associés:

«Autres commandes d'exécution de passerelle», à la page 262

Sur les systèmes d'exploitation UNIX, vous pouvez utiliser l'interface interactive SQL (l'utilitaire `nco_sql`) pour exécuter des commandes sur la passerelle. Outre les commandes SQL ObjectServer, vous pouvez exécuter les commandes d'exécution suivantes sur la passerelle : SHOW PROPS, GET CONFIG et FAILOVER SYNCH. Ces commandes se trouvent dans le fichier de commandes de la passerelle.

Propriétés des passerelles unidirectionnelles

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Pour plus d'informations sur les propriétés communes et les propriétés Interprocess Communication (IPC), reportez-vous au *Guide des sondes et des passerelles d'IBM Tivoli Netcool/OMNIBus SC43-0824*. Le tableau suivant décrit les propriétés de passerelle communes.

Tableau 50. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles

Nom de propriété	Option de ligne de commande	Description
Gate.Reader.CommonNames <i>chaîne</i>	<code>-readercommonnames</code> <i>chaîne</i>	Si la passerelle se connecte à un ObjectServer par SSL et que la zone Nom commun du certificat reçu ne correspond pas au nom spécifié par la propriété Gate.Reader.Server (par exemple dans une paire de reprise en ligne ou dans le paramètre d'un serveur virtuel), utilisez cette propriété pour spécifier une liste séparée par des virgules de Noms communs SSL acceptables. Le paramètre par défaut est d'utiliser la propriété Gate.Reader.Server .
Gate.Reader.Debug <i>booléen</i>	<code>-readerdebug</code> <i>booléen</i>	Cette propriété permet de spécifier si la passerelle inclut des messages de débogage du programme de lecture de la passerelle dans le journal de débogage. La valeur par défaut est TRUE.

Tableau 50. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Reader.Description chaîne	-readerdescription chaîne	Cette propriété permet de spécifier la description de la connexion du programme de lecture. Cette description sert dans les déclencheurs et vous permet de déterminer quel composant de la passerelle a tenté de réaliser une action. La valeur par défaut est "Gateway Reader".
Gate.Reader.DetailsTableName chaîne	-readerdetailstblname chaîne	Cette propriété permet de spécifier le nom de la table des détails lue par la passerelle. La valeur par défaut est alerts.details.
Gate.Reader.FailbackEnabled booléen	-readerfailbackenabled booléen	Cette propriété permet de spécifier la reprise par restauration de l'ObjectServer. La valeur par défaut est TRUE.
Gate.Reader.FailbackTimeout entier	-readerfailbacktimeout entier	Cette propriété permet de spécifier la période de temps (en secondes) que la passerelle doit attendre avant d'entrer en mode de reprise par restauration. La valeur par défaut est 30.
Gate.Reader.IDUCFlushRate entier	-readeriducflushrate entier	Ces propriétés permettent de contrôler la fréquence à laquelle la passerelle recherche des modifications de l'ObjectServer. La valeur par défaut est 0. Si vous conservez la valeur 0 pour la propriété, l'ObjectServer peut notifier à la passerelle que des modifications sont en attente. Si vous avez besoin que la passerelle capture des modifications plus détaillées des événements, paramétrez la propriété sur une valeur différente. Si vous affectez à la propriété une valeur différente de 0, il se peut que la charge sur l'ObjectServer augmente. Pour plus d'informations, consultez la description de la propriété Granularity de l'ObjectServer dans le <i>Guide d'administration d'IBM Tivoli Netcool/OMNIbus</i> .

Tableau 50. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Reader.JournalTableName chaîne	-readerjournaltblname chaîne	Cette propriété permet de spécifier le nom de la table des journaux lue par la passerelle. La valeur par défaut est alerts.journal.
Gate.Reader.LogOSSql booléen	-readerlogosql booléen	Cette propriété permet de spécifier si la passerelle enregistre toutes les commandes SQL envoyées à l'ObjectServer en mode débogage. La valeur par défaut est FALSE.
Gate.Reader.Password chaîne	-readerpassword chaîne	Cette propriété permet d'associer un mot de passe à l'utilisateur spécifié par la propriété Gate.Reader.Username . La valeur par défaut est " ".
Gate.Reader.ReconnectTimeout entier	-readerreconnecttimeout entier	Cette propriété permet de spécifier la durée (en secondes) entre deux tentatives de rétablissement de la connexion si la passerelle perd la connexion à l'ObjectServer. La valeur par défaut est 30.
Gate.Reader.Server chaîne	-readerserver chaîne	Cette propriété permet de spécifier le nom de l'ObjectServer duquel la passerelle peut lire les alertes. La valeur par défaut est NCOMS.
Gate.Reader.StatusTableName chaîne	-readerstatustblname chaîne	Cette propriété permet de spécifier le nom de la table d'états lue par la passerelle. La valeur par défaut est alerts.status.
Gate.Reader.TblReplicateDefFile chaîne	-readertblrepdeffile chaîne	Cette propriété permet de spécifier le chemin du fichier de définition de réplication de table. La valeur par défaut est \$OMNIHOME/gates/objserv_uni/objserv_uni.reader.tblrep.def.
Gate.Reader.Username chaîne	-readerusername chaîne	Cette propriété permet de spécifier le nom d'utilisateur à utiliser pour authentifier la connexion à ObjectServer. La valeur par défaut est root.

Tableau 50. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Resync.Enable <i>booléen</i>	-resyncenable <i>booléen</i>	Utilisez cette propriété pour que la passerelle resynchronise les ObjectServers une fois que la passerelle établit ou rétablit une connexion aux deux ObjectServers. Pour plus d'informations, voir «Configuration de la resynchronisation», à la page 233. La valeur par défaut est TRUE.
Gate.Resync.LockType <i>chaîne</i>	-resynclocktype <i>chaîne</i>	Cette propriété permet de spécifier les options de verrouillage des serveurs ObjectServer source et cible pendant la resynchronisation des événements. Vous disposez des options suivantes : <ul style="list-style-type: none"> • FULL : la passerelle verrouille les serveurs ObjectServer source et cible. • PARTIAL : la passerelle ne bloque que l'ObjectServer de destination. • NONE : la passerelle ne bloque ni l'ObjectServer source, ni l'ObjectServer cible. <div style="background-color: #4a69bd; color: white; padding: 2px; display: inline-block; font-weight: bold;">Fix Pack 1</div> La valeur par défaut est PARTIAL.
Gate.Resync.Type <i>chaîne</i>	-resynctype <i>chaîne</i>	Cette propriété permet d'indiquer comment la passerelle doit resynchroniser les données de table entre les serveurs ObjectServer lorsqu'elle démarre ou restaure une connexion perdue. La passerelle resynchronise les tables définies dans le fichier de définition de réplication de table. Pour plus d'informations sur les valeurs possibles pour cette propriété, voir «Options de Gate.Resync.Type», à la page 248. La valeur par défaut est NORMAL.
Gate.Writer.BufferSize <i>entier</i>	-writerbufsize <i>entier</i>	Utilisez ces propriétés pour indiquer le nombre d'entrées que la passerelle stocke dans la mémoire tampon pour cet ObjectServer avant de les effacer, si la mise en mémoire tampon est activée. La passerelle vide la mémoire tampon à la fin d'un lot d'instructions SQL, quelle que soit la taille de la mémoire tampon. Elle permet d'affiner l'efficacité de la passerelle. La valeur par défaut est 25.

Tableau 50. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Writer.CommonNames chaîne	-writercommonnames chaîne	Si la passerelle se connecte à un ObjectServer par SSL et que la zone Nom commun du certificat reçu ne correspond pas au nom spécifié par la propriété Gate.Writer.Server (par exemple dans une paire de reprise en ligne ou dans le paramètre d'un serveur virtuel), utilisez cette propriété pour spécifier une liste séparée par des virgules de Noms communs SSL acceptables. Le paramètre par défaut est d'utiliser la propriété Gate.Writer.Server .
Gate.Writer.Debug booléen	-writerdebug booléen	Cette propriété permet de spécifier si la passerelle inclut des messages de débogage du programme d'écriture de la passerelle dans le journal de débogage. La valeur par défaut est TRUE.
Gate.Writer.Description chaîne	-writerdescription chaîne	Cette propriété permet de spécifier la description de la connexion du programme d'écriture. Cette description sert dans les déclencheurs et vous permet de déterminer quel composant de la passerelle a tenté de réaliser une action. La valeur par défaut est "Gateway Writer".
Gate.Writer.Failback Enabled booléen	-writerfailback enabled booléen	Cette propriété permet de spécifier la reprise par restauration de l'ObjectServer. La valeur par défaut est TRUE.
Gate.Writer.Failback Timeout entier	-writerfailback timeout entier	Cette propriété permet de spécifier la durée (en secondes) avant laquelle la passerelle vérifie si l'ObjectServer principal est revenu et tente de se connecter à nouveau. La valeur par défaut est 30.
Gate.Writer.LogOSSql booléen	-writerlogossql booléen	Cette propriété permet de spécifier si la passerelle enregistre toutes les commandes SQL envoyées à l'ObjectServer en mode débogage. La valeur par défaut est FALSE.

Tableau 50. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Writer.Password <i>chaîne</i>	-writerpassword <i>chaîne</i>	Cette propriété permet de spécifier le mot de passe associé à l'utilisateur défini par la propriété Gate.Writer.Username . La valeur par défaut est " ".
Gate.Writer.ReconnectTimeout <i>entier</i>	-writerreconnecttimeout <i>entier</i>	Cette propriété permet de spécifier la durée (en secondes) entre deux tentatives de rétablissement de la connexion si la passerelle perd la connexion à l'ObjectServer. La valeur par défaut est 30.
Gate.Writer.RefreshCacheOnUpdate <i>booléen</i>	-writerrefcacheonupd <i>booléen</i>	Cette propriété permet de spécifier si le cache de la table de hachage de l'ObjectServer est actualisé. Vous disposez des options suivantes : <ul style="list-style-type: none"> • TRUE : le cache est resynchronisé avec l'ObjectServer cible avant le traitement de chaque collecte de mises à jour des lignes d'une table de la fenêtre IDUC en cours. • FALSE : la passerelle suppose que son cache est exact et par conséquent ne le resynchronise pas La valeur par défaut est TRUE.
Gate.Writer.SAF <i>booléen</i>	-writersaf <i>booléen</i>	Cette propriété permet de spécifier que la passerelle doit stocker toutes les entrées d'une table lorsque l'ObjectServer de destination est indisponible et ensuite les acheminer lorsque l'ObjectServer redevient disponible. La valeur par défaut est FALSE.
Gate.Writer.SAFFile <i>chaîne</i>	-writersaffile <i>chaîne</i>	Cette propriété permet de spécifier le nom du fichier dont se sert la passerelle pour stocker les entrées d'une table pendant que l'ObjectServer de destination est indisponible. La valeur par défaut est \$OMNIHOME/var/objserv_uni_NCO_GATE_Writer.store. Remarque : Ce fichier n'est utilisé que si la propriété Gate.Writer.SAF est paramétrée sur TRUE.
Gate.Writer.Server <i>chaîne</i>	-writerserver <i>chaîne</i>	Cette propriété permet de spécifier le nom de l'ObjectServer sur lequel la passerelle écrit les alertes. La valeur par défaut est REMOTE.

Tableau 50. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Writer.Username <i>chaîne</i>	<code>-writerusername</code> <i>chaîne</i>	Cette propriété permet de spécifier le nom d'utilisateur à utiliser pour authentifier la connexion à ObjectServer. Le nom d'utilisateur permet d'établir la connexion IDUC du programme d'écriture et la connexion secondaire des commandes SQL. La valeur par défaut est root.
Gate.Writer.SAF ReplayOnResync <i>booléen</i>	<code>-writersafreplay onresync</code> <i>booléen</i>	Cette propriété permet de spécifier la façon dont le fichier stockage et transmission (SAF) est réexécuté lors de la resynchronisation. Vous disposez des options suivantes : <ul style="list-style-type: none"> • TRUE : le fichier SAF est réexécuté que la propriété Gate.Resync.Enable ait été paramétrée sur TRUE ou non. • FALSE : le fichier SAF n'est réexécuté que lorsque la propriété Gate.Resync.Enable a été paramétrée sur FALSE. La valeur par défaut est FALSE.
Gate.Writer. UseBulkInsCmd <i>booléen</i>	<code>-usebulkinscmd</code> <i>booléen</i>	Cette propriété permet d'utiliser des insertions par blocs pour accélérer la resynchronisation. Vous disposez des options suivantes : <ul style="list-style-type: none"> • TRUE : la passerelle modifie le format de l'instruction d'insertion qu'elle envoie à l' ObjectServer, lui permettant ainsi de traiter les insertions par blocs de façon plus efficace. • FALSE : la passerelle n'effectue aucun changement à l'instruction d'insertion avant d'envoyer les événements à l'ObjectServer. La valeur par défaut est FALSE.

Options de Gate.Resync.Type

Vous pouvez définir la propriété **Gate.Resync.Type** sur l'une des valeurs suivantes :

- **NORMAL** : Pour chaque table, la passerelle supprime toutes les données de l'ObjectServer esclave. La passerelle re-transfère ensuite l'ensemble complet de tables du maître à l'esclave. Avec ce type de resynchronisation, le maître et l'esclave sont totalement synchronisés. Cependant, les lignes de table figurant sur l'esclave et non sur le maître sont perdues. En outre, si des lignes de table

figurent sur le maître et l'esclave, la copie de la ligne figurant sur le maître est conservée sur le maître et sur l'esclave. Toutes les mises à jour précédentes de la ligne sur l'esclave sont perdues.

- **UPDATE** : Pour chaque table, la passerelle génère un cache contenant toutes les lignes des ObjectServer maître et esclave. La passerelle examine ensuite le contenu du cache pour chaque table et compare les données de ligne du maître à celles de l'esclave. Les données sont resynchronisées comme suit :
 - Les lignes de l'esclave qui figurent également sur le maître sont mises à jour avec les données issues du maître, si les données du maître sont différentes de celles de l'esclave.
 - Les lignes figurant sur le maître et non sur l'esclave sont copiées sur l'esclave.
 - Les lignes de l'esclave qui ne figurent pas sur le maître sont conservées.

Avec ce type de resynchronisation, aucun événement n'est perdu, mais il se peut que les serveurs ObjectServer maître et esclave ne soient pas totalement synchronisés.

- **MINIMUM** : Cette option fonctionne de la même manière que UPDATE. Suite à l'intervention de la passerelle :
 1. Les mémoires caches des événements des ObjectServers maître et esclave sont comparées.
 2. Toute ligne apparaissant sur l'esclave mais pas sur le maître est écrite sur le maître.

Une fois la resynchronisation **TWOWAYUPDATE** effectuée, le maître et l'esclave comportent des lignes identiques.

- **MINIMAL** : Cette option fonctionne de la même manière que UPDATE. En outre, les événements (c'est-à-dire les lignes de la table alerts.status) qui figurent sur l'esclave et non sur le maître sont marqués pour suppression. Pour marquer ces événements à des fins de suppression, la passerelle se comporte comme suit :
 1. Pour chaque ligne de la table alerts.status sur l'ObjectServer esclave qui ne figure pas sur le maître, la zone OldRow est paramétrée sur 1.
 2. Le déclencheur pass_deletes s'exécute sur l'ObjectServer esclave et supprime toutes les lignes dont la zone OldRow est paramétrée sur 1.

L'avantage d'une resynchronisation **MINIMAL** est que les ObjectServers maître et esclave sont totalement synchronisés mais moins de données sont envoyées pendant le processus de resynchronisation. La resynchronisation **MINIMAL** consomme moins de données car les lignes ne sont pas supprimées puis recopiées, contrairement à ce qui se passe avec une resynchronisation **NORMAL**.

Propriétés des passerelles bidirectionnelles

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

Tableau 51. Propriétés et options de ligne de commande utilisées par les passerelles bidirectionnelles

Nom de propriété	Option de ligne de commande	Description
Gate.ObjectServerA. BufferSize entier Gate.ObjectServerB. BufferSize entier	-objectserverabufsize entier -objectserverbbufsize entier	Utilisez ces propriétés pour indiquer le nombre d'entrées que la passerelle stocke dans la mémoire tampon pour cet ObjectServer avant de les effacer, si la mise en mémoire tampon est activée. La passerelle vide la mémoire tampon à la fin d'un lot d'instructions SQL, quelle que soit la taille de la mémoire tampon. Elle permet d'affiner l'efficacité de la passerelle. La valeur par défaut est 25.
Gate.ObjectServerA. CommonNames chaîne Gate.ObjectServerB. CommonNames chaîne	-objectservera commonnames chaîne -objectserverb commonnames chaîne	Si la passerelle se connecte à un ObjectServer par SSL et que la zone Nom commun du certificat reçu ne correspond pas au nom spécifié par la propriété Gate.ObjectServerA.Server ou Gate.ObjectServerB.Server (par exemple dans une paire de reprise en ligne ou dans le paramètre d'un serveur virtuel), utilisez ces propriétés pour spécifier une liste de Noms communs SSL acceptables séparés par des virgules. Le paramètre par défaut est d'utiliser la propriété Gate.ObjectServerA.Server ou Gate.ObjectServerB.Server .
Gate.ObjectServerA. Debug booléen Gate.ObjectServerB. Debug booléen	-objectserveradebug booléen -objectserverbdebug booléen	Ces propriétés permettent de spécifier si la passerelle inclut des messages de débogage pour cet ObjectServer dans le journal de débogage de la passerelle. La valeur par défaut est TRUE.
Gate.ObjectServerA. DeleteIfNoDup booléen Gate.ObjectServerB. DeleteIfNoDup booléen	-objectservera delifnodedup booléen -objectserverb delifnodedup booléen	Ces propriétés permettent d'indiquer comment la passerelle transmet les suppressions. Vous disposez des options suivantes : <ul style="list-style-type: none"> • FALSE : La suppression est toujours effectuée. • TRUE : La suppression n'est pas effectuée, à condition que cet événement dans le serveur cible indique qu'il s'est produit à nouveau depuis l'émission de la suppression. La valeur par défaut est FALSE.
Gate.ObjectServerA. Description chaîne Gate.ObjectServerB. Description chaîne	-objectservera description chaîne -objectserverb description chaîne	Cette propriété permet de spécifier la description d'une application pour la connexion à l'ObjectServer A. Cette description sert dans les déclencheurs et vous permet de déterminer quel composant de la passerelle a tenté de réaliser une action. La valeur par défaut est "Gateway Reader/Writer". Dans une configuration de passerelle bidirectionnelle ObjectServer, où la passerelle relie un serveur ObjectServer principal et un serveur ObjectServer de sauvegarde, définissez cette propriété sur failover_gate.
Gate.ObjectServerA. DetailsTableName chaîne Gate.ObjectServerB. DetailsTableName chaîne	-objectservera detailstblname chaîne -objectserverb detailstblname chaîne	Ces propriétés permettent de spécifier le nom de la table des détails lue par la passerelle. La valeur par défaut est alerts.details.
Gate.ObjectServerA. FailbackEnabled booléen Gate.ObjectServerB. FailbackEnabled booléen	-objectservera failbackenabled booléen -objectserverb failbackenabled booléen	Ces propriétés permettent de spécifier la reprise par restauration de l'ObjectServer. La valeur par défaut est FALSE.

Tableau 51. Propriétés et options de ligne de commande utilisées par les passerelles bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.ObjectServerA.FailbackTimeout entier Gate.ObjectServerB.FailbackTimeout entier	-objectservera failbacktimeoutentier -objectserverb failbacktimeoutentier	Ces propriétés permettent de spécifier la période de temps (en secondes) que la passerelle doit attendre avant de vérifier si le serveur ObjectServer maître est une sauvegarde ; ainsi, la passerelle peut basculer en cas de défaillance sur le serveur ObjectServer maître. La valeur par défaut est 30.
Gate.ObjectServerA.IDUCFlushRate entier Gate.ObjectServerB.IDUCFlushRate entier	-objectservera iducflushrate entier -objectserverb iducflushrate entier	Ces propriétés permettent de contrôler la fréquence à laquelle la passerelle recherche des modifications de l'ObjectServer. La valeur par défaut est 0. Si vous conservez la valeur 0 pour la propriété, l'ObjectServer peut notifier à la passerelle que des modifications sont en attente. Si vous avez besoin que la passerelle capture des modifications plus détaillées des événements, paramétrez la propriété sur une valeur différente. Si vous affectez à la propriété une valeur différente de 0, il se peut que la charge sur l'ObjectServer augmente.
Gate.ObjectServerA.JournalTableName chaîne Gate.ObjectServerB.JournalTableName chaîne	-objectservera journaltblname chaîne -objectserverb journaltblname chaîne	Ces propriétés permettent de spécifier le nom de la table des journaux lue par la passerelle. La valeur par défaut est alerts.journal.
Gate.ObjectServerA.LogOSSql booléen Gate.ObjectServerB.LogOSSql booléen	-objectservera logossql booléen -objectserverb logossql booléen	Ces propriétés permettent de spécifier si la passerelle enregistre toutes les commandes SQL envoyées à l'ObjectServer en mode débogage. La valeur par défaut est FALSE.
Gate.ObjectServerA.Password chaîne Gate.ObjectServerB.Password chaîne	-objectservera password chaîne -objectserverb password chaîne	Ces propriétés permettent d'associer un mot de passe à l'utilisateur spécifié par la propriété Gate.ObjectServerA.Username ou Gate.ObjectServerB.Password . La valeur par défaut est "".
Gate.ObjectServerA.ReconnectTimeout entier Gate.ObjectServerB.ReconnectTimeout entier	-objectservera reconntimeout entier -objectserverb reconntimeout entier	Ces propriétés permettent de spécifier la durée, en secondes, entre deux tentatives de rétablissement de la connexion si la connexion à l'ObjectServer est perdue. La valeur par défaut est 30.
Gate.ObjectServerA.RefreshCacheOnUpdate booléen Gate.ObjectServerB.RefreshCacheOnUpdate booléen	-objectservera refcacheonupd booléen -objectserverb refcacheonupd booléen	Ces propriétés permettent de spécifier la manière dont le cache de la table de hachage de l'ObjectServer est actualisé. Vous disposez des options suivantes : <ul style="list-style-type: none"> • TRUE : le cache est resynchronisé avec l'ObjectServer cible avant que chaque collecte de mises à jour des lignes d'une table de la fenêtre IDUC en cours ne soit traitée • FALSE: la passerelle suppose que son cache est exact et par conséquent ne le resynchronise pas La valeur par défaut est FALSE.
Gate.ObjectServerA.SAF booléen Gate.ObjectServerB.SAF booléen	-objectserverasaf booléen -objectserverbsaf booléen	Ces propriétés permettent d'indiquer si la passerelle doit stocker toutes les modifications si l'ObjectServer de destination est indisponible et de les réacheminer lorsque l'ObjectServer redevient disponible. La valeur par défaut est FALSE.

Tableau 51. Propriétés et options de ligne de commande utilisées par les passerelles bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.ObjectServerA.SAFFile chaîne Gate.ObjectServerB.SAFFile chaîne	-objectserverasaffile chaîne -objectserverbsaffile chaîne	Ces propriétés permettent de spécifier le nom du fichier dans lequel la passerelle stocke des modifications alors que l'ObjectServer de destination est indisponible. La valeur par défaut est la suivante : <ul style="list-style-type: none"> • Pour la propriété Gate.ObjectServerA.SAFFile, la valeur par défaut est \$OMNIHOME/var/objserv_bi/NCO_GATE_ObjectServerA.store. • Pour la propriété Gate.ObjectServerB.SAFFile, la valeur par défaut est \$OMNIHOME/var/objserv_bi/NCO_GATE_ObjectServerB.store. Ce fichier n'est utilisé que si la propriété Gate.ObjectServerA.SAF ou Gate.ObjectServerB.SAF est paramétrée sur TRUE.
Gate.ObjectServerA.SAFReplayOnResync booléen Gate.ObjectServerB.SAFReplayOnResync booléen	-objectservera safreplayonresync booléen -objectserverb safreplayonresync booléen	Ces propriétés permettent de spécifier la façon dont le fichier stockage et transmission (SAF) de l'ObjectServerA est réexécuté avant la resynchronisation. <p>Vous disposez des options suivantes :</p> <ul style="list-style-type: none"> • TRUE : le fichier SAF est réexécuté, que la propriété Gate.Resync.Enable ait été paramétrée sur TRUE ou non. • FALSE : le fichier SAF n'est réexécuté que lorsque la propriété Gate.Resync.Enable a été paramétrée sur FALSE La valeur par défaut est FALSE.
Gate.ObjectServerA.Server chaîne Gate.ObjectServerB.Server chaîne	-objectserveraserver chaîne -objectserverbserver chaîne	Ces propriétés permettent de spécifier le nom de l'ObjectServer auquel la passerelle se connecte. <p>La valeur par défaut est NCOMS.</p>
Gate.ObjectServerA.StatusTableName chaîne Gate.ObjectServerB.StatusTableName chaîne	-objectservera statustblname chaîne -objectserverb statustblname chaîne	Ces propriétés permettent de spécifier le nom de la table d'états lue par la passerelle. <p>La valeur par défaut est alerts.status.</p>
Gate.ObjectServerA.TblReplicateDefFile chaîne Gate.ObjectServerB.TblReplicateDefFile chaîne	-objectservera tblrepdefile chaîne -objectserverb tblrepdefile chaîne	Ces propriétés permettent de spécifier le chemin d'accès au fichier de définition de réplication de table. <p>La valeur par défaut est la suivante :</p> <ul style="list-style-type: none"> • Pour la propriété Gate.ObjectServerA.TblReplicateDefFile, la valeur par défaut est \$OMNIHOME/gates/objserv_bi/objserv_bi.objectservera.tblrep.def • Pour la propriété Gate.ObjectServerB.TblReplicateDefFile, la valeur par défaut est \$OMNIHOME/gates/objserv_bi/objserv_bi.objectservera.tblrep.def
Gate.ObjectServerA.UseBulkInsCmd booléen Gate.ObjectServerB.UseBulkInsCmd booléen	-usebulkinscmd booléen -usebulkinscmd booléen	Ces propriétés permettent d'utiliser des insertions par blocs pour accélérer la resynchronisation. <p>Vous disposez des options suivantes :</p> <ul style="list-style-type: none"> • TRUE : la passerelle modifie le format de l'instruction d'insertion qu'elle envoie au ObjectServerA, lui permettant ainsi de traiter les insertions de façon plus efficace. • FALSE : la passerelle n'effectue aucun changement à l'instruction d'insertion avant d'envoyer les événements à l'ObjectServerA. La valeur par défaut est FALSE.

Tableau 51. Propriétés et options de ligne de commande utilisées par les passerelles bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.ObjectServerA.Username chaîne	-objectservera username chaîne	Ces propriétés permettent de spécifier le nom d'utilisateur à utiliser pour authentifier la connexion à l'ObjectServer.
Gate.ObjectServerB.Username chaîne	-objectserverb username chaîne	La valeur par défaut est root.
Gate.Resync.Enable booléen	-resyncenable booléen	Utilisez cette propriété pour que la passerelle resynchronise les ObjectServers une fois que la passerelle établit ou rétablit une connexion aux deux ObjectServers. Pour plus d'informations, voir «Configuration de la resynchronisation», à la page 233. La valeur par défaut est TRUE.
Gate.Resync.LockType chaîne	-resynclocktype chaîne	Cette propriété permet de spécifier les options de verrouillage des serveurs ObjectServer source et de destination pendant la resynchronisation des événements. Vous disposez des options suivantes : <ul style="list-style-type: none"> • FULL : la passerelle verrouille les serveurs ObjectServer source et cible. • PARTIAL : la passerelle ne bloque que l'ObjectServer de destination. • NONE : la passerelle ne bloque ni l'ObjectServer source, ni l'ObjectServer cible. <div style="background-color: #e0e0e0; padding: 2px; display: inline-block;">Fix Pack 1</div> La valeur par défaut est PARTIAL.
Gate.Resync.Master chaîne	-resyncmaster chaîne	Cette propriété permet d'indiquer à la passerelle quel ObjectServer elle doit utiliser comme serveur maître pendant une resynchronisation. Pour utiliser l'ObjectServer qui aura été exécuté le plus longtemps, conservez la propriété sous la forme d'une chaîne vide. Pour utiliser un ObjectServer nommé, quel que soit l'ObjectServer qui aura été exécuté le plus longtemps, paramétrez la propriété sur ObjectServerA ou sur ObjectServerB.
Gate.Resync.Preferred chaîne	-resyncpreferred chaîne	Cette propriété permet d'indiquer à la passerelle quel ObjectServer elle doit utiliser comme serveur maître pendant une resynchronisation si la propriété Gate.Resynch.Master est restée vide et que les deux ObjectServers ont été exécutés pendant la même durée. Pour utiliser un ObjectServer nommé, paramétrez la propriété sur ObjectServerA ou sur ObjectServerB.
Gate.Resync.Type chaîne	-resynctype chaîne	Cette propriété permet d'indiquer comment la passerelle doit resynchroniser les données de table entre les serveurs ObjectServer lorsqu'elle démarre ou restaure une connexion perdue. La passerelle resynchronise les tables définies dans le fichier de définition de réplication de table. Pour plus d'informations sur les valeurs possibles pour cette propriété, voir «Options de Gate.Resync.Type». La valeur par défaut est NORMAL.

Options de Gate.Resync.Type

Vous pouvez définir la propriété **Gate.Resync.Type** sur l'une des valeurs suivantes :

- **NORMAL** : Pour chaque table, la passerelle supprime toutes les données de l'ObjectServer esclave. La passerelle re-transfère ensuite l'ensemble complet des

tables du maître à l'esclave. Avec ce type de resynchronisation, le maître et l'esclave sont totalement synchronisés. Cependant, les lignes de table figurant sur l'esclave et non sur le maître sont perdues. En outre, si des lignes de table figurent sur le maître et l'esclave, la copie de la ligne figurant sur le maître est conservée sur le maître et sur l'esclave. Toutes les mises à jour précédentes de la ligne sur l'esclave sont perdues.

- UPDATE : Pour chaque table, la passerelle génère un cache contenant toutes les lignes des ObjectServer maître et esclave. La passerelle examine ensuite le contenu du cache pour chaque table et compare les données de ligne du maître à celles de l'esclave. Les données sont resynchronisées comme suit :
 - Les lignes de l'esclave qui figurent également sur le maître sont mises à jour avec les données issues du maître, si les données du maître sont différentes de celles de l'esclave.
 - Les lignes figurant sur le maître et non sur l'esclave sont copiées sur l'esclave.
 - Les lignes de l'esclave qui ne figurent pas sur le maître sont conservées.

Avec ce type de resynchronisation, aucun événement n'est perdu, mais il se peut que les serveurs ObjectServer maître et esclave ne soient pas totalement synchronisés.

- MINIMUM : Cette option fonctionne de la même manière que UPDATE. Suite à l'intervention de la passerelle :
 1. Les mémoires caches des événements des ObjectServers maître et esclave sont comparées.
 2. Toute ligne apparaissant sur l'esclave mais pas sur le maître est écrite sur le maître.

Une fois la resynchronisation TWOWAYUPDATE effectuée, le maître et l'esclave comportent des lignes identiques.

- MINIMAL : Cette option fonctionne de la même manière que UPDATE. En outre, les événements (c'est-à-dire les lignes de la table alerts.status) qui figurent sur l'esclave et non sur le maître sont marqués pour suppression. Pour marquer ces événements à des fins de suppression, la passerelle se comporte comme suit :
 1. Pour chaque ligne de la table alerts.status sur l'ObjectServer esclave qui ne figure pas sur le maître, la zone OldRow est paramétrée sur 1.
 2. Le déclencheur pass_deletes s'exécute sur l'ObjectServer esclave et supprime toutes les lignes dont la zone OldRow est paramétrée sur 1.

L'avantage d'une resynchronisation MINIMAL est que les ObjectServers maître et esclave sont totalement synchronisés mais moins de données sont envoyées pendant le processus de resynchronisation. La resynchronisation MINIMAL consomme moins de données car les lignes ne sont pas supprimées puis recopiées, contrairement à ce qui se passe avec une resynchronisation NORMAL.

Exécution des passerelles ObjectServer

Après avoir configuré la passerelle, vous pouvez exécuter la commande permettant de la démarrer.

Procédure

- Pour démarrer une passerelle bidirectionnelle, entrez la commande suivante :
`nco_g_objserv_bi -name passerelle`, où *passerelle* représente le nom de la passerelle.

- Pour démarrer une passerelle unidirectionnelle, entrez la commande suivante : `nco_g_objserv_uni -name passerelle`, où *passerelle* représente le nom de la passerelle.

Mappage de passerelles ObjectServer

La passerelle peut répliquer n'importe quelle table de l'ObjectServer. Pour ce faire, elle mappe des données aux zones appropriées de l'ObjectServer à l'aide d'un fichier de définition de mappe.

Pour répliquer des tables système associées à l'utilisateur, par exemple, SecurityUsers, SecurityGroups, SecurityRoles, SecurityRoleGrants et SecurityGroupMembers, incluez les détails de ces mappages dans le fichier de mappe.

Le chemin du fichier de définition de mappe est déterminé par la propriété **Gate.Mapfile** du fichier de propriétés.

La passerelle lit cette table uniquement au démarrage. Si vous modifiez cette table lors de l'exécution de la passerelle, redémarrez la passerelle de sorte que les modifications prennent effet.

Les fonctions de conversion de fichier de définition de mappe suivantes peuvent être utilisées dans le fichier de définition de mappe :

- TO_STRING (<nom_colonne>)
- TO_INTEGER (<nom_colonne>)
- TO_TIME (<nom_colonne>)

Syntaxe

Les mappages à utiliser avec le programme d'écriture de l'ObjectServer doivent respecter la syntaxe suivante :

```
CREATE MAPPING nom_mappage
(
  'nomzone_dest' = '@nomzone_src' | expression_simple | attribut
  [ ON INSERT ONLY ] [ CONVERT TO type ] [ NOT NULL '@nomzone_src' ],
  ...
) ;
```

Où :

- *nom_mappage* représente le nom du mappage à créer.
- *nom_zone_dest* représente le nom de la zone à écrire dans l'ObjectServer de destination.
- *nomzone_src* représente le nom d'une zone de la table alerts.status de l'ObjectServer.
- *expression_simple* représente une chaîne, de type booléen ou entier, ou une séquence de chaînes ou d'entiers joints à l'aide des opérateurs +, -, *, /. Tous les opérateurs concatènent des chaînes. Tous les opérateurs fonctionnent de gauche à droite lorsqu'ils opèrent sur des entiers.
- *attribut* représente un nom d'attribut.

Les commandes suivantes sont facultatives :

ON INSERT ONLY

Contrôle la mise à jour de la zone pendant la durée de vie de l'alerte. Si cette

commande est omise, la zone est mise à jour lors des changements de l'état de l'alerte. Si elle est incluse, la zone est créée une fois pour l'alerte, mais n'est jamais mise à jour. Utilisez cette commande uniquement lors de la définition des valeurs des variables.

CONVERT TO

Définit une conversion forcée si une zone source ne correspond pas au type de la zone de destination. Les options possibles sont INTEGER, STRING ou DATE.

NOT NULL

Fournit une valeur alternative qui utilisée si la valeur source est zéro ou la chaîne vide. La valeur alternative peut être une colonne ou une constante, et non une expression.

Mappage des attributs

Utilisez des noms d'attributs pour inclure des données supplémentaires dans les définitions de mappage. Vous pouvez indiquer deux types d'attributs : des attributs d'accès aux valeurs du cache ou des attributs dynamiques.

Attributs d'accès aux valeurs du cache

La passerelle des attributs de valeurs du cache pour accéder aux valeurs qui sont stockées dans le cache des références croisées. La table suivante décrit les attributs de valeurs du cache utilisables dans les définitions de mappage.

Tableau 52. Attributs d'accès aux valeurs du cache

Nom d'attribut	Description
STATUS.SERIAL	Numéro de série mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
STATUS.SERVER_SERIAL	Numéro de série du serveur mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
STATUS.SERVER_NAME	Nom du serveur mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
STATUS.IDENTIFIER	Identificateur mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
JOURNAL.SERIAL	Numéro de série de la ligne de la table des journaux mis en cache.
DETAILS.IDENTIFIER	Numéro de série de la ligne de la table des détails mis en cache.

Attributs dynamiques

Les attributs dynamiques permettent à la passerelle d'accéder aux valeurs dynamiques qu'elle génère automatiquement. La table suivante décrit les attributs dynamiques utilisables dans les définitions de mappage.

Tableau 53. Atributs dynamiques

Nom d'attribut	Description
ACTION_CODE	Cet attribut affiche une chaîne d'un seul caractère spécifiant le type d'opération réalisé. Les valeurs admises sont les suivantes : <ul style="list-style-type: none"> • I : Insérer • U : Mettre à jour • D : Supprimer
ACTION_TIME	Cet attribut affiche l'heure UTC à laquelle l'action s'est produite.
DELETEDAT	Cet attribut affiche la date à laquelle la ligne a été effacée, le cas échéant.

Exemple de mappage

Les mappages définissent la façon dont les passerelles répliquent les tables en affectant des données aux zones appropriées dans ObjectServer.

L'exemple suivant présente les mappages à effectuer pour les tables ObjectServer d'écriture de la passerelle. Dans cet exemple, les zones sur le côté gauche du signe égal (=) sont le serveur ObjectServer cible (B) et les zones sur le côté droit du signe égal sont le serveur ObjectServer source (A).

```
CREATE MAPPING StatusMap
(
  'Identifiant' = '@Identifiant' ON INSERT ONLY,
  'Node' = '@Node' ON INSERT ONLY,
  'NodeAlias' = '@NodeAlias' ON INSERT ONLY NOTNULL '@Node',
  'Manager' = '@Manager' ON INSERT ONLY,
  'Agent' = '@Agent' ON INSERT ONLY,
  'AlertGroup' = '@AlertGroup' ON INSERT ONLY,
  'AlertKey' = '@AlertKey' ON INSERT ONLY,
  'Severity' = '@Severity',
  'Summary' = '@Summary',
  'StateChange' = '@StateChange',
  'FirstOccurrence' = '@FirstOccurrence' ON INSERT ONLY,
  'LastOccurrence' = '@LastOccurrence',
  'InternalLast' = '@InternalLast',
  'Poll' = '@Poll' ON INSERT ONLY,
  'Type' = '@Type' ON INSERT ONLY,
  'Tally' = '@Tally',
  'ProbeSubSecondId' = '@ProbeSubSecondId',
  'Class' = '@Class' ON INSERT ONLY,
  'Grade' = '@Grade' ON INSERT ONLY,
  'Location' = '@Location' ON INSERT ONLY,
  'OwnerUID' = '@OwnerUID',
  'OwnerGID' = '@OwnerGID',
  'Acknowledged' = '@Acknowledged',
  'BSM_Identity' = '@BSM_Identity'
  'Flash' = '@Flash',
  'EventId' = '@EventId' ON INSERT ONLY,
  'ExpireTime' = '@ExpireTime' ON INSERT ONLY,
  'ProcessReq' = '@ProcessReq',
  'SuppressEscl' = '@SuppressEscl',
  'Customer' = '@Customer' ON INSERT ONLY,
  'Service' = '@Service' ON INSERT ONLY,
  'PhysicalSlot' = '@PhysicalSlot' ON INSERT ONLY,
  'PhysicalPort' = '@PhysicalPort' ON INSERT ONLY,
  'PhysicalCard' = '@PhysicalCard' ON INSERT ONLY,
  'TaskList' = '@TaskList',
  'NmosSerial' = '@NmosSerial',
```

```

'NmosObjInst' = '@NmosObjInst',
'NmosCauseType' = '@NmosCauseType',
'NmosDomainName' = '@NmosDomainName',
'NmosEntityId' = '@NmosEntityId',
'NmosManagedStatus' = '@NmosManagedStatus',
'NmosEventMap' = '@NmosEventMap',
'LocalNodeAlias' = '@LocalNodeAlias' ON INSERT ONLY,
'LocalPriObj' = '@LocalPriObj' ON INSERT ONLY,
'LocalSecObj' = '@LocalSecObj' ON INSERT ONLY,
'LocalRootObj' = '@LocalRootObj' ON INSERT ONLY,
'RemoteNodeAlias' = '@RemoteNodeAlias' ON INSERT ONLY,
'RemotePriObj' = '@RemotePriObj' ON INSERT ONLY,
'RemoteSecObj' = '@RemoteSecObj' ON INSERT ONLY,
'RemoteRootObj' = '@RemoteRootObj' ON INSERT ONLY,
'X733EventType' = '@X733EventType' ON INSERT ONLY,
'X733ProbableCause' = '@X733ProbableCause' ON INSERT ONLY,
'X733SpecificProb' = '@X733SpecificProb' ON INSERT ONLY,
'X733CorrNotif' = '@X733CorrNotif' ON INSERT ONLY,
'URL' = '@URL' ON INSERT ONLY,
'ExtendedAttr' = '@ExtendedAttr' ON INSERT ONLY,
'ServerName' = '@ServerName' ON INSERT ONLY,
'ServerSerial' = '@ServerSerial' ON INSERT ONLY
);

```

```

CREATE MAPPING JournalMap
(
'KeyField' = TO_STRING(STATUS.SERIAL) + ":" +
TO_STRING('@UID') + ":" +
TO_STRING('@Chrono') ON INSERT ONLY,
'Serial' = STATUS.SERIAL,
'Chrono' = '@Chrono',
'UID' = TO_INTEGER('@UID'),
'Text1' = '@Text1',
'Text2' = '@Text2',
'Text3' = '@Text3',
'Text4' = '@Text4',
'Text5' = '@Text5',
'Text6' = '@Text6',
'Text7' = '@Text7',
'Text8' = '@Text8',
'Text9' = '@Text9',
'Text10' = '@Text10',
'Text11' = '@Text11',
'Text12' = '@Text12',
'Text13' = '@Text13',
'Text14' = '@Text14',
'Text15' = '@Text15',
'Text16' = '@Text16'
);

```

```

CREATE MAPPING DetailsMap
(
'KeyField' = '@Identifier' + '####' +
TO_STRING('@Sequence') ON INSERT ONLY,
'Identifier' = '@Identifier',
'AttrVal' = '@AttrVal',
'Sequence' = '@Sequence',
'Name' = '@Name',
'Detail' = '@Detail'
);

```

```

CREATE MAPPING IducMap
(
'ServerName' = '@ServerName' ON INSERT ONLY,

```

```

'AppName' = '@AppName',
'AppDesc' = '@AppDesc'    ON INSERT ONLY,
'ConnectionId' = '@ConnectionId'    ON INSERT ONLY,
'LastIducTime' = '@LastIducTime'
);

# CREER UN MAPPAGE SecurityUsersMap
# (
# 'UserID' = '@UserID'    ON INSERT ONLY,
# 'UserName' = '@UserName',
# 'SystemUser' = '@SystemUser',
# 'FullName' = '@FullName',
# 'Passwd' = '@Passwd',
# 'UsePAM' = '@UsePAM',
# 'Enabled' = '@Enabled'
# );
#
#
# CREATE MAPPING SecurityGroupsMap
# (
# 'GroupID' = '@GroupID' ON INSERT ONLY,
# 'GroupName' = '@GroupName',
# 'SystemGroup' = '@SystemGroup',
# 'Description' = '@Description'
# );
#
#
# CREATE MAPPING SecurityRolesMap
# (
# 'RoleID' = '@RoleID'    ON INSERT ONLY,
# 'RoleName' = '@RoleName',
# 'SystemRole' = '@SystemRole',
# 'Description' = '@Description',
# 'RoleScope' = '@RoleScope'
# );
#
#
# CREATE MAPPING SecurityRoleGrantsMap
# (
# 'GranteeType' = '@GranteeType'    ON INSERT ONLY,
# 'GranteeID' = '@GranteeID' ON INSERT ONLY,
# 'RoleID' = '@RoleID' ON INSERT ONLY
# );
#
#
# CREATE MAPPING SecurityGroupMembersMap
# (
# 'UserID' = '@UserID'    ON INSERT ONLY,
# 'GroupID' = '@GroupID' ON INSERT ONLY,
# 'Compat' = '@Compat'
# );
#
#
# CREATE MAPPING CatalogRestrictionFiltersMap
# (
# 'RestrictionName' = '@RestrictionName'    ON INSERT ONLY,
# 'TableName' = '@TableName',
# 'DatabaseName' = '@DatabaseName',
# 'ConditionText' = '@ConditionText',
# 'CreationText' = '@CreationText'
# );
#
#
# CREATE MAPPING SecurityRestrictionFiltersMap
# (
# 'GranteeType' = '@GranteeType'    ON INSERT ONLY,

```

```

# 'GranteeID' = '@GranteeID' ON INSERT ONLY,
# 'RestrictionName' = '@RestrictionName',
# 'DatabaseName' = '@DatabaseName',
# 'TableName' = '@TableName'
# );
#
#
# CREATE MAPPING SecurityPermissionsMap
# (
# 'ApplicationID' = '@ApplicationID' ON INSERT ONLY,
# 'ObjectType' = '@ObjectType' ON INSERT ONLY,
# 'Object' = '@Object' ON INSERT ONLY,
# 'GranteeType' = '@GranteeType' ON INSERT ONLY,
# 'GranteeID' = '@GranteeID' ON INSERT ONLY,
# 'Allows' = '@Allows',
# 'Denies' = '@Denies',
# 'GrantOptions' = '@GrantOptions'
# );
#
#
# CREATE MAPPING ToolsMenusMap
# (
# 'MenuID' = '@MenuID' ON INSERT ONLY,
# 'Name' = '@Name',
# 'Owner' = '@Owner',
# 'Enabled' = '@Enabled'
# );
#
# # CREATE MAPPING ToolsMenuItemsMap
# (
# 'KeyField' = TO_STRING('@MenuID') + ":" +
# TO_STRING('@MenuItemID')
# ON INSERT ONLY,
# 'MenuID' = '@MenuID' ON INSERT ONLY,
# 'MenuItemID' = '@MenuItemID' ON INSERT ONLY,
# 'Title' = '@Title',
# 'Description' = '@Description',
# 'Enabled' = '@Enabled',
# 'InvokeType' = '@InvokeType',
# 'InvokeID' = '@InvokeID',
# 'Position' = '@Position',
# 'Accelerator' = '@Accelerator'
# );
#
#
# CREATE MAPPING ToolsActionsMap
# (
# 'ActionID' = '@ActionID' ON INSERT ONLY,
# 'Name' = '@Name',
# 'Owner' = '@Owner',
# 'Enabled' = '@Enabled',
# 'Description1' = '@Description1',
# 'Description2' = '@Description2',
# 'Description3' = '@Description3',
# 'Description4' = '@Description4',
# 'HasInternal' = '@HasInternal',
# 'InternalEffect1' = '@InternalEffect1',
# 'InternalEffect2' = '@InternalEffect2',
# 'InternalEffect3' = '@InternalEffect3',
# 'InternalEffect4' = '@InternalEffect4',
# 'InternalForEach' = '@InternalForEach',
# 'HasExternal' = '@HasExternal',
# 'ExternalEffect1' = '@ExternalEffect1',
# 'ExternalEffect2' = '@ExternalEffect2',
# 'ExternalEffect3' = '@ExternalEffect3',
# 'ExternalEffect4' = '@ExternalEffect4',

```

```

# 'ExternalForEach' = '@ExternalForEach',
# 'RedirectOut' = '@RedirectOut',
# 'RedirectErr' = '@RedirectErr',
# 'Platform' = '@Platform',
# 'JournalText1' = '@JournalText1',
# 'JournalText2' = '@JournalText2',
# 'JournalText3' = '@JournalText3',
# 'JournalText4' = '@JournalText4',
# 'JournalForEach' = '@JournalForEach',
# 'HasForcedJournal' = '@HasForcedJournal'
# );
#
#
# CREATE MAPPING ToolsActionAccessMap
# (
# 'ActionAccessID' = '@ActionAccessID' ON INSERT ONLY,
# 'ActionID' = '@ActionID',
# 'GID' = '@GID',
# 'ClassID' = '@ClassID'
# );
#
#
# CREATE MAPPING ToolsMenuDefsMap
# (
# 'Name' = '@Name' ON INSERT ONLY,
# 'DatabaseName' = '@DatabaseName',
# 'TableName' = '@TableName',
# 'ShowField' = '@ShowField',
# 'AssignField' = '@AssignField',
# 'OrderbyField' = '@OrderbyField',
# 'WhereClause' = '@WhereClause'
# );
#
# CREATE MAPPING ToolsPromptDefsMap
# (
# 'Name' = '@Name' ON INSERT ONLY,
# 'Prompt' = '@Prompt',
# 'Default' = '@Default',
# 'Value' = '@Value',
# 'Type' = '@Type'
# );
#
#
# CREATE MAPPING AlertsConversionsMap
# (
# 'KeyField' = '@KeyField' ON INSERT ONLY,
# 'Colname' = '@Colname' ON INSERT ONLY,
# 'Value' = '@Value' ON INSERT ONLY,
# 'Conversion' = '@Conversion'
# );
#
# CREATE MAPPING AlertsColVisualsMap
# (
# 'Colname' = '@Colname' ON INSERT ONLY,
# 'Title' = '@Title',
# 'DefWidth' = '@DefWidth',
# 'MaxWidth' = '@MaxWidth',
# 'TitleJustify' = '@TitleJustify',
# 'DataJustify' = '@DataJustify'
# );
#
#
# CREATE MAPPING AlertsColorsMap
# (
# 'Severity' = '@Severity' ON INSERT ONLY,
# 'AckedRed' = '@AckedRed',
# 'AckedGreen' = '@AckedGreen',

```

```

# 'AckedBlue' = '@AckedBlue',
# 'UnackedRed' = '@UnackedRed',
# 'UnackedGreen' = '@UnackedGreen',
# 'UnackedBlue' = '@UnackedBlue'
# );
#
#
# CREATE MAPPING MasterServergroupsMap
# (
# 'ServerName' = '@ServerName' ON INSERT ONLY,
# 'GroupID' = '@GroupID',
# 'Weight'=@Weight'
# );

```

Autres commandes d'exécution de passerelle

Sur les systèmes d'exploitation UNIX, vous pouvez utiliser l'interface interactive SQL (l'utilitaire **nco_sql**) pour exécuter des commandes sur la passerelle. Outre les commandes SQL ObjectServer, vous pouvez exécuter les commandes d'exécution suivantes sur la passerelle : SHOW PROPS, GET CONFIG et FAILOVER SYNCH. Ces commandes se trouvent dans le fichier de commandes de la passerelle.

Les informations suivantes sont importantes pour l'exécution des commandes SQL sur la passerelle :

- Pour faire appel à l'utilitaire **nco_sql** sur la passerelle, l'option de ligne de commande **-server** doit spécifier le nom de serveur de passerelle indiqué dans le fichier de connexions de données **omni.dat**.
- Si la propriété **Gate.UsePamAuth** est paramétrée sur FALSE, l'utilisateur exécutant le processus de passerelle doit disposer du droit de lecture de la base de données utilisateur. En fonction de la configuration du système d'exploitation, l'utilisateur doit être autorisé à lire **etc/passwd**, **etc/shadow** et **etc/group**.
- Si la propriété **Gate.UsePamAuth** est paramétrée sur TRUE, le service PAM **nco_g_objserv_bi** ou **nco_g_objserv_uni** doit être configuré pour les modules **auth** et **account**.
- Le nom d'utilisateur servant à démarrer l'utilitaire **nco_sql** doit correspondre à un utilisateur UNIX spécifié par la propriété **Gate.UnixAdminGrp**.

Restriction : Vous ne pouvez pas utiliser l'interface interactive pour exécuter des commandes sur la passerelle si votre environnement est en mode FIPS 140-2.

Pour plus d'informations sur l'interface interactive SQL, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Référence associée:

«Propriétés de passerelles ObjectServer génériques», à la page 238
 Certaines propriétés sont partagées par des passerelles ObjectServer unidirectionnelles et bidirectionnelles.

GET CONFIG

Utilisez la commande **GET CONFIG** pour afficher la configuration en cours de la passerelle en répertoriant toutes les propriétés et leurs valeurs.

La commande **GET CONFIG** est identique à la commande **SHOW PROPS**. Elle peut être supprimée des versions plus récentes de la passerelle ObjectServer.

Syntaxe

```
GET CONFIG;
```

Exemple

```
GET CONFIG;  
go
```

SHOW PROPS

La commande **SHOW PROPS** vous permet d'afficher la configuration actuelle de la passerelle en répertoriant toutes les propriétés et leurs valeurs.

Syntaxe

```
SHOW PROPS ;
```

Exemple

```
SHOW PROPS;  
go
```

FAILOVER SYNCH

La commande **FAILOVER SYNCH** vous permet de synchroniser les données entre les ObjectServer principal et de sauvegarde. La commande spécifie les tables maîtres transférées durant l'opération de transfert de données.

Pour plus d'informations sur la reprise en ligne de l'ObjectServer, reportez-vous au *Guide d'administration d'IBM Tivoli Netcool/OMNIBus (SC43-0823)*.

Syntaxe

```
FAILOVER_SYNC [ ADD 'TABLENAME' TO | REMOVE 'TABLENAME' FROM  
] WRITERNAME ;
```

Exemple

```
FAILOVER_SYNC ADD 'master.names' TO ObjectServerA;  
FAILOVER_SYNC ADD 'master.groups' TO ObjectServerA;  
FAILOVER_SYNC ADD 'master.members' TO ObjectServerA;  
FAILOVER_SYNC ADD 'master.permissions' TO ObjectServerA;  
FAILOVER_SYNC ADD 'master.profiles' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.actions' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.action_access' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.menus' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.menu_defs' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.menu_items' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.prompt_defs' TO ObjectServerA;  
FAILOVER_SYNC ADD 'alerts.conversions' TO ObjectServerA;  
FAILOVER_SYNC ADD 'alerts.col_visuals' TO ObjectServerA;  
FAILOVER_SYNC ADD 'alerts.colors' TO ObjectServerA;
```

```
FAILOVER_SYNC ADD 'alerts.objclass' TO ObjectServerA;
FAILOVER_SYNC ADD 'alerts.objmenus' TO ObjectServerA;
FAILOVER_SYNC ADD 'alerts.objmenuitems' TO ObjectServerA;
```

SET LOG LEVEL

La commande SET LOG LEVEL peut être utilisée dans le fichier de commandes de démarrage. La commande ne prend pas effet immédiatement ; une alternative consiste à configurer le fichier de propriétés.

Syntaxe

```
SET LOG LEVEL TO FATAL|ERROR|WARNING|INFORMATION|DEBUG;
```

Exemple

```
SET LOG LEVEL TO ERROR ;
go
```

Tâches associées:

«Définition du niveau des messages de débogage», à la page 225

Vous pouvez résoudre les problèmes liés à la passerelle en consultant les messages d'erreur. La passerelle possède une fonction de gestion des erreurs, qui est fournie par la bibliothèque NGTK (Netcool/OMNIBus Gateway Toolkit). Vous pouvez indiquer quels messages sont inclus dans les fichiers de débogage.

Fichier de définition de réplication de table

Les passerelles ObjectServer peuvent répliquer les données de n'importe quelle table entre les ObjectServers. Les détails des tables à répliquer sont stockés dans le fichier de définition de réplication de table. Le fichier de définition de réplication de table définit les tables de l'ObjectServer source que la passerelle ObjectServer réplique dans l'ObjectServer cible. La passerelle lit cette table uniquement au démarrage. Si vous modifiez cette table lors de l'exécution de la passerelle, redémarrez la passerelle de sorte que les modifications prennent effet.

Pour les passerelles unidirectionnelles, la propriété

Gate.Reader.TblReplicateDefFile indique l'emplacement du fichier de définition de réplication de table.

Pour les passerelles bidirectionnelles, les propriétés

Gate.ObjectServerA.TblReplicateDefFile et

Gate.ObjectServerB.TblReplicateDefFile indiquent les emplacements des fichiers.

Syntaxe

La syntaxe du fichier de définition de réplication de table est la suivante :

```
REPLICATE {ALL | INSERTS, UPDATES, DELETES, FT_INSERTS, FT_UPDATES, FT_DELETES}
FROM TABLE table_source
  USING MAP nom_mappe
[FILTER WITH clause_filtre] [INTO table_destination] [ ORDER BY chaîne_ordre]
[ASC | DESC] [WITH NORESYNC]
  [RESYNC DELETES FILTER condition]
[SET UPDTOINS CHECK TO {ENABLED|DISABLED|FORCED}]
[AFTER IDUC DO commande]
[CACHE FILTER condition]
```

Le tableau ci-dessous décrit les variables de la syntaxe.

Variable	Description
<i>table_source</i>	Table à répliquer dans l'ObjectServer cible.
<i>nom_mappe</i>	Nom de la définition de mappe qui définit la table.
<i>clause_filtre</i>	<p>Filtre utilisé par la passerelle pour sélectionner des lignes à répliquer. Par défaut, le filtrage est inclusif, ce qui signifie que le filtre n'envoie que les événements qui correspondent à la définition de filtre.</p> <p>Pour envoyer des événements qui ne correspondent pas à la définition de filtre, faites précéder le signe égal (=) d'un point d'exclamation (!). Par exemple, la clause de filtre suivante envoie tous les événements dont la gravité n'est pas paramétrée sur 5 : FILTER with 'Severity !=5'.</p>
<i>table_destination</i>	Table devant recevoir la table répliquée. Si cette clause est omise, le nom de la table de destination est identique à la valeur de <i>table_source</i> .
<i>chaîne_ordre</i>	Liste des noms de colonne séparés par des virgules. Chaque nom de colonne peut être suivi de ASC ou DESC, indiquant si les valeurs dans la colonne doivent être dans l'ordre ascendant ou descendant.
<i>condition</i>	Condition SQL que la passerelle ajoute à l'instruction SELECT lorsqu'elle limite les entrées de cache extraites par la passerelle durant une actualisation du cache.
<i>nom_propriété</i>	Propriété utilisée par la passerelle pour filtrer les données tabulaires. Seules les lignes qui satisfont au filtre sont répliquées.
<i>valeur_propriété</i>	Argument à utiliser dans le filtre.
<i>table_cible</i>	Nom de la table dans laquelle répliquer les données.
<i>clause_filtre_suppression</i>	Définit le filtre de suppression de resynchronisation que la passerelle émet vers l'ObjectServer cible.

Le tableau ci-dessous décrit les options de la commande REPLICATE.

Option	Description
ALL	Equivalent de INSERTS, UPDATES, DELETES.
WITH NORESYNC	Facultatif : indique les tables à ne pas resynchroniser.

Option	Description
ORDER BY	<p>Facultatif : spécifie l'ordre dans lequel les lignes sont renvoyées à la passerelle par l'ObjectServer. Pour chaque colonne, vous pouvez indiquer la façon dont les lignes sont triées.</p> <ul style="list-style-type: none"> • ASC : indique que les valeurs de la colonne sont triées dans l'ordre ascendant. • DESC : indique que les valeurs de la colonne sont triées dans l'ordre descendant. <p>Si vous n'indiquez aucune de ces options, la valeur ASC est utilisée.</p> <p>Pour définir plusieurs colonnes de tri, spécifiez une liste séparée par des virgules, par exemple : ORDER BY 'Serial DESC, StateChange ACS'</p>
RESYNC DELETE FILTER	<p>Facultatif : définit un filtre de suppression de resynchronisation, qui spécifie les lignes à supprimer avant l'insertion dans la table cible. Ce filtre est utilisé lorsque les lignes des tables cible et source ne sont pas une correspondance exacte.</p>
SET UPDTAINS CHECK TO	<p>Facultatif : vous permet de configurer la fonctionnalité de conversion des mises à jour en insertions.</p> <ul style="list-style-type: none"> • ENABLED : la passerelle exécute des conversions de mise à jour en insertion normales. Si une mise à jour reçue de l'ObjectServer source contient une ligne de table qui n'existe pas dans l'ObjectServer cible, la mise à jour est convertie en insertion de sorte que la table est de nouveau remplie dans l'ObjectServer cible. Il s'agit de la valeur par défaut. • DISABLED : pour chaque mise à jour reçue de l'ObjectServer source, la passerelle envoie une mise à jour à l'ObjectServer de destination. Si la mise à jour contient des lignes inexistantes dans la cible, ces lignes sont supprimées. • FORCED : la passerelle convertit toutes les mises à jour reçues de l'ObjectServer source en insertion sur l'ObjectServer cible. Si la ligne existe déjà dans l'ObjectServer cible, la ligne est dé-dupliquée. Ce comportement est identique au comportement des sondes.
AFTER IDUC DO	<p>Facultatif : spécifie une colonne et une valeur associée, que la passerelle applique à toutes les lignes qui ont été insérées, mises à jour ou supprimées.</p>

Option	Description
CACHE FILTER	<p>Facultatif : réduit la quantité de données extraites par la passerelle durant une actualisation de cache de la passerelle unidirectionnelle. La condition est ajoutée à la fin de l'instruction SELECT qui sert à extraire les entrées de cache.</p> <p>Si vous souhaitez utiliser l'option CACHE FILTER, il doit s'agir de la dernière entrée dans la définition de réplication de table.</p>

Effets de la transmission des suppressions sur la taille de mémoire

L'activation ou la désactivation de la transmission des suppressions peut avoir une incidence sur l'utilisation de la mémoire de la passerelle.

Le tableau ci-dessous décrit les effets de l'activation ou la désactivation de la transmission des suppressions sur la mémoire.

Tableau 54. Effets de la transmission des suppressions sur la mémoire

Statut de la transmission des suppressions	Exemple de commande	Effet sur la mémoire
Activée	REPLICATE INSERTS, UPDATES	Si l'ObjectServer cible n'est pas suffisamment rapide pour suivre le flux des données qui lui sont envoyées, l'utilisation de la mémoire de la passerelle augmente.
Désactivée	REPLICATE ALL	L'associateur de passerelle supprime les détails de suppression et ne les transmet pas au programme d'écriture. Pour empêcher de conserver les entrées des lignes supprimées dans le cache, l'associateur de passerelle intervient au nom du programme d'écriture de la passerelle en supprimant du cache de l'ObjectServer cible les entrées des lignes supprimées. Ce comportement aide à stabiliser l'utilisation de la mémoire.

Exemple de fichier de définition de réplication de table

Le fichier de définition de réplication de table définit la façon dont la passerelle ObjectServer réplique des tables entre les serveurs ObjectServers source et cible. Utilisez cet exemple afin de vous familiariser avec le mode de fonctionnement du fichier.

L'exemple suivant présente un fichier de définition de réplication de table :

```
REPLICATE INSERT, DELETE FROM TABLE 'alerts.status'
  USING MAP 'StatusMap'
  ORDER BY 'Serial ASC'
  FILTER WITH 'Severity!=5'
  SET UPDTOINS CHECK TO FORCED
  AFTER IDUC DO 'Location=\\PASSED BY GW\\'
  CACHE FILTER 'ServerName IN (\\'NCOMBS_P\\',\\'NCOMBS_B\\')';

REPLICATE ALL FROM TABLE 'alerts.journal' USING MAP 'JournalMap';

REPLICATE ALL FROM TABLE 'alerts.details' USING MAP 'DetailsMap';

#####
# REMARQUE : Si la réplication des tables système associées à l'utilisateur
# est obligatoire, annulez le commentaire des définitions de réplication
# ci-dessous. Les mappes associées doivent également voir leur commentaire
# annulé.
#####

# REPLICATE ALL FROM TABLE 'security.users'
# USING MAP 'SecurityUsersMap'
# INTO 'transfer.users';
#
# REPLICATE ALL FROM TABLE 'security.groups'
# USING MAP 'SecurityGroupsMap'
# INTO 'transfer.groups';
#
# REPLICATE ALL FROM TABLE 'security.roles'
# USING MAP 'SecurityRolesMap'
# INTO 'transfer.roles';
#
# REPLICATE ALL FROM TABLE 'security.role_grants'
# USING MAP 'SecurityRoleGrantsMap'
# INTO 'transfer.role_grants';
#
# REPLICATE ALL FROM TABLE 'security.group_members'
# USING MAP 'SecurityGroupMembersMap'
# INTO 'transfer.group_members';
```

Chapitre 12. Mappage de passerelles ObjectServer

La passerelle peut répliquer n'importe quelle table de l'ObjectServer. Pour ce faire, elle mappe des données aux zones appropriées de l'ObjectServer à l'aide d'un fichier de définition de mappe.

Pour répliquer des tables système associées à l'utilisateur, par exemple, SecurityUsers, SecurityGroups, SecurityRoles, SecurityRoleGrants et SecurityGroupMembers, incluez les détails de ces mappages dans le fichier de mappe.

Le chemin du fichier de définition de mappe est déterminé par la propriété **Gate.Mapfile** du fichier de propriétés.

La passerelle lit cette table uniquement au démarrage. Si vous modifiez cette table lors de l'exécution de la passerelle, redémarrez la passerelle de sorte que les modifications prennent effet.

Les fonctions de conversion de fichier de définition de mappe suivantes peuvent être utilisées dans le fichier de définition de mappe :

- TO_STRING (<nom_colonne>)
- TO_INTEGER (<nom_colonne>)
- TO_TIME (<nom_colonne>)

Syntaxe

Les mappages à utiliser avec le programme d'écriture de l'ObjectServer doivent respecter la syntaxe suivante :

```
CREATE MAPPING nom_mappage
(
  'nomzone_dest' = '@nomzone_src' | expression_simple | attribut
  [ ON INSERT ONLY ] [ CONVERT TO type ] [ NOT NULL '@nomzone_src' ],
  ...
) ;
```

Où :

- *nom_mappage* représente le nom du mappage à créer.
- *nom_zone_dest* représente le nom de la zone à écrire dans l'ObjectServer de destination.
- *nomzone_src* représente le nom d'une zone de la table alerts.status de l'ObjectServer.
- *expression_simple* représente une chaîne, de type booléen ou entier, ou une séquence de chaînes ou d'entiers joints à l'aide des opérateurs +, -, *, /. Tous les opérateurs concatènent des chaînes. Tous les opérateurs fonctionnent de gauche à droite lorsqu'ils opèrent sur des entiers.
- *attribut* représente un nom d'attribut.

Les commandes suivantes sont facultatives :

ON INSERT ONLY

Contrôle la mise à jour de la zone pendant la durée de vie de l'alerte. Si cette commande est omise, la zone est mise à jour lors des changements de l'état de

l'alerte. Si elle est incluse, la zone est créée une fois pour l'alerte, mais n'est jamais mise à jour. Utilisez cette commande uniquement lors de la définition des valeurs des variables.

CONVERT TO

Définit une conversion forcée si une zone source ne correspond pas au type de la zone de destination. Les options possibles sont INTEGER, STRING ou DATE.

NOT NULL

Fournit une valeur alternative qui utilisée si la valeur source est zéro ou la chaîne vide. La valeur alternative peut être une colonne ou une constante, et non une expression.

Mappage des attributs

Utilisez des noms d'attributs pour inclure des données supplémentaires dans les définitions de mappage. Vous pouvez indiquer deux types d'attributs : des attributs d'accès aux valeurs du cache ou des attributs dynamiques.

Attributs d'accès aux valeurs du cache

La passerelle des attributs de valeurs du cache pour accéder aux valeurs qui sont stockées dans le cache des références croisées. La table suivante décrit les attributs de valeurs du cache utilisables dans les définitions de mappage.

Tableau 55. Attributs d'accès aux valeurs du cache

Nom d'attribut	Description
STATUS.SERIAL	Numéro de série mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
STATUS.SERVER_SERIAL	Numéro de série du serveur mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
STATUS.SERVER_NAME	Nom du serveur mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
STATUS.IDENTIFIER	Identificateur mis en cache représentant la ligne de la table d'états associée à la ligne de la table des journaux ou de la table des détails en cours.
JOURNAL.SERIAL	Numéro de série de la ligne de la table des journaux mis en cache.
DETAILS.IDENTIFIER	Numéro de série de la ligne de la table des détails mis en cache.

Attributs dynamiques

Les attributs dynamiques permettent à la passerelle d'accéder aux valeurs dynamiques qu'elle génère automatiquement. La table suivante décrit les attributs dynamiques utilisables dans les définitions de mappage.

Tableau 56. Atributs dynamiques

Nom d'attribut	Description
ACTION_CODE	Cet attribut affiche une chaîne d'un seul caractère spécifiant le type d'opération réalisé. Les valeurs admises sont les suivantes : <ul style="list-style-type: none"> • I : Insérer • U : Mettre à jour • D : Supprimer
ACTION_TIME	Cet attribut affiche l'heure UTC à laquelle l'action s'est produite.
DELETEDAT	Cet attribut affiche la date à laquelle la ligne a été effacée, le cas échéant.

Exemple de mappage

Les mappages définissent la façon dont les passerelles répliquent les tables en affectant des données aux zones appropriées dans ObjectServer.

L'exemple suivant présente les mappages à effectuer pour les tables ObjectServer d'écriture de la passerelle. Dans cet exemple, les zones sur le côté gauche du signe égal (=) sont le serveur ObjectServer cible (B) et les zones sur le côté droit du signe égal sont le serveur ObjectServer source (A).

```
CREATE MAPPING StatusMap
(
  'Identifiant' = '@Identifiant' ON INSERT ONLY,
  'Node' = '@Node' ON INSERT ONLY,
  'NodeAlias' = '@NodeAlias' ON INSERT ONLY NOTNULL '@Node',
  'Manager' = '@Manager' ON INSERT ONLY,
  'Agent' = '@Agent' ON INSERT ONLY,
  'AlertGroup' = '@AlertGroup' ON INSERT ONLY,
  'AlertKey' = '@AlertKey' ON INSERT ONLY,
  'Severity' = '@Severity',
  'Summary' = '@Summary',
  'StateChange' = '@StateChange',
  'FirstOccurrence' = '@FirstOccurrence' ON INSERT ONLY,
  'LastOccurrence' = '@LastOccurrence',
  'InternalLast' = '@InternalLast',
  'Poll' = '@Poll' ON INSERT ONLY,
  'Type' = '@Type' ON INSERT ONLY,
  'Tally' = '@Tally',
  'ProbeSubSecondId' = '@ProbeSubSecondId',
  'Class' = '@Class' ON INSERT ONLY,
  'Grade' = '@Grade' ON INSERT ONLY,
  'Location' = '@Location' ON INSERT ONLY,
  'OwnerUID' = '@OwnerUID',
  'OwnerGID' = '@OwnerGID',
  'Acknowledged' = '@Acknowledged',
  'BSM_Identity' = '@BSM_Identity',
  'Flash' = '@Flash',
  'EventId' = '@EventId' ON INSERT ONLY,
  'ExpireTime' = '@ExpireTime' ON INSERT ONLY,
  'ProcessReq' = '@ProcessReq',
  'SuppressEscl' = '@SuppressEscl',
  'Customer' = '@Customer' ON INSERT ONLY,
  'Service' = '@Service' ON INSERT ONLY,
  'PhysicalSlot' = '@PhysicalSlot' ON INSERT ONLY,
  'PhysicalPort' = '@PhysicalPort' ON INSERT ONLY,
  'PhysicalCard' = '@PhysicalCard' ON INSERT ONLY,
  'TaskList' = '@TaskList',
```

```

'NmosSerial' = '@NmosSerial',
'NmosObjInst' = '@NmosObjInst',
'NmosCauseType' = '@NmosCauseType',
'NmosDomainName' = '@NmosDomainName',
'NmosEntityId' = '@NmosEntityId',
'NmosManagedStatus' = '@NmosManagedStatus',
'NmosEventMap' = '@NmosEventMap',
'LocalNodeAlias' = '@LocalNodeAlias' ON INSERT ONLY,
'LocalPriObj' = '@LocalPriObj' ON INSERT ONLY,
'LocalSecObj' = '@LocalSecObj' ON INSERT ONLY,
'LocalRootObj' = '@LocalRootObj' ON INSERT ONLY,
'RemoteNodeAlias' = '@RemoteNodeAlias' ON INSERT ONLY,
'RemotePriObj' = '@RemotePriObj' ON INSERT ONLY,
'RemoteSecObj' = '@RemoteSecObj' ON INSERT ONLY,
'RemoteRootObj' = '@RemoteRootObj' ON INSERT ONLY,
'X733EventType' = '@X733EventType' ON INSERT ONLY,
'X733ProbableCause' = '@X733ProbableCause' ON INSERT ONLY,
'X733SpecificProb' = '@X733SpecificProb' ON INSERT ONLY,
'X733CorrNotif' = '@X733CorrNotif' ON INSERT ONLY,
'URL' = '@URL' ON INSERT ONLY,
'ExtendedAttr' = '@ExtendedAttr' ON INSERT ONLY,
'ServerName' = '@ServerName' ON INSERT ONLY,
'ServerSerial' = '@ServerSerial' ON INSERT ONLY
);

```

```

CREATE MAPPING JournalMap
(
'KeyField' = TO_STRING(STATUS.SERIAL) + ":" +
TO_STRING('@UID') + ":" +
TO_STRING('@Chrono') ON INSERT ONLY,
'Serial' = STATUS.SERIAL,
'Chrono' = '@Chrono',
'UID' = TO_INTEGER('@UID'),
'Text1' = '@Text1',
'Text2' = '@Text2',
'Text3' = '@Text3',
'Text4' = '@Text4',
'Text5' = '@Text5',
'Text6' = '@Text6',
'Text7' = '@Text7',
'Text8' = '@Text8',
'Text9' = '@Text9',
'Text10' = '@Text10',
'Text11' = '@Text11',
'Text12' = '@Text12',
'Text13' = '@Text13',
'Text14' = '@Text14',
'Text15' = '@Text15',
'Text16' = '@Text16'
);

```

```

CREATE MAPPING DetailsMap
(
'KeyField' = '@Identifier' + '####' +
TO_STRING('@Sequence') ON INSERT ONLY,
'Identifier' = '@Identifier',
'AttrVal' = '@AttrVal',
'Sequence' = '@Sequence',
'Name' = '@Name',
'Detail' = '@Detail'
);

```

```

CREATE MAPPING IducMap
(

```

```

'ServerName' = '@ServerName'    ON INSERT ONLY,
'AppName' = '@AppName',
'AppDesc' = '@AppDesc'    ON INSERT ONLY,
'ConnectionId' = '@ConnectionId'    ON INSERT ONLY,
'LastIducTime' = '@LastIducTime'
);

# CREER UN MAPPAGE SecurityUsersMap
# (
# 'UserID' = '@UserID'    ON INSERT ONLY,
# 'UserName' = '@UserName',
# 'SystemUser' = '@SystemUser',
# 'FullName' = '@FullName',
# 'Passwd' = '@Passwd',
# 'UsePAM' = '@UsePAM',
# 'Enabled' = '@Enabled'
# );
#
#
# CREATE MAPPING SecurityGroupsMap
# (
# 'GroupID' = '@GroupID'    ON INSERT ONLY,
# 'GroupName' = '@GroupName',
# 'SystemGroup' = '@SystemGroup',
# 'Description' = '@Description'
# );
#
#
# CREATE MAPPING SecurityRolesMap
# (
# 'RoleID' = '@RoleID'    ON INSERT ONLY,
# 'RoleName' = '@RoleName',
# 'SystemRole' = '@SystemRole',
# 'Description' = '@Description',
# 'RoleScope' = '@RoleScope'
# );
#
#
# CREATE MAPPING SecurityRoleGrantsMap
# (
# 'GranteeType' = '@GranteeType'    ON INSERT ONLY,
# 'GranteeID' = '@GranteeID'    ON INSERT ONLY,
# 'RoleID' = '@RoleID'    ON INSERT ONLY
# );
#
#
# CREATE MAPPING SecurityGroupMembersMap
# (
# 'UserID' = '@UserID'    ON INSERT ONLY,
# 'GroupID' = '@GroupID'    ON INSERT ONLY,
# 'Compat' = '@Compat'
# );
#
#
# CREATE MAPPING CatalogRestrictionFiltersMap
# (
# 'RestrictionName' = '@RestrictionName'    ON INSERT ONLY,
# 'TableName' = '@TableName',
# 'DatabaseName' = '@DatabaseName',
# 'ConditionText' = '@ConditionText',
# 'CreationText' = '@CreationText'
# );
#
#
# CREATE MAPPING SecurityRestrictionFiltersMap
# (

```

```

# 'GranteeType' = '@GranteeType' ON INSERT ONLY,
# 'GranteeID' = '@GranteeID' ON INSERT ONLY,
# 'RestrictionName' = '@RestrictionName',
# 'DatabaseName' = '@DatabaseName',
# 'TableName' = '@TableName'
# );
#
#
# CREATE MAPPING SecurityPermissionsMap
# (
# 'ApplicationID' = '@ApplicationID' ON INSERT ONLY,
# 'ObjectType' = '@ObjectType' ON INSERT ONLY,
# 'Object' = '@Object' ON INSERT ONLY,
# 'GranteeType' = '@GranteeType' ON INSERT ONLY,
# 'GranteeID' = '@GranteeID' ON INSERT ONLY,
# 'Allows' = '@Allows',
# 'Denies' = '@Denies',
# 'GrantOptions' = '@GrantOptions'
# );
#
#
# CREATE MAPPING ToolsMenusMap
# (
# 'MenuID' = '@MenuID' ON INSERT ONLY,
# 'Name' = '@Name',
# 'Owner' = '@Owner',
# 'Enabled' = '@Enabled'
# );
#
# # CREATE MAPPING ToolsMenuItemsMap
# (
# 'KeyField' = TO_STRING('@MenuID') + ":" +
# TO_STRING('@MenuItemID')
# ON INSERT ONLY,
# 'MenuID' = '@MenuID' ON INSERT ONLY,
# 'MenuItemID' = '@MenuItemID' ON INSERT ONLY,
# 'Title' = '@Title',
# 'Description' = '@Description',
# 'Enabled' = '@Enabled',
# 'InvokeType' = '@InvokeType',
# 'InvokeID' = '@InvokeID',
# 'Position' = '@Position',
# 'Accelerator' = '@Accelerator'
# );
#
#
# CREATE MAPPING ToolsActionsMap
# (
# 'ActionID' = '@ActionID' ON INSERT ONLY,
# 'Name' = '@Name',
# 'Owner' = '@Owner',
# 'Enabled' = '@Enabled',
# 'Description1' = '@Description1',
# 'Description2' = '@Description2',
# 'Description3' = '@Description3',
# 'Description4' = '@Description4',
# 'HasInternal' = '@HasInternal',
# 'InternalEffect1' = '@InternalEffect1',
# 'InternalEffect2' = '@InternalEffect2',
# 'InternalEffect3' = '@InternalEffect3',
# 'InternalEffect4' = '@InternalEffect4',
# 'InternalForEach' = '@InternalForEach',
# 'HasExternal' = '@HasExternal',
# 'ExternalEffect1' = '@ExternalEffect1',
# 'ExternalEffect2' = '@ExternalEffect2',
# 'ExternalEffect3' = '@ExternalEffect3',

```

```

# 'ExternalEffect4' = '@ExternalEffect4',
# 'ExternalForEach' = '@ExternalForEach',
# 'RedirectOut' = '@RedirectOut',
# 'RedirectErr' = '@RedirectErr',
# 'Platform' = '@Platform',
# 'JournalText1' = '@JournalText1',
# 'JournalText2' = '@JournalText2',
# 'JournalText3' = '@JournalText3',
# 'JournalText4' = '@JournalText4',
# 'JournalForEach' = '@JournalForEach',
# 'HasForcedJournal' = '@HasForcedJournal'
# );
#
#
# CREATE MAPPING ToolsActionAccessMap
# (
# 'ActionAccessID' = '@ActionAccessID'    ON INSERT ONLY,
# 'ActionID' = '@ActionID',
# 'GID' = '@GID',
# 'ClassID' = '@ClassID'
# );
#
#
# CREATE MAPPING ToolsMenuDefsMap
# (
# 'Name' = '@Name'    ON INSERT ONLY,
# 'DatabaseName' = '@DatabaseName',
# 'TableName' = '@TableName',
# 'ShowField' = '@ShowField',
# 'AssignField' = '@AssignField',
# 'OrderbyField' = '@OrderbyField',
# 'WhereClause' = '@WhereClause'
# );
#
#
# CREATE MAPPING ToolsPromptDefsMap
# (
# 'Name' = '@Name'    ON INSERT ONLY,
# 'Prompt' = '@Prompt',
# 'Default' = '@Default',
# 'Value' = '@Value',
# 'Type' = '@Type'
# );
#
#
# CREATE MAPPING AlertsConversionsMap
# (
# 'KeyField' = '@KeyField' ON INSERT ONLY,
# 'Colname' = '@Colname' ON INSERT ONLY,
# 'Value' = '@Value' ON INSERT ONLY,
# 'Conversion' = '@Conversion'
# );
#
#
# CREATE MAPPING AlertsColVisualsMap
# (
# 'Colname' = '@Colname' ON INSERT ONLY,
# 'Title' = '@Title',
# 'DefWidth' = '@DefWidth',
# 'MaxWidth' = '@MaxWidth',
# 'TitleJustify' = '@TitleJustify',
# 'DataJustify' = '@DataJustify'
# );
#
#
# CREATE MAPPING AlertsColorsMap
# (
# 'Severity' = '@Severity' ON INSERT ONLY,
# 'AckedRed' = '@AckedRed',

```

```
# 'AckedGreen' = '@AckedGreen',
# 'AckedBlue' = '@AckedBlue',
# 'UnackedRed' = '@UnackedRed',
# 'UnackedGreen' = '@UnackedGreen',
# 'UnackedBlue' = '@UnackedBlue'
# );
#
#
# CREATE MAPPING MasterServergroupsMap
# (
# 'ServerName' = '@ServerName' ON INSERT ONLY,
# 'GroupID' = '@GroupID',
# 'Weight'=@Weight'
# );
```

Chapitre 13. Configuration des propriétés de passerelles ObjectServer

Pour définir l'environnement opérationnel de la passerelle, notamment les détails de connexion et l'emplacement des autres fichiers de configuration, précisez les propriétés dans le fichier de propriétés. Il s'agit d'un fichier texte qui contient un ensemble de propriétés et leurs valeurs correspondantes.

Par défaut, la passerelle ObjectServer lit le fichier de propriétés dans le fichier `$OMNIHOME/etc/NCO_GATE.props`. Vous pouvez utiliser l'option de ligne de commande `-name` pour renommer ce fichier ou l'option `-propsfile` pour demander à la passerelle de lire un autre fichier de propriétés. Des exemples de fichier de propriétés accessibles en écriture sont fournis dans `$NCHOME/omnibus/gates/objserv_uni/objserv_uni.props` pour les passerelles bidirectionnelles et dans `$NCHOME/omnibus/gates/objserv_bi/objserv_bi.props` pour les passerelles bidirectionnelles.

Pour plus d'informations sur les propriétés communes et les propriétés Interprocess Control (IPC), reportez-vous au *guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

La plupart des tâches de configuration sont identiques pour les passerelles unidirectionnelles et les passerelles bidirectionnelles. Cependant, si une tâche de configuration exige la définition de propriétés différentes, selon si la passerelle est unidirectionnelle ou bidirectionnelle, les tâches sont décrites séparément, par exemple la configuration des opérations de reprise par restauration.

Après avoir édité le fichier de propriétés, redémarrez la passerelle pour que les modifications prennent effet.

L'architecture standard à plusieurs niveaux comprend des exemples de configuration, y compris le fichier de propriétés, pour les passerelles de la couche de collecte, de la couche d'agrégation et de la couche d'affichage. Ces configurations se trouvent dans `$NCHOME/omnibus/extensions/multitier/gateway`. Par exemple, `$NCHOME/omnibus/extensions/multitier/gateway/AGG_GATE` contient la configuration d'un serveur ObjectServer de la couche d'agrégation qui peut être déployé lorsque vous exécutez un serveur ObjectServer principal unique sur un serveur ObjectServer de sauvegarde.

Contrôle de la taille de la mémoire cache

La passerelle utilise une mémoire cache pour stocker les informations des tables qui requièrent le transfert d'un ObjectServer vers un autre. Modifiez la taille de la mémoire cache si l'ObjectServer auquel la passerelle est liée gère un grand nombre d'événements.

La principale fonction de la mémoire cache est de faciliter les opérations d'insertion de journal et de table d'informations. Lorsqu'un journal ou une information est acheminé pour l'insertion dans un ObjectServer cible, le programme d'écriture de la passerelle a besoin de connaître la série de l'état correspondant dans l'ObjectServer cible. Ces informations sont contenues dans la mémoire cache. La mémoire cache est également utilisée pour toute autre table spécifiée dans la table de définition de réplification de table.

La mémoire cache augmente les performances en fournissant à la passerelle une vue résumée en mémoire du contenu des ObjectServer auxquels elle est liée. En conséquence, la passerelle n'a pas besoin d'effectuer une requête sur un ObjectServer pour vérifier l'existence d'un événement ou la valeur Serial ou Tally d'un événement. En revanche, elle peut vérifier la mémoire cache de l'ObjectServer.

Procédure

La mémoire cache est implémentée par une table de hachage. Les performances diminuent si le nombre de lignes dans la table d'états d'ObjectServer est considérablement supérieur au nombre d'entrées de la table de hachage.

Pour contrôler la taille de la mémoire cache de table de hachage, modifiez la valeur de la propriété **Gate.CacheHashTblSize**. La valeur par défaut est 5023 éléments ou lignes. Si une table d'états possède un grand nombre de lignes, par exemple, plus de 20000, augmentez le nombre. Pour plus d'efficacité, indiquez une valeur correspondant à un nombre premier.

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Définition du niveau des messages de débogage

Vous pouvez résoudre les problèmes liés à la passerelle en consultant les messages d'erreur. La passerelle possède une fonction de gestion des erreurs, qui est fournie par la bibliothèque NGTK (Netcool/OMNIBus Gateway Toolkit). Vous pouvez indiquer quels messages sont inclus dans les fichiers de débogage.

Procédure

1. Pour spécifier que la bibliothèque NGTK consigne les messages de débogage, paramétrez la propriété **Gate.NGtkDebug** sur TRUE.
2. Pour spécifier les messages de débogage qui sont inclus dans les fichiers de débogage, paramétrez les propriétés suivantes sur TRUE ou FALSE, selon les besoins :
 - **Gate.Mapper.Debug**
 - **Gate.ObjectServerB.Debug**
 - **Gate.ObjectServerA.Debug**

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242
Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Vérifier si la passerelle s'exécute sous le contrôle de processus

La passerelle peut être placée sous le contrôle de Process Agent (PA), auquel cas PA indique comment la passerelle s'exécute. Vous pouvez déterminer si une passerelle est sous le contrôle de PA en vérifiant les propriétés de la passerelle.

Procédure

Pour vérifier si une passerelle est sous le contrôle de PA, procédez comme suit :

- Pour vérifier si la passerelle est activée pour PA, examinez la valeur de la propriété **Gate.PAAware**. Si la propriété a pour valeur 0, cela signifie que la passerelle n'est pas activée pour PA.
- Pour le nom de l'agent PA exécutant la passerelle, vérifiez la valeur de la propriété **Gate.PAAwareName**

Important : Ne modifiez pas ces propriétés dans le fichier de propriétés de la passerelle. Ces propriétés sont tenues à jour automatiquement par le serveur d'agent de processus et fournissent des informations uniquement.

Pour plus d'informations sur le contrôle PA, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Modification de la taille de mémoire tampon

La taille de mémoire tampon contrôle le nombre d'entrées que la passerelle stocke dans sa mémoire tampon avant de les transmettre à l'ObjectServer. La valeur optimale pour la taille de mémoire tampon dépend de la taille moyenne des événements et de la vitesse du réseau. La taille de mémoire tampon par défaut est souvent suffisante. Pour réduire le temps d'attente, vous pouvez ajuster la taille de mémoire tampon.

La passerelle utilise des mémoires tampon séparées pour les serveurs ObjectServer source et de destination.

Procédure

Vous pouvez ajuster la taille de la mémoire tampon, en procédant comme suit :

1. Pour déterminer le paramètre le plus efficace pour votre système, notez la durée des opérations de resynchronisation. Modifiez ensuite les propriétés de resynchronisation, puis relancez une resynchronisation et comparez les chiffres de durée. Renouvelez cette opération, selon vos besoins.
2. Pour modifier la taille de mémoire tampon pour l'ObjectServer source et l'ObjectServer de destination, apportez les modifications appropriées aux propriétés suivantes :

- **Gate.ObjectServerA.BufferSize**
- **Gate.ObjectServerB.BufferSize**

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Tâches associées:

«Configuration de la resynchronisation», à la page 233

Les propriétés de passerelle permettent d'indiquer comment la passerelle est resynchronisée avec ses serveurs ObjectServer cible. Vous pouvez configurer la resynchronisation pour les passerelles ObjectServer bidirectionnelles et unidirectionnelles.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Modification du mécanisme d'authentification

La passerelle prend en charge l'authentification UNIX standard ou l'authentification PAM (Pluggable Authentication Modules). L'authentification UNIX standard est l'authentification par défaut. L'authentification PAM est requis pour exécuter la passerelle en mode FIPS 140-2.

Pour plus d'information sur l'authentification, voir le *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Procédure

Pour modifier le mécanisme d'authentification en PAM, paramétrez la propriété **Gate.UsePamAuth** sur TRUE.

Que faire ensuite

1. Configurez Tivoli Netcool/OMNIbus de sorte qu'il utilise PAM pour l'authentification externe. Les noms de service sont `nco_g_objserv_uni` et `nco_g_objserv_bi`. Les deux comportent le type de module `account`. Pour plus d'informations sur la configuration de PAM, voir le *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.
2. Redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Connexion à des serveurs ObjectServer sécurisés

Lorsqu'un ObjectServer s'exécute en mode sécurisé, la passerelle doit établir sa connexion soit en tant qu'utilisateur ObjectServer connu, soit en tant que superutilisateur. L'utilisateur doit disposer de droits d'accès aux tables à répliquer, ainsi qu'à certaines tables système. La configuration d'ObjectServer par défaut inclut un groupe appelé Gateway, qui dispose des droits d'accès requis.

Pour plus d'informations sur l'exécution de l'ObjectServer en mode sécurisé, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Resynchronisation des données de sécurité avec des serveurs ObjectServer sécurisés

Si vous souhaitez resynchroniser les données de sécurité lorsque les serveurs ObjectServer s'exécutent en mode sécurisé, exécutez la passerelle en tant que superutilisateur. Si vous ne procédez pas ainsi, lorsque vous tentez la resynchronisation, la passerelle quitte et aucune donnée de sécurité n'est transférée vers l'ObjectServer de destination.

Aucune donnée de sécurité n'est transférée car la passerelle supprime les autorisations de destination et ne peut pas insérer de lignes copiées à partir de la table source. Si vous exécutez la passerelle en tant que superutilisateur, ce problème est ignoré car aucune autorisation n'a besoin d'être définie explicitement. Pour plus d'informations sur l'ObjectServer, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Configuration des opérations de reprise par restauration

Utilisez les propriétés de passerelle pour configurer la façon dont un ObjectServer de sauvegarde rebascule sur un ObjectServer principal une fois que ce dernier est restauré. Si la fonction de reprise par restauration est activée, la passerelle se connecte à l'ObjectServer de sauvegarde lorsque la connexion à l'ObjectServer principal est perdue. Lorsque l'ObjectServer principal devient actif, la passerelle s'y reconnecte.

Important : Bien que vous puissiez utiliser des passerelles ObjectServer pour contrôler la reprise par restauration des clients, pour réduire le risque de perte de données, le comportement de reprise par restauration doit être géré par une paire de serveurs ObjectServer de reprise en ligne et non pas par les clients eux-mêmes.

Le risque de perte de données peut avoir lieu si les clients sont repris par restauration sur un serveur ObjectServer principal avant que la synchronisation soit terminée.

Configuration des opérations de reprise par restauration pour les passerelles ObjectServer unidirectionnelles

Deux serveurs ObjectServer peuvent être configurés en tant que paire, un serveur jouant le rôle de serveur principal et l'autre serveur étant le serveur de sauvegarde. Vous pouvez spécifier comment l'ObjectServer de sauvegarde effectue la reprise par restauration sur l'ObjectServer principal.

Procédure

Ces instructions s'appliquent uniquement aux programmes de lecture ou d'écriture de passerelle qui se connectent à une paire ObjectServer virtuelle.

1. Dans l'ObjectServer de sauvegarde, paramétrez la propriété **BackupObjectServer** sur TRUE, puis redémarrez l'ObjectServer, si nécessaire.
2. Dans le fichier de propriétés de la passerelle, définissez les propriétés **Gate.Reader.Failback** et **Gate.Writer.Failback** sur TRUE.
3. Pour déterminer la fréquence à laquelle les parties de programme de lecture et d'écriture de la passerelle interrogent l'ObjectServer principal ayant échoué, définissez les propriétés **Gate.Reader.FailbackTimeout** et/ou **Gate.Writer.FailbackTimeout**.
4. Pour indiquer les paires ObjectServer, définissez les propriétés **Gate.ObjectServerA.Server** et **Gate.ObjectServerB.Server** sur le nom de serveur ObjectServer virtuel.

Résultats

Lorsque l'ObjectServer principal échoue, le programme de lecture et le programme d'écriture basculent sur l'ObjectServer de sauvegarde sans s'arrêter. Lorsque le programme de lecture ou d'écriture détecte qu'il est désormais connecté à un ObjectServer de sauvegarde, il interroge régulièrement le retour de l'ObjectServer principal. Lorsque l'ObjectServer principal est de nouveau détecté, le programme de lecture ou d'écriture rebascule automatiquement sur l'ObjectServer principal.

Exemple

La figure suivante montre un exemple de configuration de reprise par restauration avec passerelle unidirectionnelle.

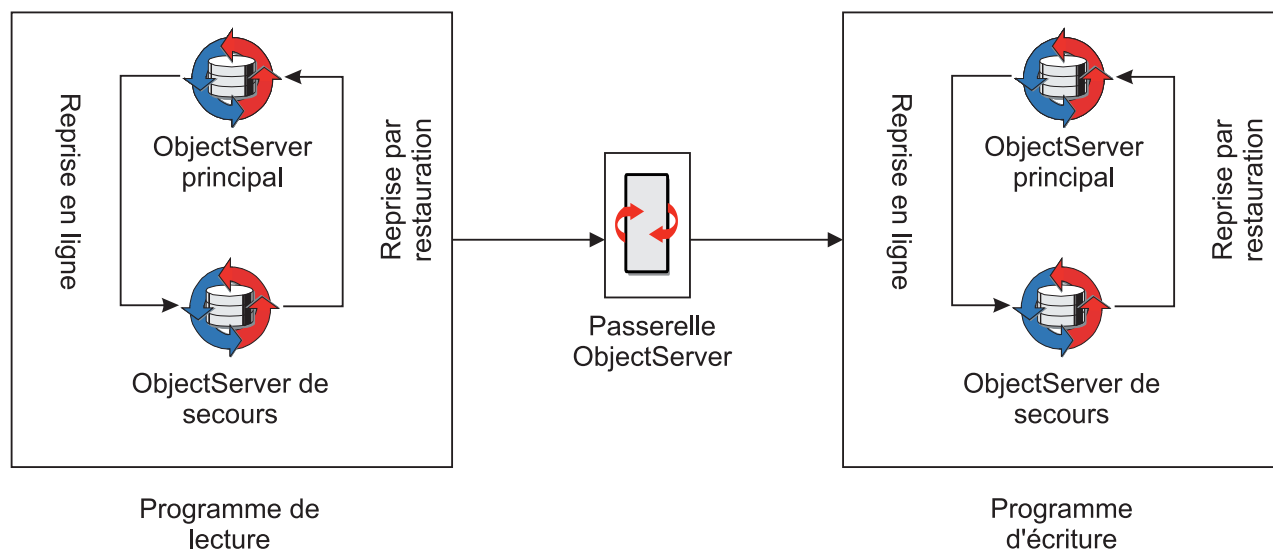


Figure 13. Configuration de la reprise par restauration pour les passerelles ObjectServer unidirectionnelles

Généralement, dans une architecture à plusieurs niveaux, la couche de collecte possède un serveur ObjectServer unique comme programme de lecture et une paire de serveurs ObjectServer comme programme d'écriture. La couche d'agrégation possède une paire de serveurs ObjectServer comme programmes de lecture et d'écriture. La couche d'affichage possède un serveur ObjectServer unique comme programme de lecture et une paire de serveurs ObjectServer comme programme d'écriture.

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Programmes d'écriture et reprise par restauration de la passerelle», à la page 181
 La reprise en ligne se produit lorsqu'une passerelle perd sa connexion avec le serveur ObjectServer principal et se connecte à un serveur ObjectServer de secours. La fonction de reprise par restauration permet à la passerelle de se reconnecter au serveur ObjectServer principal lorsqu'il est à nouveau actif.

Configuration des opérations de reprise par restauration pour les passerelles ObjectServer bidirectionnelles

Si les données sont modifiées sur un seul serveur ObjectServer d'une paire virtuelle, vous pouvez configurer une passerelle bidirectionnelle pour la reprise par restauration. L'opération est identique à celle d'une passerelle unidirectionnelle mais vous devez configurer des propriétés différentes.

Restriction : La reprise par restauration n'a pas lieu lorsque la passerelle bidirectionnelle se connecte à un ObjectServer qui ne se trouve pas dans une paire virtuelle. Par exemple, si vous utilisez une passerelle bidirectionnelle pour gérer un ObjectServer de sauvegarde, la reprise par restauration n'a pas lieu.

Procédure

Pour configurer des opérations de reprise par restauration, procédez comme suit :

1. Dans l'ObjectServer de sauvegarde, paramétrez la propriété **BackupObjectServer** sur TRUE, puis redémarrez l'ObjectServer, si nécessaire.
2. Définissez les propriétés suivantes dans le fichier de propriétés de la passerelle.
 - Si l'ObjectServer A possède un ObjectServer de sauvegarde, paramétrez la propriété **Gate.ObjectServerA.Failback** sur TRUE.
 - Si l'ObjectServer B possède un ObjectServer de sauvegarde, paramétrez la propriété **ObjectServerGate.ObjectServerB.Failback** sur TRUE.
3. Pour spécifier la fréquence à laquelle la passerelle cherche à se reconnecter à l'ObjectServer défaillant, définissez la valeur des propriétés **Gate.ObjectServerA.FailbackTimeout** et **Gate.ObjectServerB.FailbackTimeout**.
4. Pour indiquer les paires ObjectServer, définissez les propriétés **Gate.ObjectServerA.Server** et **Gate.ObjectServerB.Server** sur le nom de serveur ObjectServer virtuel.

Résultats

Lorsque l'ObjectServer principal échoue, la passerelle bascule sur l'ObjectServer de sauvegarde sans s'arrêter. Lorsque la passerelle est connectée à un ObjectServer de sauvegarde, elle cherche régulièrement à se connecter à nouveau à l'ObjectServer principal. Lorsque l'ObjectServer principal est de nouveau détecté, la passerelle rebasculé automatiquement sur l'ObjectServer principal.

Exemple

La figure ci-dessous présente un exemple d'opération de reprise par restauration pour une passerelle bidirectionnelle.

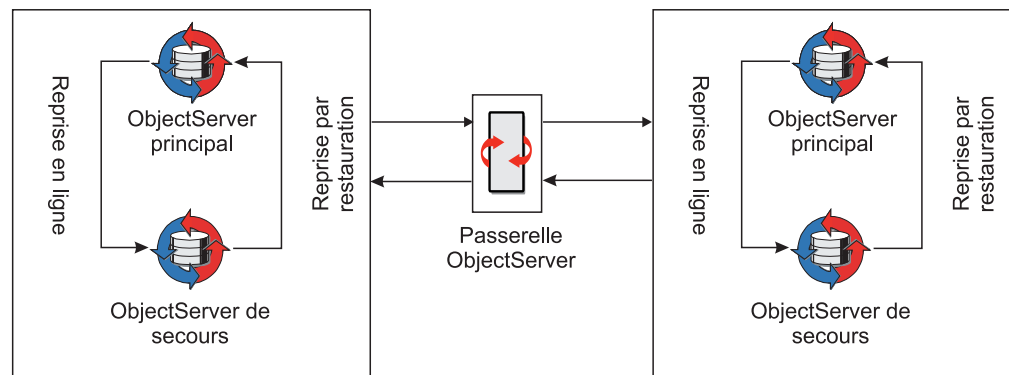


Figure 14. Passerelle ObjectServer bidirectionnelle

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Programmes d'écriture et reprise par restauration de la passerelle», à la page 181
 La reprise en ligne se produit lorsqu'une passerelle perd sa connexion avec le serveur ObjectServer principal et se connecte à un serveur ObjectServer de secours. La fonction de reprise par restauration permet à la passerelle de se reconnecter au serveur ObjectServer principal lorsqu'il est à nouveau actif.

Configuration des opérations de stockage et de retransmission

Vous pouvez configurer les passerelles bidirectionnelles et unidirectionnelles de sorte qu'elles stockent et retransmettent les données des tables répliquées si l'ObjectServer de destination passe hors connexion.

Configuration des opérations de stockage et de retransmission pour les passerelles ObjectServer unidirectionnelles

Les passerelles unidirectionnelles prennent en charge le stockage et la retransmission sur n'importe quelle table lorsque l'ObjectServer cible passe hors connexion. Vous pouvez configurer la passerelle de sorte qu'elle stocke et retransmette toutes ou aucune des tables répliquées.

Procédure

Pour configurer la fonction de stockage et de retransmission :

1. Paramétrez la propriété **Gate.Writer.SAF** sur TRUE.
2. Pour spécifier le fichier dans lequel sont consignées les alertes lorsque l'ObjectServer de destination est hors ligne, paramétrez la propriété **Gate.Writer.SAFFile**.
3. Redémarrez la passerelle.

Que faire ensuite

Pour désactiver la fonction de stockage et de transmission, définissez la propriété **Gate.Writer.SAF** sur FALSE et redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Configuration des opérations de stockage et de retransmission pour les passerelles ObjectServer bidirectionnelles

La passerelle prend en charge le stockage et la retransmission sur n'importe quelle table lorsque l'ObjectServer cible passe hors connexion. Vous pouvez configurer la passerelle de sorte qu'elle stocke et retransmette toutes ou aucune des tables répliquées.

Procédure

1. Pour configurer la fonction de stockage et de retransmission pour l'ObjectServer A :
 - a. Paramétrez la propriété **Gate.ObjectServerA.SAF** sur TRUE.
 - b. Pour spécifier le fichier dans lequel sont consignées les alertes lorsque l'ObjectServer de destination est hors ligne, paramétrez la propriété **Gate.ObjectServerA.SAFFile**.

2. Pour configurer la fonction de stockage et de retransmission pour l'ObjectServer B :
 - a. Paramétrez la propriété **Gate.ObjectServerB.SAF** sur TRUE.
 - b. Pour spécifier le fichier dans lequel sont consignées les alertes lorsque l'ObjectServer de destination est hors ligne, paramétrez la propriété **Gate.ObjectServerB.SAFFile**.
3. Après avoir redéfini les propriétés, redémarrez la passerelle.

Que faire ensuite

Pour désactiver la fonction de stockage et de retransmission, paramétrez la propriété **Gate.ObjectServerA.SAF** sur FALSE et la propriété **Gate.ObjectServerB.SAF** sur FALSE, puis redémarrez la passerelle.

Référence associée:

«Programmes d'écriture et reprise par restauration de la passerelle», à la page 181
La reprise en ligne se produit lorsqu'une passerelle perd sa connexion avec le serveur ObjectServer principal et se connecte à un serveur ObjectServer de secours. La fonction de reprise par restauration permet à la passerelle de se reconnecter au serveur ObjectServer principal lorsqu'il est à nouveau actif.

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Configuration de la resynchronisation

Les propriétés de passerelle permettent d'indiquer comment la passerelle est resynchronisée avec ses serveurs ObjectServer cible. Vous pouvez configurer la resynchronisation pour les passerelles ObjectServer bidirectionnelles et unidirectionnelles.

Tâches associées:

«Modification de la taille de mémoire tampon», à la page 227

La taille de mémoire tampon contrôle le nombre d'entrées que la passerelle stocke dans sa mémoire tampon avant de les transmettre à l'ObjectServer. La valeur optimale pour la taille de mémoire tampon dépend de la taille moyenne des événements et de la vitesse du réseau. La taille de mémoire tampon par défaut est souvent suffisante. Pour réduire le temps d'attente, vous pouvez ajuster la taille de mémoire tampon.

Configuration de la resynchronisation pour les passerelles ObjectServer unidirectionnelles

Pour indiquer comment les passerelles unidirectionnelles sont resynchronisées avec les serveurs ObjectServer, définissez les propriétés de resynchronisation qui se trouvent dans le fichier de propriétés. Les passerelles ObjectServer unidirectionnelles disposent de moins de propriétés de resynchronisation que les passerelles ObjectServer bidirectionnelles.

Procédure

Pour configurer la resynchronisation, définissez les propriétés décrites dans le tableau ci-dessous.

Tableau 57. Propriétés de resynchronisation et options de ligne de commande des Passerelles ObjectServer unidirectionnelles

Nom de propriété	Option de ligne de commande	Description
Gate.Resync.Enable <i>booléen</i>	<i>-resyncenable booléen</i>	Cette propriété permet de spécifier que la passerelle doit utiliser la resynchronisation. La valeur par défaut est TRUE.
Gate.Resync.Type <i>chaîne</i>	<i>-resynctype chaîne</i>	Cette propriété permet d'indiquer comment la passerelle doit resynchroniser les données de table entre les serveurs ObjectServer lorsqu'elle démarre ou restaure une connexion perdue. La passerelle resynchronise les tables définies dans le fichier de définition de réplication de table. Voir les informations situées à la suite de tableau pour obtenir les valeurs sur lesquelles vous pouvez définir cette propriété. La valeur par défaut est NORMAL.
Gate.Resync.LockType <i>chaîne</i>	<i>-resynclocktype chaîne</i>	Cette propriété permet de spécifier les options de verrouillage des serveurs ObjectServer source et de destination pendant la resynchronisation des événements. Choisissez l'une des options suivantes : <ul style="list-style-type: none"> • FULL : la passerelle verrouille les serveurs ObjectServer source et cible. • PARTIAL : la passerelle ne bloque que l'ObjectServer de destination. • NONE : la passerelle ne bloque ni l'ObjectServer source, ni l'ObjectServer cible. <div style="border: 1px solid black; padding: 2px; display: inline-block; margin-top: 5px;">Fix Pack 1</div> La valeur par défaut est PARTIAL.

Vous pouvez définir la propriété **Gate.Resync.Type** sur l'une des valeurs suivantes :

- **NORMAL** : Pour chaque table, la passerelle supprime toutes les données de l'ObjectServer esclave. La passerelle re-transfère ensuite l'ensemble complet des tables du maître à l'esclave. Avec ce type de resynchronisation, le maître et l'esclave sont totalement synchronisés. Cependant, les lignes de table figurant sur l'esclave et non sur le maître sont perdues. En outre, si des lignes de table figurent sur le maître et l'esclave, la copie de la ligne figurant sur le maître est conservée sur le maître et sur l'esclave. Toutes les mises à jour précédentes de la ligne sur l'esclave sont perdues.
- **UPDATE** : Pour chaque table, la passerelle génère un cache contenant toutes les lignes des ObjectServer maître et esclave. La passerelle examine ensuite le contenu du cache pour chaque table et compare les données de ligne du maître à celles de l'esclave. Les données sont resynchronisées comme suit :
 - Les lignes de l'esclave qui figurent également sur le maître sont mises à jour avec les données issues du maître, si les données du maître sont différentes de celles de l'esclave.
 - Les lignes figurant sur le maître et non sur l'esclave sont copiées sur l'esclave.
 - Les lignes de l'esclave qui ne figurent pas sur le maître sont conservées.

Avec ce type de resynchronisation, aucun événement n'est perdu, mais il se peut que les serveurs ObjectServer maître et esclave ne soient pas totalement synchronisés.

- **MINIMUM** : Cette option fonctionne de la même manière que UPDATE. Suite à l'intervention de la passerelle :
 1. Les mémoires caches des événements des ObjectServers maître et esclave sont comparées.

2. Toute ligne apparaissant sur l'esclave mais pas sur le maître est écrite sur le maître.

Une fois la resynchronisation TWOWAYUPDATE effectuée, le maître et l'esclave comportent des lignes identiques.

- **MINIMAL** : Cette option fonctionne de la même manière que UPDATE. En outre, les événements (c'est-à-dire les lignes de la table alerts.status) qui figurent sur l'esclave et non sur le maître sont marqués pour suppression. Pour marquer ces événements à des fins de suppression, la passerelle se comporte comme suit :
 1. Pour chaque ligne de la table alerts.status sur l'ObjectServer esclave qui ne figure pas sur le maître, la zone OldRow est paramétrée sur 1.
 2. Le déclencheur pass_deletes s'exécute sur l'ObjectServer esclave et supprime toutes les lignes dont la zone OldRow est paramétrée sur 1.

L'avantage d'une resynchronisation MINIMAL est que les ObjectServers maître et esclave sont totalement synchronisés mais moins de données sont envoyées pendant le processus de resynchronisation. La resynchronisation MINIMAL consomme moins de données car les lignes ne sont pas supprimées puis recopiées, contrairement à ce qui se passe avec une resynchronisation NORMAL.

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Configuration de la resynchronisation pour les passerelles ObjectServer bidirectionnelles

Pour indiquer comment les passerelles bidirectionnelles sont resynchronisées avec les serveurs ObjectServer, définissez les propriétés de resynchronisation qui se trouvent dans le fichier de propriétés. Les passerelles ObjectServer bidirectionnelles disposent de plus de propriétés de resynchronisation que les passerelles ObjectServer unidirectionnelles.

Procédure

Pour configurer la resynchronisation, définissez les propriétés décrites dans le tableau ci-dessous.

Tableau 58. Propriétés de resynchronisation et options de ligne de commande des Passerelles ObjectServer bidirectionnelles

Nom de propriété	Option de ligne de commande	Description
Gate.Resync.Enable <i>booléen</i>	<code>-resyncenable</code> <i>booléen</i>	Cette propriété permet de spécifier que la passerelle doit utiliser la resynchronisation. La valeur par défaut est TRUE.

Tableau 58. Propriétés de resynchronisation et options de ligne de commande des Passerelles ObjectServer bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Resync.Master chaîne	-resyncmaster chaîne	<p>Cette propriété permet d'indiquer à la passerelle quel ObjectServer elle doit toujours considérer comme étant le maître lors d'une resynchronisation.</p> <p>Les valeurs valides sont ObjectServerA et ObjectServerB.</p> <p>La valeur par défaut est "", soit "aucun maître". Si aucun maître n'est défini, la passerelle exécute une procédure visant à identifier le maître.¹</p>
Gate.Resync.Preferred chaîne	-resyncpreferred chaîne	<p>Cette propriété permet d'indiquer à la passerelle quel ObjectServer elle doit considérer comme étant le maître lors d'une resynchronisation si Gate.Resync.Master est omis, si aucune sauvegarde n'est définie et si le temps de disponibilité est le même pour chaque ObjectServer.¹</p> <p>Les valeurs valides sont ObjectServerA et ObjectServerB.</p> <p>La valeur par défaut est "".</p>
Gate.Resync.Type chaîne	-resynctype chaîne	<p>Cette propriété permet d'indiquer comment la passerelle doit resynchroniser les données de table entre les serveurs ObjectServer lorsqu'elle démarre ou restaure une connexion perdue. La passerelle resynchronise les tables définies dans le fichier de définition de réplication de table.²</p> <p>La valeur par défaut est NORMAL.</p>
Gate.Resync.LockType chaîne	-resynclocktype chaîne	<p>Cette propriété permet de spécifier les options de verrouillage des serveurs ObjectServer source et de destination pendant la resynchronisation des événements.</p> <p>Vous avez ensuite plusieurs options :</p> <ul style="list-style-type: none"> • FULL : la passerelle verrouille les serveurs ObjectServer source et cible. • PARTIAL : la passerelle ne bloque que l'ObjectServer de destination. • NONE : la passerelle ne bloque ni l'ObjectServer source, ni l'ObjectServer cible. <p>Fix Pack 1 La valeur par défaut est PARTIAL.</p>

Tableau 58. Propriétés de resynchronisation et options de ligne de commande des Passerelles ObjectServer bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Notes du tableau :		
<p>1. Si la propriété Gate.Resync.Master n'est pas définie, la passerelle agit comme décrit ci-après pour identifier le maître. Les propriétés listées dans le fichier de propriétés de la passerelle et dans les fichiers de propriétés de chaque ObjectServer sont utilisées.</p> <ol style="list-style-type: none"> La passerelle vérifie la propriété Gate.Resync.Master dans son fichier de propriétés. Si la propriété est définie, l'ObjectServer indiqué pour cette propriété est utilisé. Dans le cas contraire, la passerelle vérifie la propriété BackupObjectServer dans le fichier de propriétés de chaque ObjectServer et opère comme suit : <ul style="list-style-type: none"> Si la propriété prend la valeur TRUE pour l'un des ObjectServers (autrement dit, si celui-ci est défini en tant que serveur de secours), la passerelle vérifie la propriété ActingPrimary de cet ObjectServer. Si la propriété prend la valeur TRUE, l'ObjectServer de secours est utilisé. Si la propriété ne prend la valeur TRUE pour aucun des ObjectServers, la passerelle passe à l'étape 1c, à la page 237. Si aucun ObjectServer de secours n'est défini, ou si la propriété ActingPrimary de l'ObjectServer de secours prend la valeur FALSE, la passerelle vérifie le temps de disponibilité de chaque ObjectServer et opère comme suit : <ul style="list-style-type: none"> L'ObjectServer avec le temps de disponibilité le plus long est utilisé. Si les temps de disponibilité sont identiques, la passerelle vérifie la propriété Gate.Resync.Preferred dans son fichier de propriétés. Si un ObjectServer est spécifié pour la propriété, cet ObjectServer est utilisé. Si la propriété n'est pas définie, l'ObjectServer utilisé est celui qui est spécifié en tant que ObjectServerA dans le fichier de propriétés de la passerelle. <p>2. Vous pouvez indiquer l'une des valeurs suivantes pour la propriété Gate.Resync.Type :</p> <ul style="list-style-type: none"> NORMAL : Pour chaque table, la passerelle supprime toutes les données de l'ObjectServer esclave. La passerelle re-transfère ensuite l'ensemble complet des tables du maître à l'esclave. Avec ce type de resynchronisation, le maître et l'esclave sont totalement synchronisés. Cependant, les lignes de table figurant sur l'esclave et non sur le maître sont perdues. En outre, si des lignes de table figurent sur le maître et l'esclave, la copie de la ligne figurant sur le maître est conservée sur le maître et sur l'esclave. Toutes les mises à jour précédentes de la ligne sur l'esclave sont perdues. UPDATE : Pour chaque table, la passerelle génère un cache contenant toutes les lignes des ObjectServer maître et esclave. La passerelle examine ensuite le contenu du cache pour chaque table et compare les données de ligne du maître à celles de l'esclave. Les données sont resynchronisées comme suit : <ul style="list-style-type: none"> Les lignes de l'esclave qui figurent également sur le maître sont mises à jour avec les données issues du maître, si les données du maître sont différentes de celles de l'esclave. Les lignes figurant sur le maître et non sur l'esclave sont copiées sur l'esclave. Les lignes de l'esclave qui ne figurent pas sur le maître sont conservées. <p>Avec ce type de resynchronisation, aucun événement n'est perdu, mais il se peut que les serveurs ObjectServer maître et esclave ne soient pas totalement synchronisés.</p> MINIMUM : Cette option fonctionne de la même manière que UPDATE. Suite à l'intervention de la passerelle : <ol style="list-style-type: none"> Les mémoires caches des événements des ObjectServers maître et esclave sont comparés. Toute ligne apparaissant sur l'esclave mais pas sur le maître est écrite sur le maître. <p>Une fois la resynchronisation TWOWAYUPDATE effectuée, le maître et l'esclave comportent des lignes identiques.</p> MINIMAL : Cette option fonctionne de la même manière que UPDATE. En outre, les événements (c'est-à-dire les lignes de la table alerts.status) qui figurent sur l'esclave et non sur le maître sont marqués pour suppression. Pour marquer ces événements à des fins de suppression, la passerelle se comporte comme suit : <ol style="list-style-type: none"> Pour chaque ligne de la table alerts.status sur l'ObjectServer esclave qui ne figure pas sur le maître, la zone OldRow est paramétrée sur 1. Le déclencheur pass_deletes s'exécute sur l'ObjectServer esclave et supprime toutes les lignes dont la zone OldRow est paramétrée sur 1. <p>L'avantage d'une resynchronisation MINIMAL est que les ObjectServers maître et esclave sont totalement synchronisés mais moins de données sont envoyées pendant le processus de resynchronisation. La resynchronisation MINIMAL consomme moins de données car les lignes ne sont pas supprimées puis recopiées, contrairement à ce qui se passe avec une resynchronisation NORMAL.</p> 		

Que faire ensuite

Après avoir édité le fichier de propriétés, redémarrez la passerelle.

Référence associée:

«Propriétés des passerelles bidirectionnelles», à la page 249

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

«Propriétés des passerelles unidirectionnelles», à la page 242

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Propriétés de passerelles ObjectServer génériques

Certaines propriétés sont partagées par des passerelles ObjectServer unidirectionnelles et bidirectionnelles.

Pour plus d'informations sur les propriétés de passerelles génériques et les propriétés Interprocess Communication (IPC), voir le *Guide des sondes et des passerelles IBM Netcool/OMNIBus*. Le tableau ci-dessous décrit les propriétés partagées par les passerelles ObjectServer unidirectionnelles et bidirectionnelles.

Tableau 59. Propriétés et options de ligne de commande de passerelles ObjectServer génériques

Nom de propriété	Option de ligne de commande	Description
Gate.CacheHashTblSize entier	-chashtblsize entier	Cette propriété permet de spécifier la taille (en éléments) que la passerelle alloue à la mémoire cache de la table de hachage. La valeur par défaut est 5023.
Gate.MapFile chaîne	-mapfile chaîne	Cette propriété permet de spécifier l'emplacement du fichier de définition de mappe. L'emplacement par défaut est \$OMNIBUS_HOME/gates/objserv_uni/objserv_uni.map pour les passerelles unidirectionnelles et \$OMNIBUS_HOME/gates/objserv_bi/objserv_bi.map pour les passerelles bidirectionnelles.
Gate.StartupCmdFile chaîne	-startupcmdfile chaîne	Cette propriété permet de spécifier l'emplacement du fichier de commandes de démarrage. L'emplacement par défaut est \$OMNIBUS_HOME/objserv_uni/objserv_uni.startup.cmd pour les passerelles unidirectionnelles et \$OMNIBUS_HOME/objserv_bi/objserv_bi.startup.cmd pour les passerelles bidirectionnelles.

Tableau 59. Propriétés et options de ligne de commande de passerelles ObjectServer génériques (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Transfer.FailoverSyncRate entier	-fsyncrate entier	<p>Cette propriété permet de spécifier la durée (en secondes) de la synchronisation de reprise en ligne.</p> <p>La valeur par défaut est 60, c'est-à-dire que toutes les 60 secondes, la passerelle transfère le contenu de toute table définie pour la synchronisation de reprise en ligne depuis le serveur ObjectServer source vers le serveur ObjectServer cible.</p>
Gate.NGtkDebug booléen	-ngtkdebug booléen	<p>Cette propriété permet de spécifier si la bibliothèque NGTK consigne les messages de débogage.</p> <p>La valeur par défaut est TRUE.</p> <p>Conseil : Utilisez les propriétés suivantes pour spécifier les messages de débogage inclus dans le fichier journal de débogage :</p> <ul style="list-style-type: none"> • Gate.Mapper.Debug • Gate.Reader.Debug • Gate.Writer.Debug
Gate.PAAware entier	-paaware entier	<p>Cette propriété est réservée à l'agent de processus (PA) et indique si la passerelle est activée pour PA. Elle est incluse dans le fichier de propriétés pour information uniquement. Ne modifiez pas cette propriété.</p> <p>La valeur par défaut est " ", ce qui signifie que la passerelle n'est pas activée pour PA.</p>

Tableau 59. Propriétés et options de ligne de commande de passerelles ObjectServer génériques (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.PAAwareName chaîne	-paname chaîne	Cette propriété est réservée à l'agent de processus (PA) et indique le nom du PA qui contrôle la passerelle. Elle est incluse dans le fichier de propriétés pour information uniquement. Ne modifiez pas cette propriété. La valeur par défaut est " ", ce qui signifie qu'aucun PA ne contrôle la passerelle.
Gate.UsePamAuth booléen	-usepamauth booléen	Systèmes d'exploitation UNIX uniquement : cette propriété permet de spécifier si l'authentification PAM est utilisée. La valeur par défaut est FALSE. Remarque : Pour exécuter la passerelle en mode FIPS 140-2, paramétrez cette propriété sur TRUE.
Gate.UnixAdminGrp chaîne	-unixadmingroup chaîne	Systèmes d'exploitation UNIX uniquement : cette propriété permet de spécifier le groupe d'administration auquel la passerelle doit appartenir si l'authentification UNIX standard est utilisée. Valeur par défaut : ncoadmin.
MaxLogFileSize entier	-maxlogfilesize entier	Cette propriété permet de spécifier la taille, en ko, que la passerelle attribue au fichier journal. Lorsque le fichier journal atteint cette taille, la passerelle renomme le fichier journal en ajoutant au nom les caractères .old et crée un nouveau fichier journal. La valeur par défaut est 1024.

Tableau 59. Propriétés et options de ligne de commande de passerelles ObjectServer génériques (suite)

Nom de propriété	Option de ligne de commande	Description
OldTimeStamp <i>booléen</i>	<code>-oldtimestamp</code> <i>booléen</i>	<p>Cette propriété permet de spécifier le format d'obsolescence de l'horodatage utilisé par la passerelle dans le fichier journal. Les options sont les suivantes :</p> <ul style="list-style-type: none"> • TRUE : Indique le format d'horodatage spécifique aux paramètres régionaux, qui est utilisé par Tivoli Netcool/OMNIBus V7.2.1 ou version antérieure. Par exemple, <code>jj/MM/AAAA hh:mm:ss AM</code> ou <code>jj/MM/AAAA hh:mm:ss PM</code> lorsque les paramètres régionaux ont pour valeur <code>en_GB</code>. • FALSE : Affiche l'horodatage au format ISO 8601, qui est <code>AAAA-MM-JJThh:mm:ss</code>, où T sépare la date et l'heure, et hh représente l'heure au format 24 heures. <p>La valeur par défaut est FALSE.</p> <p>Important : Ne paramétrez pas la propriété OldTimeStamp sur TRUE lors d'une exécution en mode UTF-8.</p>
N/D	<code>-utf8enabled</code> <i>booléen</i>	
Gate.Mapper.Debug <i>booléen</i>	<code>-mapperdebug</code> <i>booléen</i>	<p>Cette propriété permet de spécifier si la passerelle inclut des messages de débogage de l'associateur dans le journal de débogage.</p> <p>La valeur par défaut est TRUE.</p>
Gate.Mapper.ForwardHistoricDetails <i>booléen</i>	<code>-mapperforhistdtls</code> <i>booléen</i>	<p>Cette propriété permet de spécifier si la passerelle achemine toutes les informations d'historique sur la mise à jour convertie.</p> <p>La valeur par défaut est FALSE.</p>

Tableau 59. Propriétés et options de ligne de commande de passerelles ObjectServer génériques (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Mapper.ForwardHistoricJournals <i>booléen</i>	<code>-mapperforhistjrn1</code> <i>booléen</i>	Cette propriété permet de spécifier si la passerelle achemine tous les journaux historiques sur la mise à jour convertie. La valeur par défaut est FALSE.

Concepts associés:

«Autres commandes d'exécution de passerelle», à la page 262

Sur les systèmes d'exploitation UNIX, vous pouvez utiliser l'interface interactive SQL (l'utilitaire `nco_sql`) pour exécuter des commandes sur la passerelle. Outre les commandes SQL ObjectServer, vous pouvez exécuter les commandes d'exécution suivantes sur la passerelle : SHOW PROPS, GET CONFIG et FAILOVER SYNCH. Ces commandes se trouvent dans le fichier de commandes de la passerelle.

Propriétés des passerelles unidirectionnelles

Outre les propriétés de passerelle ObjectServer génériques, les passerelles unidirectionnelles possèdent des propriétés spécifiques.

Pour plus d'informations sur les propriétés communes et les propriétés Interprocess Communication (IPC), reportez-vous au *Guide des sondes et des passerelles d'IBM Tivoli Netcool/OMNibus SC43-0824*. Le tableau suivant décrit les propriétés de passerelle communes.

Tableau 60. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles

Nom de propriété	Option de ligne de commande	Description
Gate.Reader.CommonNames <i>chaîne</i>	<code>-readercommonnames</code> <i>chaîne</i>	Si la passerelle se connecte à un ObjectServer par SSL et que la zone Nom commun du certificat reçu ne correspond pas au nom spécifié par la propriété Gate.Reader.Server (par exemple dans une paire de reprise en ligne ou dans le paramètre d'un serveur virtuel), utilisez cette propriété pour spécifier une liste séparée par des virgules de Noms communs SSL acceptables. Le paramètre par défaut est d'utiliser la propriété Gate.Reader.Server .
Gate.Reader.Debug <i>booléen</i>	<code>-readerdebug</code> <i>booléen</i>	Cette propriété permet de spécifier si la passerelle inclut des messages de débogage du programme de lecture de la passerelle dans le journal de débogage. La valeur par défaut est TRUE.

Tableau 60. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Reader. Description chaîne	-readerdescription chaîne	Cette propriété permet de spécifier la description de la connexion du programme de lecture. Cette description sert dans les déclencheurs et vous permet de déterminer quel composant de la passerelle a tenté de réaliser une action. La valeur par défaut est "Gateway Reader".
Gate.Reader. DetailsTableName chaîne	-readerdetailstblname chaîne	Cette propriété permet de spécifier le nom de la table des détails lue par la passerelle. La valeur par défaut est alerts.details.
Gate.Reader. FailbackEnabled booléen	-readerfailbackenabled booléen	Cette propriété permet de spécifier la reprise par restauration de l'ObjectServer. La valeur par défaut est TRUE.
Gate.Reader. FailbackTimeout entier	-readerfailbacktimeout entier	Cette propriété permet de spécifier la période de temps (en secondes) que la passerelle doit attendre avant d'entrer en mode de reprise par restauration. La valeur par défaut est 30.
Gate.Reader. IDUCFlushRate entier	-readeriducflushrate entier	Ces propriétés permettent de contrôler la fréquence à laquelle la passerelle recherche des modifications de l'ObjectServer. La valeur par défaut est 0. Si vous conservez la valeur 0 pour la propriété, l'ObjectServer peut notifier à la passerelle que des modifications sont en attente. Si vous avez besoin que la passerelle capture des modifications plus détaillées des événements, paramétrez la propriété sur une valeur différente. Si vous affectez à la propriété une valeur différente de 0, il se peut que la charge sur l'ObjectServer augmente. Pour plus d'informations, consultez la description de la propriété Granularity de l'ObjectServer dans le <i>Guide d'administration d'IBM Tivoli Netcool/OMNIbus</i> .

Tableau 60. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Reader.JournalTableName chaîne	-readerjournaltblname chaîne	Cette propriété permet de spécifier le nom de la table des journaux lue par la passerelle. La valeur par défaut est alerts.journal.
Gate.Reader.LogOSSql booléen	-readerlogossql booléen	Cette propriété permet de spécifier si la passerelle enregistre toutes les commandes SQL envoyées à l'ObjectServer en mode débogage. La valeur par défaut est FALSE.
Gate.Reader.Password chaîne	-readerpassword chaîne	Cette propriété permet d'associer un mot de passe à l'utilisateur spécifié par la propriété Gate.Reader.Username . La valeur par défaut est " ".
Gate.Reader.ReconnectTimeout entier	-readerreconnecttimeout entier	Cette propriété permet de spécifier la durée (en secondes) entre deux tentatives de rétablissement de la connexion si la passerelle perd la connexion à l'ObjectServer. La valeur par défaut est 30.
Gate.Reader.Server chaîne	-readerserver chaîne	Cette propriété permet de spécifier le nom de l'ObjectServer duquel la passerelle peut lire les alertes. La valeur par défaut est NCOMS.
Gate.Reader.StatusTableName chaîne	-readerstatustblname chaîne	Cette propriété permet de spécifier le nom de la table d'états lue par la passerelle. La valeur par défaut est alerts.status.
Gate.Reader.TblReplicateDefFile chaîne	-readertblrepdefeffile chaîne	Cette propriété permet de spécifier le chemin du fichier de définition de réplication de table. La valeur par défaut est \$OMNIHOME/gates/objserv_uni/objserv_uni.reader.tblrep.def.
Gate.Reader.Username chaîne	-readerusername chaîne	Cette propriété permet de spécifier le nom d'utilisateur à utiliser pour authentifier la connexion à ObjectServer. La valeur par défaut est root.

Tableau 60. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Resync.Enable <i>booléen</i>	-resyncenable <i>booléen</i>	Utilisez cette propriété pour que la passerelle resynchronise les ObjectServers une fois que la passerelle établit ou rétablit une connexion aux deux ObjectServers. Pour plus d'informations, voir «Configuration de la resynchronisation», à la page 233. La valeur par défaut est TRUE.
Gate.Resync.LockType <i>chaîne</i>	-resynclocktype <i>chaîne</i>	Cette propriété permet de spécifier les options de verrouillage des serveurs ObjectServer source et cible pendant la resynchronisation des événements. Vous disposez des options suivantes : <ul style="list-style-type: none"> • FULL : la passerelle verrouille les serveurs ObjectServer source et cible. • PARTIAL : la passerelle ne bloque que l'ObjectServer de destination. • NONE : la passerelle ne bloque ni l'ObjectServer source, ni l'ObjectServer cible. <div style="background-color: #e0e0e0; padding: 2px; display: inline-block;">Fix Pack 1</div> La valeur par défaut est PARTIAL.
Gate.Resync.Type <i>chaîne</i>	-resynctype <i>chaîne</i>	Cette propriété permet d'indiquer comment la passerelle doit resynchroniser les données de table entre les serveurs ObjectServer lorsqu'elle démarre ou restaure une connexion perdue. La passerelle resynchronise les tables définies dans le fichier de définition de réplication de table. Pour plus d'informations sur les valeurs possibles pour cette propriété, voir «Options de Gate.Resync.Type», à la page 248. La valeur par défaut est NORMAL.
Gate.Writer.BufferSize <i>entier</i>	-writerbufsize <i>entier</i>	Utilisez ces propriétés pour indiquer le nombre d'entrées que la passerelle stocke dans la mémoire tampon pour cet ObjectServer avant de les effacer, si la mise en mémoire tampon est activée. La passerelle vide la mémoire tampon à la fin d'un lot d'instructions SQL, quelle que soit la taille de la mémoire tampon. Elle permet d'affiner l'efficacité de la passerelle. La valeur par défaut est 25.

Tableau 60. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Writer.CommonNames chaîne	-writercommonnames chaîne	Si la passerelle se connecte à un ObjectServer par SSL et que la zone Nom commun du certificat reçu ne correspond pas au nom spécifié par la propriété Gate.Writer.Server (par exemple dans une paire de reprise en ligne ou dans le paramètre d'un serveur virtuel), utilisez cette propriété pour spécifier une liste séparée par des virgules de Noms communs SSL acceptables. Le paramètre par défaut est d'utiliser la propriété Gate.Writer.Server .
Gate.Writer.Debug booléen	-writerdebug booléen	Cette propriété permet de spécifier si la passerelle inclut des messages de débogage du programme d'écriture de la passerelle dans le journal de débogage. La valeur par défaut est TRUE.
Gate.Writer.Description chaîne	-writerdescription chaîne	Cette propriété permet de spécifier la description de la connexion du programme d'écriture. Cette description sert dans les déclencheurs et vous permet de déterminer quel composant de la passerelle a tenté de réaliser une action. La valeur par défaut est "Gateway Writer".
Gate.Writer.Failback Enabled booléen	-writerfailback enabled booléen	Cette propriété permet de spécifier la reprise par restauration de l'ObjectServer. La valeur par défaut est TRUE.
Gate.Writer.Failback Timeout entier	-writerfailback timeout entier	Cette propriété permet de spécifier la durée (en secondes) avant laquelle la passerelle vérifie si l'ObjectServer principal est revenu et tente de se connecter à nouveau. La valeur par défaut est 30.
Gate.Writer.LogOSSql booléen	-writerlogossql booléen	Cette propriété permet de spécifier si la passerelle enregistre toutes les commandes SQL envoyées à l'ObjectServer en mode débogage. La valeur par défaut est FALSE.

Tableau 60. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Writer.Password <i>chaîne</i>	-writerpassword <i>chaîne</i>	Cette propriété permet de spécifier le mot de passe associé à l'utilisateur défini par la propriété Gate.Writer.Username . La valeur par défaut est " ".
Gate.Writer.ReconnectTimeout <i>entier</i>	-writerreconntimeout <i>entier</i>	Cette propriété permet de spécifier la durée (en secondes) entre deux tentatives de rétablissement de la connexion si la passerelle perd la connexion à l'ObjectServer. La valeur par défaut est 30.
Gate.Writer.RefreshCacheOnUpdate <i>booléen</i>	-writerrefcacheonupd <i>booléen</i>	Cette propriété permet de spécifier si le cache de la table de hachage de l'ObjectServer est actualisé. Vous disposez des options suivantes : <ul style="list-style-type: none"> • TRUE : le cache est resynchronisé avec l'ObjectServer cible avant le traitement de chaque collecte de mises à jour des lignes d'une table de la fenêtre IDUC en cours. • FALSE : la passerelle suppose que son cache est exact et par conséquent ne le resynchronise pas La valeur par défaut est TRUE.
Gate.Writer.SAF <i>booléen</i>	-writersaf <i>booléen</i>	Cette propriété permet de spécifier que la passerelle doit stocker toutes les entrées d'une table lorsque l'ObjectServer de destination est indisponible et ensuite les acheminer lorsque l'ObjectServer redevient disponible. La valeur par défaut est FALSE.
Gate.Writer.SAFFile <i>chaîne</i>	-writersaffile <i>chaîne</i>	Cette propriété permet de spécifier le nom du fichier dont se sert la passerelle pour stocker les entrées d'une table pendant que l'ObjectServer de destination est indisponible. La valeur par défaut est \$OMNIHOME/var/objserv_uni_NCO_GATE_Writer.store. Remarque : Ce fichier n'est utilisé que si la propriété Gate.Writer.SAF est paramétrée sur TRUE.
Gate.Writer.Server <i>chaîne</i>	-writerserver <i>chaîne</i>	Cette propriété permet de spécifier le nom de l'ObjectServer sur lequel la passerelle écrit les alertes. La valeur par défaut est REMOTE.

Tableau 60. Propriétés et options de ligne de commande utilisées par les passerelles unidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Writer.Username <i>chaîne</i>	<code>-writerusername</code> <i>chaîne</i>	Cette propriété permet de spécifier le nom d'utilisateur à utiliser pour authentifier la connexion à ObjectServer. Le nom d'utilisateur permet d'établir la connexion IDUC du programme d'écriture et la connexion secondaire des commandes SQL. La valeur par défaut est root.
Gate.Writer.SAF ReplayOnResync <i>booléen</i>	<code>-writersafreplay onresync</code> <i>booléen</i>	Cette propriété permet de spécifier la façon dont le fichier stockage et transmission (SAF) est réexécuté lors de la resynchronisation. Vous disposez des options suivantes : <ul style="list-style-type: none"> • TRUE : le fichier SAF est réexécuté que la propriété Gate.Resync.Enable ait été paramétrée sur TRUE ou non. • FALSE : le fichier SAF n'est réexécuté que lorsque la propriété Gate.Resync.Enable a été paramétrée sur FALSE. La valeur par défaut est FALSE.
Gate.Writer.UseBulkInsCmd <i>booléen</i>	<code>-usebulkinscmd</code> <i>booléen</i>	Cette propriété permet d'utiliser des insertions par blocs pour accélérer la resynchronisation. Vous disposez des options suivantes : <ul style="list-style-type: none"> • TRUE : la passerelle modifie le format de l'instruction d'insertion qu'elle envoie à l' ObjectServer, lui permettant ainsi de traiter les insertions par blocs de façon plus efficace. • FALSE : la passerelle n'effectue aucun changement à l'instruction d'insertion avant d'envoyer les événements à l'ObjectServer. La valeur par défaut est FALSE.

Options de Gate.Resync.Type

Vous pouvez définir la propriété **Gate.Resync.Type** sur l'une des valeurs suivantes :

- **NORMAL** : Pour chaque table, la passerelle supprime toutes les données de l'ObjectServer esclave. La passerelle re-transfère ensuite l'ensemble complet des tables du maître à l'esclave. Avec ce type de resynchronisation, le maître et l'esclave sont totalement synchronisés. Cependant, les lignes de table figurant sur l'esclave et non sur le maître sont perdues. En outre, si des lignes de table

figurent sur le maître et l'esclave, la copie de la ligne figurant sur le maître est conservée sur le maître et sur l'esclave. Toutes les mises à jour précédentes de la ligne sur l'esclave sont perdues.

- UPDATE : Pour chaque table, la passerelle génère un cache contenant toutes les lignes des ObjectServer maître et esclave. La passerelle examine ensuite le contenu du cache pour chaque table et compare les données de ligne du maître à celles de l'esclave. Les données sont resynchronisées comme suit :
 - Les lignes de l'esclave qui figurent également sur le maître sont mises à jour avec les données issues du maître, si les données du maître sont différentes de celles de l'esclave.
 - Les lignes figurant sur le maître et non sur l'esclave sont copiées sur l'esclave.
 - Les lignes de l'esclave qui ne figurent pas sur le maître sont conservées.

Avec ce type de resynchronisation, aucun événement n'est perdu, mais il se peut que les serveurs ObjectServer maître et esclave ne soient pas totalement synchronisés.

- MINIMUM : Cette option fonctionne de la même manière que UPDATE. Suite à l'intervention de la passerelle :
 1. Les mémoires caches des événements des ObjectServers maître et esclave sont comparées.
 2. Toute ligne apparaissant sur l'esclave mais pas sur le maître est écrite sur le maître.

Une fois la resynchronisation TWOWAYUPDATE effectuée, le maître et l'esclave comportent des lignes identiques.

- MINIMAL : Cette option fonctionne de la même manière que UPDATE. En outre, les événements (c'est-à-dire les lignes de la table alerts.status) qui figurent sur l'esclave et non sur le maître sont marqués pour suppression. Pour marquer ces événements à des fins de suppression, la passerelle se comporte comme suit :
 1. Pour chaque ligne de la table alerts.status sur l'ObjectServer esclave qui ne figure pas sur le maître, la zone OldRow est paramétrée sur 1.
 2. Le déclencheur pass_deletes s'exécute sur l'ObjectServer esclave et supprime toutes les lignes dont la zone OldRow est paramétrée sur 1.

L'avantage d'une resynchronisation MINIMAL est que les ObjectServers maître et esclave sont totalement synchronisés mais moins de données sont envoyées pendant le processus de resynchronisation. La resynchronisation MINIMAL consomme moins de données car les lignes ne sont pas supprimées puis recopiées, contrairement à ce qui se passe avec une resynchronisation NORMAL.

Propriétés des passerelles bidirectionnelles

Outre les propriétés de passerelle ObjectServer génériques, les passerelles bidirectionnelles possèdent des propriétés spécifiques.

Tableau 61. Propriétés et options de ligne de commande utilisées par les passerelles bidirectionnelles

Nom de propriété	Option de ligne de commande	Description
Gate.ObjectServerA. BufferSize entier Gate.ObjectServerB. BufferSize entier	-objectserverabufsize entier -objectserverbbufsize entier	Utilisez ces propriétés pour indiquer le nombre d'entrées que la passerelle stocke dans la mémoire tampon pour cet ObjectServer avant de les effacer, si la mise en mémoire tampon est activée. La passerelle vide la mémoire tampon à la fin d'un lot d'instructions SQL, quelle que soit la taille de la mémoire tampon. Elle permet d'affiner l'efficacité de la passerelle. La valeur par défaut est 25.
Gate.ObjectServerA. CommonNames chaîne Gate.ObjectServerB. CommonNames chaîne	-objectservera commonnames chaîne -objectserverb commonnames chaîne	Si la passerelle se connecte à un ObjectServer par SSL et que la zone Nom commun du certificat reçu ne correspond pas au nom spécifié par la propriété Gate.ObjectServerA.Server ou Gate.ObjectServerB.Server (par exemple dans une paire de reprise en ligne ou dans le paramètre d'un serveur virtuel), utilisez ces propriétés pour spécifier une liste de Noms communs SSL acceptables séparés par des virgules. Le paramètre par défaut est d'utiliser la propriété Gate.ObjectServerA.Server ou Gate.ObjectServerB.Server .
Gate.ObjectServerA. Debug booléen Gate.ObjectServerB. Debug booléen	-objectserveradebug booléen -objectserverbdebug booléen	Ces propriétés permettent de spécifier si la passerelle inclut des messages de débogage pour cet ObjectServer dans le journal de débogage de la passerelle. La valeur par défaut est TRUE.
Gate.ObjectServerA. DeleteIfNoDup booléen Gate.ObjectServerB. DeleteIfNoDup booléen	-objectservera delifnodup booléen -objectserverb delifnodup booléen	Ces propriétés permettent d'indiquer comment la passerelle transmet les suppressions. Vous disposez des options suivantes : <ul style="list-style-type: none"> • FALSE : La suppression est toujours effectuée. • TRUE : La suppression n'est pas effectuée, à condition que cet événement dans le serveur cible indique qu'il s'est produit à nouveau depuis l'émission de la suppression. La valeur par défaut est FALSE.
Gate.ObjectServerA. Description chaîne Gate.ObjectServerB. Description chaîne	-objectservera description chaîne -objectserverb description chaîne	Cette propriété permet de spécifier la description d'une application pour la connexion à l'ObjectServer A. Cette description sert dans les déclencheurs et vous permet de déterminer quel composant de la passerelle a tenté de réaliser une action. La valeur par défaut est "Gateway Reader/Writer". Dans une configuration de passerelle bidirectionnelle ObjectServer, où la passerelle relie un serveur ObjectServer principal et un serveur ObjectServer de sauvegarde, définissez cette propriété sur failover_gate.
Gate.ObjectServerA. DetailsTableName chaîne Gate.ObjectServerB. DetailsTableName chaîne	-objectservera detailstblname chaîne -objectserverb detailstblname chaîne	Ces propriétés permettent de spécifier le nom de la table des détails lue par la passerelle. La valeur par défaut est alerts.details.

Tableau 61. Propriétés et options de ligne de commande utilisées par les passerelles bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.ObjectServerA.FailbackEnabled <i>booléen</i>	-objectservera failbackenabled <i>booléen</i>	Ces propriétés permettent de spécifier la reprise par restauration de l'ObjectServer.
Gate.ObjectServerB.FailbackEnabled <i>booléen</i>	-objectserverb failbackenabled <i>booléen</i>	La valeur par défaut est FALSE.
Gate.ObjectServerA.FailbackTimeout <i>entier</i>	-objectservera failbacktimeoutentier	Ces propriétés permettent de spécifier la période de temps (en secondes) que la passerelle doit attendre avant de vérifier si le serveur ObjectServer maître est une sauvegarde ; ainsi, la passerelle peut basculer en cas de défaillance sur le serveur ObjectServer maître.
Gate.ObjectServerB.FailbackTimeout <i>entier</i>	-objectserverb failbacktimeoutentier	La valeur par défaut est 30.
Gate.ObjectServerA.IDUCFlushRate <i>entier</i>	-objectservera iducflushrate <i>entier</i>	Ces propriétés permettent de contrôler la fréquence à laquelle la passerelle recherche des modifications de l'ObjectServer.
Gate.ObjectServerB.IDUCFlushRate <i>entier</i>	-objectserverb iducflushrate <i>entier</i>	La valeur par défaut est 0. Si vous conservez la valeur 0 pour la propriété, l'ObjectServer peut notifier à la passerelle que des modifications sont en attente. Si vous avez besoin que la passerelle capture des modifications plus détaillées des événements, paramétrez la propriété sur une valeur différente. Si vous affectez à la propriété une valeur différente de 0, il se peut que la charge sur l'ObjectServer augmente.
Gate.ObjectServerA.JournalTableName <i>chaîne</i>	-objectservera journaltblname <i>chaîne</i>	Ces propriétés permettent de spécifier le nom de la table des journaux lue par la passerelle.
Gate.ObjectServerB.JournalTableName <i>chaîne</i>	-objectserverb journaltblname <i>chaîne</i>	La valeur par défaut est alerts.journal.
Gate.ObjectServerA.LogOSSql <i>booléen</i>	-objectservera logossql <i>booléen</i>	Ces propriétés permettent de spécifier si la passerelle enregistre toutes les commandes SQL envoyées à l'ObjectServer en mode débogage.
Gate.ObjectServerB.LogOSSql <i>booléen</i>	-objectserverb logossql <i>booléen</i>	La valeur par défaut est FALSE.
Gate.ObjectServerA.Password <i>chaîne</i>	-objectservera password <i>chaîne</i>	Ces propriétés permettent d'associer un mot de passe à l'utilisateur spécifié par la propriété Gate.ObjectServerA.Username ou Gate.ObjectServerB.Password .
Gate.ObjectServerB.Password <i>chaîne</i>	-objectserverb password <i>chaîne</i>	La valeur par défaut est "".
Gate.ObjectServerA.ReconnectTimeout <i>entier</i>	-objectservera reconnntimeout <i>entier</i>	Ces propriétés permettent de spécifier la durée, en secondes, entre deux tentatives de rétablissement de la connexion si la connexion à l'ObjectServer est perdue.
Gate.ObjectServerB.ReconnectTimeout <i>entier</i>	-objectserverb reconnntimeout <i>entier</i>	La valeur par défaut est 30.
Gate.ObjectServerA.RefreshCacheOnUpdate <i>booléen</i>	-objectservera refcacheonupd <i>booléen</i>	Ces propriétés permettent de spécifier la manière dont le cache de la table de hachage de l'ObjectServer est actualisé.
Gate.ObjectServerB.RefreshCacheOnUpdate <i>booléen</i>	-objectserverb refcacheonupd <i>booléen</i>	Vous disposez des options suivantes : <ul style="list-style-type: none"> • TRUE : le cache est resynchronisé avec l'ObjectServer cible avant que chaque collecte de mises à jour des lignes d'une table de la fenêtre IDUC en cours ne soit traitée • FALSE: la passerelle suppose que son cache est exact et par conséquent ne le resynchronise pas
		La valeur par défaut est FALSE.

Tableau 61. Propriétés et options de ligne de commande utilisées par les passerelles bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.ObjectServerA.SAF <i>booléen</i> Gate.ObjectServerB.SAF <i>booléen</i>	-objectserverasaf <i>booléen</i> -objectserverbsaf <i>booléen</i>	Ces propriétés permettent d'indiquer si la passerelle doit stocker toutes les modifications si l'ObjectServer de destination est indisponible et de les réacheminer lorsque l'ObjectServer redevient disponible. La valeur par défaut est FALSE.
Gate.ObjectServerA.SAFFile <i>chaîne</i> Gate.ObjectServerB.SAFFile <i>chaîne</i>	-objectserverasaffile <i>chaîne</i> -objectserverbsaffile <i>chaîne</i>	Ces propriétés permettent de spécifier le nom du fichier dans lequel la passerelle stocke des modifications alors que l'ObjectServer de destination est indisponible. La valeur par défaut est la suivante : <ul style="list-style-type: none"> Pour la propriété Gate.ObjectServerA.SAFFile, la valeur par défaut est \$OMNIHOME/var/objserv_bi/NCO_GATE_ObjectServerA.store. Pour la propriété Gate.ObjectServerB.SAFFile, la valeur par défaut est \$OMNIHOME/var/objserv_bi/NCO_GATE_ObjectServerB.store. Ce fichier n'est utilisé que si la propriété Gate.ObjectServerA.SAF ou Gate.ObjectServerB.SAF est paramétrée sur TRUE.
Gate.ObjectServerA.SAFReplayOnResync <i>booléen</i> Gate.ObjectServerB.SAFReplayOnResync <i>booléen</i>	-objectservera safreplayonresync <i>booléen</i> -objectserverb safreplayonresync <i>booléen</i>	Ces propriétés permettent de spécifier la façon dont le fichier stockage et transmission (SAF) de l'ObjectServerA est réexécuté avant la resynchronisation. Vous disposez des options suivantes : <ul style="list-style-type: none"> TRUE : le fichier SAF est réexécuté, que la propriété Gate.Resync.Enable ait été paramétrée sur TRUE ou non. FALSE : le fichier SAF n'est réexécuté que lorsque la propriété Gate.Resync.Enable a été paramétrée sur FALSE La valeur par défaut est FALSE.
Gate.ObjectServerA.Server <i>chaîne</i> Gate.ObjectServerB.Server <i>chaîne</i>	-objectserveraserver <i>chaîne</i> -objectserverbserver <i>chaîne</i>	Ces propriétés permettent de spécifier le nom de l'ObjectServer auquel la passerelle se connecte. La valeur par défaut est NCOMS.
Gate.ObjectServerA.StatusTableName <i>chaîne</i> Gate.ObjectServerB.StatusTableName <i>chaîne</i>	-objectservera statustblname <i>chaîne</i> -objectserverb statustblname <i>chaîne</i>	Ces propriétés permettent de spécifier le nom de la table d'états lue par la passerelle. La valeur par défaut est alerts.status.
Gate.ObjectServerA.TblReplicateDefFile <i>chaîne</i> Gate.ObjectServerB.TblReplicateDefFile <i>chaîne</i>	-objectservera tblrepdefile <i>chaîne</i> -objectserverb tblrepdefile <i>chaîne</i>	Ces propriétés permettent de spécifier le chemin d'accès au fichier de définition de réplication de table. La valeur par défaut est la suivante : <ul style="list-style-type: none"> Pour la propriété Gate.ObjectServerA.TblReplicateDefFile, la valeur par défaut est \$OMNIHOME/gates/objserv_bi/objserv_bi.objectservera.tblrep.def Pour la propriété Gate.ObjectServerB.TblReplicateDefFile, la valeur par défaut est \$OMNIHOME/gates/objserv_bi/objserv_bi.objectservera.tblrep.def

Tableau 61. Propriétés et options de ligne de commande utilisées par les passerelles bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.ObjectServerA.UseBulkInsCmd <i>booléen</i> Gate.ObjectServerB.UseBulkInsCmd <i>booléen</i>	-usebulkinscmd <i>booléen</i> -usebulkinscmd <i>booléen</i>	Ces propriétés permettent d'utiliser des insertions par blocs pour accélérer la resynchronisation. Vous disposez des options suivantes : <ul style="list-style-type: none"> • TRUE : la passerelle modifie le format de l'instruction d'insertion qu'elle envoie au ObjectServerA, lui permettant ainsi de traiter les insertions de façon plus efficace. • FALSE : la passerelle n'effectue aucun changement à l'instruction d'insertion avant d'envoyer les événements à l'ObjectServerA. La valeur par défaut est FALSE.
Gate.ObjectServerA.Username <i>chaîne</i> Gate.ObjectServerB.Username <i>chaîne</i>	-objectservera username <i>chaîne</i> -objectserverb username <i>chaîne</i>	Ces propriétés permettent de spécifier le nom d'utilisateur à utiliser pour authentifier la connexion à l'ObjectServer. La valeur par défaut est root.
Gate.Resync.Enable <i>booléen</i>	-resyncenable <i>booléen</i>	Utilisez cette propriété pour que la passerelle resynchronise les ObjectServers une fois que la passerelle établit ou rétablit une connexion aux deux ObjectServers. Pour plus d'informations, voir «Configuration de la resynchronisation», à la page 233. La valeur par défaut est TRUE.
Gate.Resync.LockType <i>chaîne</i>	-resynclocktype <i>chaîne</i>	Cette propriété permet de spécifier les options de verrouillage des serveurs ObjectServer source et de destination pendant la resynchronisation des événements. Vous disposez des options suivantes : <ul style="list-style-type: none"> • FULL : la passerelle verrouille les serveurs ObjectServer source et cible. • PARTIAL : la passerelle ne bloque que l'ObjectServer de destination. • NONE : la passerelle ne bloque ni l'ObjectServer source, ni l'ObjectServer cible. <div style="background-color: #e0e0e0; padding: 2px; display: inline-block;">Fix Pack 1</div> La valeur par défaut est PARTIAL.
Gate.Resync.Master <i>chaîne</i>	-resyncmaster <i>chaîne</i>	Cette propriété permet d'indiquer à la passerelle quel ObjectServer elle doit utiliser comme serveur maître pendant une resynchronisation. Pour utiliser l'ObjectServer qui aura été exécuté le plus longtemps, conservez la propriété sous la forme d'une chaîne vide. Pour utiliser un ObjectServer nommé, quel que soit l'ObjectServer qui aura été exécuté le plus longtemps, paramétrez la propriété sur ObjectServerA ou sur ObjectServerB.
Gate.Resync.Preferred <i>chaîne</i>	-resyncpreferred <i>chaîne</i>	Cette propriété permet d'indiquer à la passerelle quel ObjectServer elle doit utiliser comme serveur maître pendant une resynchronisation si la propriété Gate.Resynch.Master est restée vide et que les deux ObjectServers ont été exécutés pendant la même durée. Pour utiliser un ObjectServer nommé, paramétrez la propriété sur ObjectServerA ou sur ObjectServerB.

Tableau 61. Propriétés et options de ligne de commande utilisées par les passerelles bidirectionnelles (suite)

Nom de propriété	Option de ligne de commande	Description
Gate.Resync.Type chaîne	-resynctype chaîne	<p>Cette propriété permet d'indiquer comment la passerelle doit resynchroniser les données de table entre les serveurs ObjectServer lorsqu'elle démarre ou restaure une connexion perdue. La passerelle resynchronise les tables définies dans le fichier de définition de réplication de table.</p> <p>Pour plus d'informations sur les valeurs possibles pour cette propriété, voir «Options de Gate.Resync.Type», à la page 253.</p> <p>La valeur par défaut est NORMAL.</p>

Options de Gate.Resync.Type

Vous pouvez définir la propriété **Gate.Resync.Type** sur l'une des valeurs suivantes :

- **NORMAL** : Pour chaque table, la passerelle supprime toutes les données de l'ObjectServer esclave. La passerelle re-transfère ensuite l'ensemble complet des tables du maître à l'esclave. Avec ce type de resynchronisation, le maître et l'esclave sont totalement synchronisés. Cependant, les lignes de table figurant sur l'esclave et non sur le maître sont perdues. En outre, si des lignes de table figurent sur le maître et l'esclave, la copie de la ligne figurant sur le maître est conservée sur le maître et sur l'esclave. Toutes les mises à jour précédentes de la ligne sur l'esclave sont perdues.
- **UPDATE** : Pour chaque table, la passerelle génère un cache contenant toutes les lignes des ObjectServer maître et esclave. La passerelle examine ensuite le contenu du cache pour chaque table et compare les données de ligne du maître à celles de l'esclave. Les données sont resynchronisées comme suit :
 - Les lignes de l'esclave qui figurent également sur le maître sont mises à jour avec les données issues du maître, si les données du maître sont différentes de celles de l'esclave.
 - Les lignes figurant sur le maître et non sur l'esclave sont copiées sur l'esclave.
 - Les lignes de l'esclave qui ne figurent pas sur le maître sont conservées.

Avec ce type de resynchronisation, aucun événement n'est perdu, mais il se peut que les serveurs ObjectServer maître et esclave ne soient pas totalement synchronisés.

- **MINIMUM** : Cette option fonctionne de la même manière que UPDATE. Suite à l'intervention de la passerelle :
 1. Les mémoires caches des événements des ObjectServers maître et esclave sont comparées.
 2. Toute ligne apparaissant sur l'esclave mais pas sur le maître est écrite sur le maître.

Une fois la resynchronisation TWOWAYUPDATE effectuée, le maître et l'esclave comportent des lignes identiques.

- **MINIMAL** : Cette option fonctionne de la même manière que UPDATE. En outre, les événements (c'est-à-dire les lignes de la table alerts.status) qui figurent sur l'esclave et non sur le maître sont marqués pour suppression. Pour marquer ces événements à des fins de suppression, la passerelle se comporte comme suit :
 1. Pour chaque ligne de la table alerts.status sur l'ObjectServer esclave qui ne figure pas sur le maître, la zone OldRow est paramétrée sur 1.

2. Le déclencheur `pass_deletes` s'exécute sur l'ObjectServer esclave et supprime toutes les lignes dont la zone `OldRow` est paramétrée sur 1.

L'avantage d'une resynchronisation `MINIMAL` est que les ObjectServers maître et esclave sont totalement synchronisés mais moins de données sont envoyées pendant le processus de resynchronisation. La resynchronisation `MINIMAL` consomme moins de données car les lignes ne sont pas supprimées puis recopiées, contrairement à ce qui se passe avec une resynchronisation `NORMAL`.

Chapitre 14. Autres commandes d'exécution de passerelle

Sur les systèmes d'exploitation UNIX, vous pouvez utiliser l'interface interactive SQL (l'utilitaire **nco_sql**) pour exécuter des commandes sur la passerelle. Outre les commandes SQL ObjectServer, vous pouvez exécuter les commandes d'exécution suivantes sur la passerelle : SHOW PROPS, GET CONFIG et FAILOVER SYNCH. Ces commandes se trouvent dans le fichier de commandes de la passerelle.

Les informations suivantes sont importantes pour l'exécution des commandes SQL sur la passerelle :

- Pour faire appel à l'utilitaire **nco_sql** sur la passerelle, l'option de ligne de commande **-server** doit spécifier le nom de serveur de passerelle indiqué dans le fichier de connexions de données **omni.dat**.
- Si la propriété **Gate.UsePamAuth** est paramétrée sur **FALSE**, l'utilisateur exécutant le processus de passerelle doit disposer du droit de lecture de la base de données utilisateur. En fonction de la configuration du système d'exploitation, l'utilisateur doit être autorisé à lire **etc/passwd**, **etc/shadow** et **etc/group**.
- Si la propriété **Gate.UsePamAuth** est paramétrée sur **TRUE**, le service PAM **nco_g_objserv_bi** ou **nco_g_objserv_uni** doit être configuré pour les modules **auth** et **account**.
- Le nom d'utilisateur servant à démarrer l'utilitaire **nco_sql** doit correspondre à un utilisateur UNIX spécifié par la propriété **Gate.UnixAdminGrp**.

Restriction : Vous ne pouvez pas utiliser l'interface interactive pour exécuter des commandes sur la passerelle si votre environnement est en mode FIPS 140-2.

Pour plus d'informations sur l'interface interactive SQL, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIBus*.

Référence associée:

«Propriétés de passerelles ObjectServer génériques», à la page 238
Certaines propriétés sont partagées par des passerelles ObjectServer unidirectionnelles et bidirectionnelles.

GET CONFIG

Utilisez la commande **GET CONFIG** pour afficher la configuration en cours de la passerelle en répertoriant toutes les propriétés et leurs valeurs.

La commande **GET CONFIG** est identique à la commande **SHOW PROPS**. Elle peut être supprimée des versions plus récentes de la passerelle ObjectServer.

Syntaxe

```
GET CONFIG;
```

Exemple

```
GET CONFIG;  
go
```

SHOW PROPS

La commande **SHOW PROPS** vous permet d'afficher la configuration actuelle de la passerelle en répertoriant toutes les propriétés et leurs valeurs.

Syntaxe

```
SHOW PROPS ;
```

Exemple

```
SHOW PROPS;  
go
```

FAILOVER SYNCH

La commande **FAILOVER SYNCH** vous permet de synchroniser les données entre les ObjectServer principal et de sauvegarde. La commande spécifie les tables maîtres transférées durant l'opération de transfert de données.

Pour plus d'informations sur la reprise en ligne de l'ObjectServer, reportez-vous au *Guide d'administration d'IBM Tivoli Netcool/OMNIBus (SC43-0823)*.

Syntaxe

```
FAILOVER_SYNC [ ADD 'TABLENAME' TO | REMOVE 'TABLENAME' FROM  
] WRITERNAME ;
```

Exemple

```
FAILOVER_SYNC ADD 'master.names' TO ObjectServerA;  
FAILOVER_SYNC ADD 'master.groups' TO ObjectServerA;  
FAILOVER_SYNC ADD 'master.members' TO ObjectServerA;  
FAILOVER_SYNC ADD 'master.permissions' TO ObjectServerA;  
FAILOVER_SYNC ADD 'master.profiles' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.actions' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.action_access' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.menus' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.menu_defs' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.menu_items' TO ObjectServerA;  
FAILOVER_SYNC ADD 'tools.prompt_defs' TO ObjectServerA;  
FAILOVER_SYNC ADD 'alerts.conversions' TO ObjectServerA;  
FAILOVER_SYNC ADD 'alerts.col_visuals' TO ObjectServerA;  
FAILOVER_SYNC ADD 'alerts.colors' TO ObjectServerA;  
FAILOVER_SYNC ADD 'alerts.objclass' TO ObjectServerA;  
FAILOVER_SYNC ADD 'alerts.objmenus' TO ObjectServerA;  
FAILOVER_SYNC ADD 'alerts.objmenuitems' TO ObjectServerA;
```

SET LOG LEVEL

La commande **SET LOG LEVEL** peut être utilisée dans le fichier de commandes de démarrage. La commande ne prend pas effet immédiatement ; une alternative consiste à configurer le fichier de propriétés.

Syntaxe

```
SET LOG LEVEL TO FATAL|ERROR|WARNING|INFORMATION|DEBUG;
```

Exemple

```
SET LOG LEVEL TO ERROR ;  
go
```

Tâches associées:

«Définition du niveau des messages de débogage», à la page 225

Vous pouvez résoudre les problèmes liés à la passerelle en consultant les messages d'erreur. La passerelle possède une fonction de gestion des erreurs, qui est fournie par la bibliothèque NGTK (Netcool/OMNibus Gateway Toolkit). Vous pouvez indiquer quels messages sont inclus dans les fichiers de débogage.

Chapitre 15. Fichier de définition de réplication de table

Les passerelles ObjectServer peuvent répliquer les données de n'importe quelle table entre les ObjectServers. Les détails des tables à répliquer sont stockés dans le fichier de définition de réplication de table. Le fichier de définition de réplication de table définit les tables de l'ObjectServer source que la passerelle ObjectServer réplique dans l'ObjectServer cible. La passerelle lit cette table uniquement au démarrage. Si vous modifiez cette table lors de l'exécution de la passerelle, redémarrez la passerelle de sorte que les modifications prennent effet.

Pour les passerelles unidirectionnelles, la propriété **Gate.Reader.TblReplicateDefFile** indique l'emplacement du fichier de définition de réplication de table.

Pour les passerelles bidirectionnelles, les propriétés **Gate.ObjectServerA.TblReplicateDefFile** et **Gate.ObjectServerB.TblReplicateDefFile** indiquent les emplacements des fichiers.

Syntaxe

La syntaxe du fichier de définition de réplication de table est la suivante :

```
REPLICATE {ALL | INSERTS, UPDATES, DELETES, FT_INSERTS, FT_UPDATES, FT_DELETES}
FROM TABLE table_source
  USING MAP nom_mappe
  [FILTER WITH clause_filtre] [INTO table_destination] [ORDER BY chaîne_ordre]
  [ASC | DESC] [WITH NORESYNC]
  [RESYNC DELETES FILTER condition]
  [SET UPDTOINS CHECK TO {ENABLED|DISABLED|FORCED}]
  [AFTER IDUC DO commande]
  [CACHE FILTER condition]
```

Le tableau ci-dessous décrit les variables de la syntaxe.

Variable	Description
<i>table_source</i>	Table à répliquer dans l'ObjectServer cible.
<i>nom_mappe</i>	Nom de la définition de mappe qui définit la table.
<i>clause_filtre</i>	Filtre utilisé par la passerelle pour sélectionner des lignes à répliquer. Par défaut, le filtrage est inclusif, ce qui signifie que le filtre n'envoie que les événements qui correspondent à la définition de filtre. Pour envoyer des événements qui ne correspondent pas à la définition de filtre, faites précéder le signe égal (=) d'un point d'exclamation (!). Par exemple, la clause de filtre suivante envoie tous les événements dont la gravité n'est pas paramétrée sur 5 : FILTER with 'Severity !=5'.
<i>table_destination</i>	Table devant recevoir la table répliquée. Si cette clause est omise, le nom de la table de destination est identique à la valeur de <i>table_source</i> .

Variable	Description
<i>chaîne_ordre</i>	Liste des noms de colonne séparés par des virgules. Chaque nom de colonne peut être suivi de ASC ou DESC, indiquant si les valeurs dans la colonne doivent être dans l'ordre ascendant ou descendant.
<i>condition</i>	Condition SQL que la passerelle ajoute à l'instruction SELECT lorsqu'elle limite les entrées de cache extraites par la passerelle durant une actualisation du cache.
<i>nom_propriété</i>	Propriété utilisée par la passerelle pour filtrer les données tabulaires. Seules les lignes qui satisfont au filtre sont répliquées.
<i>valeur_propriété</i>	Argument à utiliser dans le filtre.
<i>table_cible</i>	Nom de la table dans laquelle répliquer les données.
<i>clause_filtre_suppression</i>	Définit le filtre de suppression de resynchronisation que la passerelle émet vers l'ObjectServer cible.

Le tableau ci-dessous décrit les options de la commande REPLICATE.

Option	Description
ALL	Equivalent de INSERTS, UPDATES, DELETES.
WITH NORESYNC	Facultatif : indique les tables à ne pas resynchroniser.
ORDER BY	<p>Facultatif : spécifie l'ordre dans lequel les lignes sont renvoyées à la passerelle par l'ObjectServer. Pour chaque colonne, vous pouvez indiquer la façon dont les lignes sont triées.</p> <ul style="list-style-type: none"> • ASC : indique que les valeurs de la colonne sont triées dans l'ordre ascendant. • DESC : indique que les valeurs de la colonne sont triées dans l'ordre descendant. <p>Si vous n'indiquez aucune de ces options, la valeur ASC est utilisée.</p> <p>Pour définir plusieurs colonnes de tri, spécifiez une liste séparée par des virgules, par exemple : ORDER BY 'Serial DESC, StateChange ACS'</p>
RESYNC DELETE FILTER	Facultatif : définit un filtre de suppression de resynchronisation, qui spécifie les lignes à supprimer avant l'insertion dans la table cible. Ce filtre est utilisé lorsque les lignes des tables cible et source ne sont pas une correspondance exacte.

Option	Description
SET UPDTOINS CHECK TO	<p>Facultatif : vous permet de configurer la fonctionnalité de conversion des mises à jour en insertions.</p> <ul style="list-style-type: none"> • ENABLED : la passerelle exécute des conversions de mise à jour en insertion normales. Si une mise à jour reçue de l'ObjectServer source contient une ligne de table qui n'existe pas dans l'ObjectServer cible, la mise à jour est convertie en insertion de sorte que la table est de nouveau remplie dans l'ObjectServer cible. Il s'agit de la valeur par défaut. • DISABLED : pour chaque mise à jour reçue de l'ObjectServer source, la passerelle envoie une mise à jour à l'ObjectServer de destination. Si la mise à jour contient des lignes inexistantes dans la cible, ces lignes sont supprimées. • FORCED : la passerelle convertit toutes les mises à jour reçues de l'ObjectServer source en insertion sur l'ObjectServer cible. Si la ligne existe déjà dans l'ObjectServer cible, la ligne est dé-dupliquée. Ce comportement est identique au comportement des sondes.
AFTER IDUC DO	<p>Facultatif : spécifie une colonne et une valeur associée, que la passerelle applique à toutes les lignes qui ont été insérées, mises à jour ou supprimées.</p>
CACHE FILTER	<p>Facultatif : réduit la quantité de données extraites par la passerelle durant une actualisation de cache de la passerelle unidirectionnelle. La condition est ajoutée à la fin de l'instruction SELECT qui sert à extraire les entrées de cache.</p> <p>Si vous souhaitez utiliser l'option CACHE FILTER, il doit s'agir de la dernière entrée dans la définition de réplication de table.</p>

Effets de la transmission des suppressions sur la taille de mémoire

L'activation ou la désactivation de la transmission des suppressions peut avoir une incidence sur l'utilisation de la mémoire de la passerelle.

Le tableau ci-dessous décrit les effets de l'activation ou la désactivation de la transmission des suppressions sur la mémoire.

Tableau 62. Effets de la transmission des suppressions sur la mémoire

Statut de la transmission des suppressions	Exemple de commande	Effet sur la mémoire
Activée	REPLICATE INSERTS, UPDATES	Si l'ObjectServer cible n'est pas suffisamment rapide pour suivre le flux des données qui lui sont envoyées, l'utilisation de la mémoire de la passerelle augmente.
Désactivée	REPLICATE ALL	L'associateur de passerelle supprime les détails de suppression et ne les transmet pas au programme d'écriture. Pour empêcher de conserver les entrées des lignes supprimées dans le cache, l'associateur de passerelle intervient au nom du programme d'écriture de la passerelle en supprimant du cache de l'ObjectServer cible les entrées des lignes supprimées. Ce comportement aide à stabiliser l'utilisation de la mémoire.

Exemple de fichier de définition de réplication de table

Le fichier de définition de réplication de table définit la façon dont la passerelle ObjectServer réplique des tables entre les serveurs ObjectServers source et cible. Utilisez cet exemple afin de vous familiariser avec le mode de fonctionnement du fichier.

L'exemple suivant présente un fichier de définition de réplication de table :

```
REPLICATE INSERT, DELETE FROM TABLE 'alerts.status'
  USING MAP 'StatusMap'
ORDER BY 'Serial ASC'
FILTER WITH 'Severity!=5'
SET UPDTOINS CHECK TO FORCED
AFTER IDUC DO 'Location=\PASSED BY GW\''
CACHE FILTER 'ServerName IN (\NCOMBS_P\',\NCOMBS_B\)';
```

```
REPLICATE ALL FROM TABLE 'alerts.journal' USING MAP 'JournalMap';
```

```
REPLICATE ALL FROM TABLE 'alerts.details' USING MAP 'DetailsMap';
```

```
#####
# REMARQUE : Si la réplication des tables système associées à l'utilisateur
# est obligatoire, annulez le commentaire des définitions de réplication
# ci-dessous. Les mappes associées doivent également voir leur commentaire
# annulé.
#####
```

```
# REPLICATE ALL FROM TABLE 'security.users'
# USING MAP 'SecurityUsersMap'
# INTO 'transfer.users';
```

```
#
# REPLICATE ALL FROM TABLE 'security.groups'
# USING MAP 'SecurityGroupsMap'
# INTO 'transfer.groups';
#
# REPLICATE ALL FROM TABLE 'security.roles'
# USING MAP 'SecurityRolesMap'
# INTO 'transfer.roles';
#
# REPLICATE ALL FROM TABLE 'security.role_grants'
# USING MAP 'SecurityRoleGrantsMap'
# INTO 'transfer.role_grants';
#
# REPLICATE ALL FROM TABLE 'security.group_members'
# USING MAP 'SecurityGroupMembersMap'
# INTO 'transfer.group_members';
```

Annexe A. Messages d'erreur des sondes et techniques d'identification et de résolution des problèmes

Certains messages d'erreur sont communs à toutes les sondes. Ils incluent les messages ProbeWatch et TSMWatch. Des informations d'identification et de résolution des problèmes sont également disponibles pour les sondes.

Consultez la documentation sur les sondes individuelles pour obtenir des informations sur les messages spécifiques aux sondes.

Messages d'erreur génériques

Les sondes peuvent générer les types de messages suivants : fatal, erreur, avertissement, information et débogage.

Messages de niveau fatal

La sonde s'arrête automatiquement quand un message fatal a été émis.

Tableau 63. Messages de sonde de niveau fatal

Message	Description	Action
Connection to ObjectServer marked DEAD - aborting...	La connexion au serveur ObjectServer a été interrompue (et le mode stocker et transmettre n'est pas activé dans la sonde).	Assurez-vous que le serveur ObjectServer est disponible.
Failed to access OMNIHOME directory: "directory name" Failed to set interfaces file location	La sonde n'a pas pu localiser le fichier d'interfaces.	Vérifiez que la variable d'environnement OMNIHOME est définie sur la destination correcte.
Failed to connect - aborting	Le serveur ObjectServer est indisponible.	Assurez-vous que le serveur ObjectServer est en cours d'exécution, que le fichier d'interfaces du système sur lequel la sonde est installée contient une entrée pour le serveur ObjectServer et qu'il n'y a aucun incident réseau entre les deux systèmes.
Failed to create property Failed to define argument Failed to initialise Failed to set property Failed to process arguments Session create failed - aborting	Erreurs internes.	Consultez votre contrat de support afin d'obtenir des informations pour savoir comment contacter service de support logiciel IBM.
Failed to read rules - aborting	Une propriété ou une option de ligne de commande indique un fichier de règles inexistant.	Assurez-vous que l'option de ligne de commande ou le fichier de propriétés fait référence au fichier de règles correct.

Tableau 63. Messages de sonde de niveau fatal (suite)

Message	Description	Action
Field "field name" not found in status table No matching field found for "field name"	Le fichier de règles utilisé fait référence à une zone au format @fieldname, qui n'existe pas dans la table de statut.	Vérifiez le fichier de règles et corrigez le problème.
Unknown data type returned from ObjectServer	Le serveur ObjectServer a renvoyé des données inconnues.	Consultez votre contrat de support afin d'obtenir des informations pour savoir comment contacter service de support logiciel IBM.

Messages de niveau erreur

Il est probable que la sonde s'arrête lorsqu'un message d'erreur est émis.

Tableau 64. Messages de sonde de niveau erreur

Message	Description	Action
Can't set generic property "property name" via command line Property "property name" for option "option name" does not exist	Une option de la sonde n'est pas mappée correctement sur une propriété.	Recherchez la propriété indiquée dans le fichier de propriétés et consultez la documentation de la sonde pour connaître les propriétés prises en charge.
Could not send alert	La sonde n'a pas pu envoyer d'alerte (généralement une alerte interne) au serveur ObjectServer.	Assurez-vous que le serveur ObjectServer est disponible.
Could not set "fieldname" field	La sonde n'a pas pu définir une valeur de zone. Cela peut être dû au fait que les tables du serveur ObjectServer ont été modifiées pour que les zones par défaut ne soient plus présentes.	Vérifiez si les tables du serveur ObjectServer ont été modifiées.
CreateAndSet failed CreateAndSet failed for attr: "element name"	La sonde n'a pas pu créer d'élément.	Consultez votre contrat de support afin d'obtenir des informations pour savoir comment contacter service de support logiciel IBM.
Error Setting SIGINT Handler Error Setting SIGQUIT Handler Error Setting SIGTERM Handler	La sonde n'a pas pu configurer de gestionnaire de signaux pour un INT, QUIT ou TERM.	Consultez votre contrat de support afin d'obtenir des informations pour savoir comment contacter le service de support logiciel IBM.
Failed to open file: "file name"	Un fichier référencé dans le fichier de règles (par exemple avec la fonction table) est inexistant. Un problème de communication est survenu lors de l'accès à un fichier distant via HTTP.	Assurez-vous que le fichier de règles est disponible.
Failed to open message log: "file name"	La sonde ne peut pas ouvrir le fichier journal spécifié.	Vérifiez la ligne de commande ou le fichier de propriétés et corrigez le problème.

Tableau 64. Messages de sonde de niveau erreur (suite)

Message	Description	Action
Failed to open Properties file: "nom du fichier de propriétés"	La sonde ne peut pas ouvrir le fichier de propriétés.	Vérifiez le fichier de propriétés ou la ligne de commande pour vous assurer que le fichier de propriétés se trouve bien à l'emplacement spécifié.
Failed to open Rules file: "rules file name" The rules file for the probe is not available or incorrectly specified.	La sonde ne peut pas ouvrir le fichier de règles.	Vérifiez le fichier de propriétés ou la ligne de commande pour vous assurer que le fichier de règles se trouve bien à l'emplacement spécifié.
No extraction data for "regexp" - missing ()'s? Regexp doesn't match for "string"	Des parenthèses peuvent manquer dans une expression régulière utilisée dans la fonction extract. Les données de chaîne utilisées pour l'extraction peuvent ne pas correspondre à l'expression régulière. La fonction extract ne peut pas extraire de données.	Vérifiez le fichier de règles et corrigez le problème.
Option "option name" used without argument	L'option utilisée attend une valeur qui n'a pas été fournie.	Consultez la documentation de la sonde et vérifiez le contenu de la ligne de commande.
OS Error: "error message" Procedure "procedure name": "error message" Server "nom du serveur": "message d'erreur"	Une erreur s'est produite dans la connexion à Sybase. La sonde devrait envoyer un autre message qui détaille les effets de cette erreur.	Consultez votre contrat de support afin d'obtenir des informations pour savoir comment contacter service de support logiciel IBM.
Properties file: "error description" at line "line no"	Le format du fichier de propriétés comporte une erreur.	Vérifiez le fichier de propriétés à la ligne spécifiée et corrigez l'erreur.
PropGetValue failed	Une propriété obligatoire n'a pas été définie.	Vérifiez le fichier de propriétés.
Regular Expression Error: "regexp"	Une expression régulière est formée de manière incorrecte dans le fichier de règles.	Recherchez l'expression régulière dans le fichier de règles et corrigez l'erreur.
Results processing failed Unexpected return from results processing Unexpected value during results processing	Un incident est apparu sur le serveur ObjectServer.	Consultez votre contrat de support afin d'obtenir des informations pour savoir comment contacter service de support logiciel IBM.
Rules file: "error description" at line "line no"	La syntaxe ou le format du fichier de règles comporte une erreur.	Vérifiez le fichier de règles à la ligne spécifiée et corrigez l'erreur.
SendAlert failed	La sonde n'a pas pu envoyer d'alerte au serveur ObjectServer.	Assurez-vous que le serveur ObjectServer est disponible.
SessionProcess failed	La sonde n'a pas pu traiter l'alerte par rapport au fichier de règles.	Consultez votre contrat de support afin d'obtenir des informations pour savoir comment contacter service de support logiciel IBM.

Tableau 64. Messages de sonde de niveau erreur (suite)

Message	Description	Action
Unknown message level "message level string" - using WARNING level	Le fichier de propriétés ou la ligne de commande a spécifié un niveau de message non pris en charge.	Vérifiez le fichier de propriétés ou la ligne de commande et utilisez un niveau de message pris en charge (debug, info, warning, error, fatal).
Unknown option: "option name"	Une option non prise en charge par la sonde a été utilisée sur la ligne de commande pour démarrer la sonde.	Consultez la documentation de la sonde et vérifiez le contenu de la ligne de commande.
Unknown property "property name" - ignored	Une propriété spécifiée dans le fichier de propriétés n'existe pas dans la sonde.	Recherchez la propriété indiquée dans le fichier de propriétés et consultez la documentation de la sonde pour connaître les propriétés prises en charge.
HTTP request with URL: "Remote rules file URL" failed. Server responded with "Failure Number" "Failure Text"	Le fichier de règles distant spécifié n'a pas été renvoyé correctement par le serveur HTTP.	Vérifiez la réponse et corrigez le problème qui affecte le serveur HTTP.
Failed to add client HTTP properties Failed to add client HTTP arguments Failed to add create client HTTP context Failed to start client HTTP context Failed to create client HTTP request structure	Un incident s'est produit lors de la configuration de la communication avec les serveurs HTTP distants.	Vérifiez les propriétés HTTP dans le fichier de propriétés ainsi que les arguments HTTP. Assurez-vous que la sonde dispose d'assez de mémoire.
Failed to send the HTTP request to the server.	Il y a un problème de communication avec le serveur de fichiers de règles distant.	Vérifiez que la communication est correctement configurée avec le serveur de fichiers de règles distant.
Unknown remote file URL format	L'adresse URL distante possède un préfixe dont le schéma n'est actuellement pas pris en charge dans ce mode.	Ajoutez à l'URL le préfixe http://
Failed to create a temporary file to write data to "filename" Failed to copy temporary file name "filename" Failed to open temporary local http/https response file "filename" Failed to stat file "filename"	Il y a un problème avec le fichier temporaire local utilisé pour recevoir les données d'un fichier distant.	Assurez-vous que la sonde peut accéder à l'emplacement du fichier temporaire et le modifier.

Messages de niveau avertissement

Ces messages sont émis en tant qu'avertissements mais ne doivent normalement pas provoquer l'arrêt de la sonde.

Tableau 65. Messages de sonde de niveau avertissement.

Message	Description	Action
Failed to install Client Message Callback Failed to install Server Message Callback Failed to retrieve connection status - attempting to continue Results processing failed	Un incident est apparu sur le serveur ObjectServer.	La sonde tente de poursuivre son exécution.
Failed to set SYBASE in environment	La sonde n'a pas pu substituer la variable d'environnement SYBASE.	Vérifiez que la variable d'environnement SYBASE est correctement définie.
New value for field "field name" truncated to "number" characters	Une chaîne en cours de copie dans une zone d'alerte a été tronquée pour s'adapter à la zone.	Vérifiez le fichier de règles.
Type mismatch for property "property name" - new value ignored	Une tentative a eu lieu en vue de définir une propriété avec un type de données incorrect.	Vérifiez le fichier de propriétés ou la ligne de commande pour vous assurer que la propriété est correctement définie.
Downloaded file "nom du fichier distant" contains no usable data Downloaded file "nom du fichier distant" contains no data	Le fichier distant a été téléchargé correctement mais il ne contient aucune donnée.	Assurez-vous que vous souhaitez bien inclure un fichier distant vide.
Failed to unlink file "nom-fichier" - it will not be deleted when closed	Le fichier temporaire utilisé pour stocker les données reçues d'un fichier de règles distant n'a pas été marqué en vue de sa suppression par le système d'exploitation. Ce fichier continue d'exister après que la sonde a terminé de l'utiliser.	Supprimez le fichier. Assurez-vous que la sonde peut supprimer des fichiers dans le répertoire temporaire indiqué.

Messages de niveau information

Ce message est utilisé à des fins d'information

Tableau 66. Messages de sonde de niveau information

Message	Description	Action
Using stderr for logging	La sonde n'a pas pu ouvrir de fichier journal.	Aucune action requise. La sonde écrit les messages dans stderr.

Messages de niveau débogage

Les messages de niveau débogage fournissent des informations sur les fonctions internes de la sonde. Ils sont destinés aux développeurs de sonde mais sont répertoriés ici dans une but d'exhaustivité.

Tableau 67. Messages de sonde de niveau débogage

Message	Description	Action
A value for "string" doesn't exist in lookup table "table name"	Une valeur demandée à une table de consultation est indisponible.	Aucune intervention n'est requise. La fonction du fichier de règles renvoie une chaîne vide.
Attempted to duplicate NULL string Attempted to free NULL pointer Attempted to realloc NULL pointer Failed to allocate memory (Requested size was "number" bytes) Failed to duplicate string Failed to reallocate memory block at address "hex address" (Requested size was "number" bytes)	Une erreur ou un incident s'est produit dans les composants d'allocation de mémoire ou de manipulation de chaîne de la bibliothèque de sondes.	Aucune intervention n'est requise. La bibliothèque gère le problème.
Failed to allocate command structure Failed to allocate context structure Failed to bind column Failed to connect Failed to describe column Failed to fetch number of columns Failed to initialise Sybase internals: "number" Failed to send command Failed to set appname Failed to set command query Failed to set hostname Failed to set password Failed to set username Got a row fail - continuing No columns in result set	Une erreur ou un problème s'est produit au niveau de la connexion avec Sybase ou au serveur ObjectServer.	N/D

Tableau 67. Messages de sonde de niveau débogage (suite)

Message	Description	Action
Failed to flush alerts before EXIT Problem during disconnect before EXIT Problem during session destruction before EXIT Problem during shutdown before EXIT	Un incident s'est produit pendant l'arrêt de la sonde.	N/D
New value for field "field name" is "value"	Une valeur de zone a été définie.	N/D
OpInitialise() called more than once	Plusieurs appels ont été effectués pour la fonction API de la sonde OpInitialise C, qui ne peut être appelée qu'une seule fois.	N/D
Opening URL: "Remote filename" Opening file: "Remote filename"	Le fichier de règles distant est en cours d'ouverture.	Aucune action n'est requise, sauf si vous êtes en train de déboguer un problème concernant ce fichier de règles distant.
HTTP response is: "Server response" HTTP response payload type is: "Type of encoding" Character set conversion: "Conversion used or none" HTTP response message header is: "One line of http message header"	Informations concernant la manière dont le serveur HTTP a répondu lors de la livraison d'un fichier de règles distant.	Aucune action n'est requise, sauf si vous êtes en train de déboguer un problème concernant ce fichier de règles distant.

Messages ProbeWatch et TSMWatch

Dans certains cas, une sonde ou TSM génère des événements de lui-même. Ces événements peuvent fournir des informations (tels que les messages de démarrage ou d'arrêt) ou identifier des problèmes.

Certains éléments sont communs à tous les messages ProbeWatch et TSMWatch.

Les messages ProbeWatch et TSMWatch sont traités dans le fichier de règles et convertis en alertes comme d'autres événements. Le tableau suivant indique les éléments communs aux événements ProbeWatch et TSMWatch.

Tableau 68. Eléments communs à ProbeWatch et TSMWatch

Nom de l'élément	Description
Résumé	Chaîne récapitulative, décrite dans les tableaux suivants.
Noeud	Nom du noeud sur lequel la sonde ou TSM est en cours d'exécution.
Agent	Nom de la sonde ou TSM.
Gestionnaire	ProbeWatch ou TSMWatch.

Le tableau suivant décrit les chaînes récapitulatives communes à toutes les sondes et à TSM.

Tableau 69. Chaînes récapitulatives communes à ProbeWatch et TSMWatch

Message ProbeWatch/TSMWatch	Description	Cause
Going down ...	La sonde ou TSM est en cours de fermeture.	La sonde ou TSM est en train d'exécuter une routine de fermeture.
Running ...	L'exécution de la sonde ou TSM a démarré.	La sonde ou TSM vient de démarrer.
Unable to get events ...	La sonde ou TSM a rencontré un incident pendant l'écoute des événements.	Un incident d'initialisation de la connexion ou une défaillance de la connexion ou de la licence après la réception d'événements s'est produit. Consultez votre contrat de support afin d'obtenir des informations pour savoir comment contacter service de support logiciel IBM.
Rules file reread upon SIGHUP successful ...	La sonde a relu avec succès son fichier de règles à réception du signal SIGHUP.	La sonde a reçu un signal SIGHUP.
Rules file reread upon SIGHUP failed ...	La sonde n'a pas pu relire son fichier de règles à réception du signal SIGHUP.	La sonde a reçu un signal SIGHUP.
Heartbeat ...	Événement de signal de présence	N/D

Consultez la documentation sur les sondes individuelles pour obtenir des chaînes récapitulatives supplémentaires pour chaque sonde.

Les messages TSMWatch et ProbeWatch sont au même format. Le tableau suivant décrit les chaînes récapitulatives communes à tous les TSM.

Tableau 70. Chaînes récapitulatives communes de TSMWatch

Message TSMWatch	Description	Action
Connection Attempted ...	Messages liés à l'établissement d'une connexion TCP/IP.	N/D
Connection Succeeded ...		
Connection Failed ...		
Connection Timed out ...		
Connection Lost ...		
Disconnection Attempted ...	Messages liés à l'abandon d'une connexion TCP/IP.	N/D
Disconnection Succeeded ...		
Disconnection Failed ...		
Wakeup Attempted ...	Messages liés à la fonctionnalité de réveil.	N/D
Wakeup Succeeded ...		
Wakeup Failed ...		
Login Attempted ...	Messages liés à la connexion à l'hôte.	N/D
Login Succeeded ...		
Login Timed out ...		
Login Failed ...		

Tableau 70. Chaînes récapitulatives communes de TSMWatch (suite)

Message TSMWatch	Description	Action
Logout Attempted ... Logout Succeeded ... Logout Timed out ... Logout Failed ...	Messages liés à la déconnexion de l'hôte.	N/D
Heartbeat Sent ... Heartbeat Received ... Heartbeat Timed out ...	Messages liés à l'envoi et à la réception de messages de contact à destination et de l'hôte.	N/D
Resynchronisation Attempted ... Resynchronisation Succeeded ... Resynchronisation Failed ...	Messages liés à la synchronisation des alertes actuelles entre le commutateur et Tivoli Netcool/OMNIBus.	N/D

Identification et résolution des problèmes de sonde

Cette rubrique décrit certains problèmes communs rencontrés par les utilisateurs de Tivoli Netcool/OMNIBus et explique les causes et solutions possibles.

Ces informations d'identification et de résolution des problèmes peuvent être divisées en deux groupes :

- Causes de problèmes communes
- Que faire si

Tableau 71. Identification et résolution des problèmes de sonde

Groupe	Description
Causes de problèmes communes	Ces informations incluent une liste des causes de problèmes communes. Si vous ne connaissez pas exactement la nature du problème, commencez par lire cette section et suivez les instructions données. Si le problème persiste, passez aux informations de la section "Que faire si".
Que faire si	Ces informations décrivent les symptômes communs provoqués par les problèmes de sonde et fournissent des instructions étape par étape pour vous aider à localiser et à résoudre le problème. Si aucun titre ne correspond aux symptômes de votre incident, lisez la liste des instructions et assurez-vous d'avoir essayé toutes les solutions les plus probables qui y sont répertoriées.

Causes de problèmes communes

Les causes de problèmes de sonde les plus communes sont :

- Une définition incorrecte de la variable d'environnement OMNIHOME
- Des erreurs dans le fichier de règles, en particulier dans les instructions extract
- Des erreurs de configuration dans le fichier de propriétés

Pour plus d'informations sur la définition de la variable d'environnement OMNIHOME, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIBus*.

Vérifiez que toutes les propriétés sont correctement définies dans le fichier de propriétés de sonde. Vérifiez, par exemple, que la propriété **Server** contient le nom du serveur ObjectServer ou de serveur proxy correct et que la propriété **RulesFile** contient le nom de fichier de règles correct.

Si vous ne pouvez pas résoudre le problème, lisez la section suivante et assurez-vous d'avoir essayé toutes les solutions probables répertoriées.

Que faire si

Les titres de cette rubrique décrivent les symptômes les plus communs des incidents liés aux sondes. Recherchez le titre qui correspond le mieux à votre incident et suivez les instructions jusqu'à ce que vous ayez identifié la cause et résolu votre incident.

Si aucun titre ne correspond aux symptômes de votre incident, lisez la liste des instructions et assurez-vous d'avoir essayé toutes les solutions les plus probables qui y sont répertoriées. Si vous avez essayé toutes les solutions suggérées et que le problème persiste, contactez le service de support logiciel IBM.

La sonde ne démarre pas

Si la sonde ne démarre pas :

1. Exécutez la sonde en mode débogage.
2. Vérifiez que le serveur ObjectServer est en cours d'exécution en tentant de vous connecter à l'aide de l'utilitaire **nco_sq1**.

Si vous parvenez à vous connecter, le serveur ObjectServer est en cours d'exécution. Si le serveur n'est pas exécuté, cela peut être la cause du problème.

3. Vérifiez qu'aucune autre sonde n'est exécutée avec la même configuration à l'aide de la commande suivante :

```
ps -ef | grep nco_p
```

Une liste des processus de sonde est affichée. Vérifiez qu'aucun des processus ne correspond au même type de sonde. Vous ne pouvez pas exécuter deux configurations de sonde identiques car cela duplique tous les événements transmis au serveur ObjectServer.

4. Veillez à utiliser la sonde correcte correspondant à la version actuelle du logiciel cible.
5. Assurez-vous que le fichier de règles ne comporte aucune erreur de syntaxe.
6. Vérifiez que votre système dispose de ressources systèmes suffisantes et qu'il peut lancer plus de processus. Pour ce faire, utilisez `df -k` ou `top`. Voir les pages concernant les commandes `df` et `top` pour obtenir plus d'informations sur l'utilisation de ces commandes.
7. Vérifiez que le fichier stocker-et-transmettre `$OMNIHOME/var/probename.saf` existe. Si tel est le cas, assurez-vous qu'il n'est pas devenu trop volumineux. Si votre disque est plein, les sondes et serveurs ObjectServer ne peuvent pas fonctionner correctement.

Avvertissement : Le fichier stocker et transmettre n'est pas conçu pour gérer un grand nombre d'événements. S'il est laissé sans surveillance, un fichier stocker-et-transmettre continue de grandir jusqu'à ce qu'il n'ait plus d'espace disque.

8. Vérifiez que le fichier stocker-et-transmettre n'a pas été endommagé. S'il est endommagé, un message d'erreur doit se trouver dans le fichier journal (\$OMNIHOME/log/nom_sonde.log). Si le fichier est endommagé, supprimez-le et redémarrez la sonde.
9. Vérifiez que le fichier binaire de la sonde que vous tentez d'exécuter est correct pour l'architecture en cours en entrant :
`$OMNIHOME/bin/arch/nom_sonde -version`
 Assurez-vous que la version de la sonde correspond à l'architecture de votre système.
Si vous exécutez la sonde sur un hôte distant :
10. Vérifiez que l'hôte de la sonde peut se connecter à l'hôte du serveur ObjectServer à l'aide de la commande ping. Essayez de lancer une commande PING sur la machine hôte du serveur ObjectServer à l'aide du nom d'hôte et de l'adresse IP. Pour plus d'information sur cette procédure, voir la page de la commande ping.
 Si vous ne parvenez pas à vous connecter à l'hôte du serveur ObjectServer à l'aide de la commande ping, il y a un problème de connexion entre l'hôte de la sonde et celui du serveur.
11. Vérifiez que le serveur ObjectServer a été configuré correctement dans l'éditeur de serveur (**nco_xigen**) et que les informations sur les interfaces ont été distribuées aux hôtes du serveur ObjectServer et de la sonde.
12. Vérifiez si un pare-feu est présent entre l'hôte de la sonde et celui du serveur ObjectServer. Si tel est le cas, assurez-vous que le pare-feu autorise le trafic entre la sonde et le serveur ObjectServer.

Tâches associées:

«Test de fichiers de règles», à la page 64

Vous pouvez tester la syntaxe d'un fichier de règles à l'aide du vérificateur de syntaxe de règles de sonde, **nco_p_syntax**. Cela se révèle plus efficace que l'exécution de la sonde pour vérifier que la syntaxe du fichier de règles est correcte.

«Débogage de fichiers de règles», à la page 65

Lorsque vous modifiez le fichier de règles, ajoutez de nouvelles règles ou encore créez des tables de consultation, il se révèle particulièrement utile de tester la sonde en l'exécutant en mode débogage. Le mode débogage montre la manière dont un événement fait l'objet d'une analyse syntaxique effectuée par la sonde et peut détecter les problèmes éventuels relatifs au fichier de règles.

Référence associée:

Chapitre 6, «Propriétés et options de ligne de commande de sonde communes», à la page 129

Certaines propriétés et options de ligne de commande sont communes à toutes les sondes et TSM.

La sonde n'envoie pas d'alerte au serveur ObjectServer

Si la sonde n'envoie pas d'alerte au serveur ObjectServer :

1. Vérifiez que la sonde est en cours d'exécution en entrant :

```
ps -ef | grep nco_p
```

Une liste des processus de sonde est affichée. Si la sonde n'est pas exécutée, démarrez-la à partir de la ligne de commande.

2. Assurez-vous qu'aucune autre sonde n'est exécutée avec la même configuration en entrant :

```
ps -ef | grep nco_p
```

Une liste des processus de sonde est affichée. Vérifiez qu'aucun des processus ne correspond au même type de sonde. Vous ne pouvez pas exécuter deux configurations de sonde identiques car cela duplique tous les événements transmis au serveur ObjectServer.

3. Lisez le fichier de propriétés de sonde et vérifiez que toutes les propriétés sont définies correctement. Vérifiez, par exemple, que la propriété **Server** contient le nom du serveur ObjectServer correct et que la propriété **RulesFile** contient le nom de fichier de règles correct.
4. Assurez-vous que la source d'événements de la sonde contient des événements à envoyer au serveur ObjectServer.
5. Vérifiez que le serveur ObjectServer auquel vous êtes connecté est identique à celui auquel la sonde transmet des événements.
6. Assurez-vous que la source d'événements que vous essayez de sonder fonctionne correctement. Consultez la documentation fournie avec votre gestionnaire d'éléments pour obtenir plus d'informations sur cette procédure.
7. Assurez-vous d'utiliser la sonde correcte.
8. Vérifiez que la sonde n'est pas exécutée en mode stocker-et-transmettre. Pour ce faire, vérifiez les fichiers `$OMNIHOME/var/nom_sonde.saf` et `$OMNIHOME/var/nom_sonde.reco` pour voir si leur taille augmente. Si tel est le cas, désactivez le mode stocker-et-transmettre.
9. Vérifiez que votre système dispose de ressources systèmes suffisantes et qu'il peut lancer plus de processus. Pour ce faire, utilisez `df -k` ou `top`. Voir les pages d'aide concernant l'utilisation de `df` et de `top` pour obtenir plus d'informations sur l'utilisation de cette commande.
10. Recherchez des fonctions `discard` dans le fichier de règles de sonde. La fonction `discard` doit faire partie d'une instruction conditionnelle, sinon tous les événements sont supprimés.

Si vous exécutez la sonde sur un hôte distant :

11. Vérifiez que l'hôte de la sonde peut se connecter à l'hôte du serveur ObjectServer à l'aide de la commande `ping`. Essayez de lancer une commande `PING` sur la machine hôte du serveur ObjectServer à l'aide du nom d'hôte et de l'adresse IP. Pour plus d'information sur cette procédure, voir la page de la commande `ping`.

Si vous ne parvenez pas à vous connecter à l'hôte du serveur ObjectServer à l'aide de la commande `ping`, il y a un problème de connexion entre l'hôte de la sonde et celui du serveur.

12. Vérifiez que le serveur ObjectServer a été configuré correctement via l'éditeur de serveur (**nco_xigen**) et que les informations sur les interfaces ont été distribuées aux hôtes du serveur ObjectServer et de la sonde.
13. Vérifiez si un pare-feu est présent entre l'hôte de la sonde et celui du serveur ObjectServer. Si tel est le cas, assurez-vous que le pare-feu autorise le trafic entre la sonde et le serveur ObjectServer.

Concepts associés:

«Mode stocker-et-transmettre pour les sondes», à la page 12

L'exécution des sondes peut continuer si le serveur ObjectServer cible est arrêté. Pendant cette période, la sonde passe en mode *stocker*. Elle repasse en mode *transmettre* lorsque le serveur ObjectServer fonctionne à nouveau.

La sonde perd des événements

Si tous les événements ne sont pas transmis au serveur ObjectServer :

1. Exécutez la sonde en mode débogage.
2. Assurez-vous que la source d'événements que vous essayez de sonder fonctionne correctement. Consultez la documentation fournie avec votre gestionnaire d'éléments pour obtenir plus d'informations sur cette procédure.
3. Assurez-vous que la source d'événements de la sonde contient des événements à envoyer au serveur ObjectServer.
4. Vérifiez que toutes les propriétés du fichier de propriétés sont définies correctement. Vérifiez, par exemple, que la propriété **Server** contient le nom du serveur ObjectServer correct et que la propriété **RulesFile** contient le nom de fichier de règles correct.
5. Recherchez des fonctions `discard` dans le fichier de règles de sonde. La fonction `discard` supprime les événements basés sur des conditions spécifiées.

Tâches associées:

«Débogage de fichiers de règles», à la page 65

Lorsque vous modifiez le fichier de règles, ajoutez de nouvelles règles ou encore créez des tables de consultation, il se révèle particulièrement utile de tester la sonde en l'exécutant en mode débogage. Le mode débogage montre la manière dont un événement fait l'objet d'une analyse syntaxique effectuée par la sonde et peut détecter les problèmes éventuels relatifs au fichier de règles.

La sonde utilise trop de temps UC

Si la sonde utilise trop de temps UC :

1. Exécutez la sonde en mode débogage.
2. Assurez-vous que la sonde peut se connecter à la source d'événements.
3. Vérifiez que le fichier `stocker-et-transmettre` `$OMNIHOME/var/nom_sonde.saf` existe. Si tel est le cas, vérifiez qu'il n'est pas devenu trop volumineux. Si votre disque est plein, les sondes et le serveur ObjectServer ne pourront pas fonctionner correctement.

Avertissement : Le fichier `stocker` et `transmettre` n'est pas conçu pour gérer un grand nombre d'alertes. S'il est laissé sans surveillance, un fichier `stocker-et-transmettre` continue de grandir jusqu'à ce qu'il n'ait plus d'espace disque.

4. Vérifiez que le fichier `stocker-et-transmettre` n'a pas été endommagé. Si le fichier `stocker-et-transmettre` est endommagé, un message d'erreur doit se trouver dans le fichier journal de la sonde (`$OMNIHOME/log/nom_sonde.log`). Si le fichier `stocker-et-transmettre` est endommagé, supprimez-le et redémarrez la sonde.

La liste d'événements n'est pas renseignée correctement

Si la sonde détecte des événements et les transmet au serveur ObjectServer mais que les zones de la liste d'événements ne sont pas renseignées correctement :

1. Exécutez la sonde en mode débogage.
2. Vérifiez que les zones qui ne sont pas renseignées correctement sont bien mappées aux éléments du fichier de règles.
3. Assurez-vous qu'il ne s'agit pas d'un incident d'interface graphique en interrogeant la table `alerts.status` à l'aide des instructions SQL du serveur ObjectServer.

Vérifier la synchronisation d'un fichier de règles

L'utilitaire Probe Rules Utility vous permet de vérifier qu'un fichier de règles de sonde est bien synchronisé avec la même version de ce fichier stockée dans un référentiel central ou dans un système de fichiers.

Avant de commencer

Cette procédure peut être utilisée si les conditions suivantes sont remplies :

- La sonde charge son fichier de règles via HTTP à partir d'un emplacement distant.
- Probe Rules Utility est installé sur l'ordinateur sur lequel les fichiers de règles résident et il a accès au système de fichiers.
- La sonde est active et connectée à un serveur ObjectServer. Vous devez également connaître le nom et le nom d'hôte de la sonde, ainsi que l'emplacement des fichiers de règles qu'elle utilise.

Si vos fichiers de règles contiennent des variables d'environnement (par exemple, `NC_RULES_HOME`), vous devez définir ces variables avant d'exécuter Probe Rules Utility afin que ce dernier puisse les résoudre.

Pourquoi et quand exécuter cette tâche

Lorsqu'une sonde se connecte à un serveur ObjectServer, elle ajoute une entrée à la table `registry.probes`. Cette entrée contient des informations sur la sonde, et notamment le total de contrôle de son fichier de règles. Ce total de contrôle est propre à chaque fichier de règles et aux fichiers Include et Lookup qui en dépendent. Probe Rules Utility génère la même valeur de total de contrôle, ce qui permet de vérifier si la sonde utilise bien la version correcte d'un fichier de règles.

Procédure

1. Démarrez Netcool/OMNIBus Administrator (`nco_config`) et cliquez sur **Systeme > SQL**.
2. Exécutez la requête SQL suivante :

```
select Name,Hostname,RulesChecksum from registry.probes;
```

Cette requête renvoie la liste des sondes enregistrées dans la table `registry.probes`, accompagnées de leur nom, de leur nom d'hôte et du total de contrôle de leur fichier de règles.
3. Localisez le nom et le nom d'hôte de votre sonde cible et notez la valeur de total de contrôle du fichier de règles.
4. Utilisez la commande Probe Rules Utility suivante pour obtenir le total de contrôle du fichier de règles utilisé par la sonde :

```
$NCHOME/omnibus/bin/nco_rules_util list -checksum nom-fichier
```

Où *nom-fichier* représente le nom du fichier de règles.

5. Comparez le total de contrôle renvoyé par Probe Rules Utility à celui renvoyé à l'étape 2.

Si ces deux totaux sont identiques, la sonde utilise bien la version du fichier de règles qui est enregistrée dans la table registry.probes. Si ces totaux sont différents, la sonde utilise une version non synchronisée.

6. Si le port de commande HTTP de la sonde est ouvert, vous pouvez demander à la sonde de recharger son fichier de règles :

- a. Pour recharger le fichier de règles, utilisez la commande utilitaire **nco_probereloadrules** suivante :

```
$NCHOME/omnibus/bin/nco_probereloadrules -host nom-d'hôte -port port
```

Où *nom-d'hôte* et *port* représente le nom d'hôte et le port de commande HTTP de la sonde.

- b. Répétez l'étape 2 et assurez-vous que le total de contrôle du fichier de règles a bien été mis à jour.

Tâches associées:

«Rechargement de fichiers de règles (nco_probereloadrules)», à la page 114

Vous pouvez utiliser l'utilitaire **nco_probereloadrules** pour recharger à distance un fichier de règles de sonde sans avoir à redémarrer la sonde.

Référence associée:

«Probe Rules Utility», à la page 72

Vous pouvez utiliser l'outil de ligne de commande de Probe Rules Utility (**nco_rules_util**) pour gérer plus facilement les fichiers de règles de sonde qui sont stockés dans un référentiel central ou un système de fichiers.

Annexe B. Messages d'erreur de passerelle communs

Certains messages d'erreur sont communs à toutes les passerelles. La variable *nom_passerelle* de chaque message d'erreur fait référence au nom de la passerelle individuelle et indique la passerelle qui a généré l'erreur.

Tableau 72. Messages d'erreur de passerelle communs

Erreur	Description	Action
<p><i>Nom_passerelle</i> Writer: HashAlloc failure in <i>_nom_passerelle</i> CacheAdd().</p> <p><i>Nom_passerelle</i> Writer: MemStrDup() failure in <i>_nom_passerelle</i> CacheAdd().</p>	La passerelle n'a pas pu allouer de mémoire.	Essayez de libérer plus de mémoire.
<p><i>Nom_passerelle</i> Writer: Failed to allocate memory.</p> <p><i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i>: Memory allocation failed.</p> <p><i>nom_passerelle</i> Writer: Memory allocation failure.</p> <p><i>Nom_passerelle</i> Writer: Memory allocation error.</p> <p><i>Nom_passerelle</i> Writer: Memory reallocation error.</p> <p>Failed to allocate memory in writer <i>writer_name</i>.</p>	La passerelle n'a pas pu allouer de mémoire.	Essayez de libérer plus de mémoire.
<p><i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i>: Could not create serial cache - memory problems.</p> <p><i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i>: Failed to allocate memory for a GPCModule handle.</p>	La passerelle n'a pas pu allouer de mémoire.	Essayez de libérer plus de mémoire.
<p><i>Nom_passerelle</i> Writer: Failed to lock connection mutex.</p>	Le programme d'écriture n'a pas pu verrouiller la connexion retour du serveur ObjectServer afin d'accéder à la connexion et de renvoyer les données de ticket d'incident pour l'alerte associée. L'échec de verrouillage peut être dû à des ressources insuffisantes ou résulter du système d'unités d'exécution sous-jacent qui empêche l'interblocage entre plusieurs unités d'exécutions concurrentes pour la ressource.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.

Tableau 72. Messages d'erreur de passerelle communs (suite)

Erreur	Description	Action
<i>Nom_passerelle</i> Writer: Failed to re-acquire alert details from OS.	Ce message d'erreur provient du sous-système de récupération du cache de la passerelle. Il indique que la passerelle n'a pas été en mesure d'acquiescer à nouveau le numéro du ticket d'incident et de récupérer son entrée de cache interne du serveur ObjectServer.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Invalid datatype for problem number feedback field.	Le type de données est invalide.	Pour plus d'informations sur les types de données, consultez le <i>Guide d'administration d'IBM Tivoli Netcool/OMNIBus</i> .
<i>Nom_passerelle</i> Writer: Serial x already in serial Cache. Cannot add.	La passerelle a tenté d'ajouter un numéro de série qui existe déjà.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Serial x not found in serial cache. Cannot Delete.	La passerelle n'a pas pu supprimer cette alerte car elle a déjà été supprimée dans Tivoli Netcool/OMNIBus.	Aucune action n'est requise.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to construct path to <i>nom_passerelle</i> Read/Write Module.	La passerelle n'a pas pu localiser l'application du module de programme de lecture ou d'écriture.	Vérifiez que le module est installé à l'emplacement correct.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to construct the argument list for <i>nom_passerelle</i> Module.	Echec de la génération de la liste d'arguments pour le module de passerelle.	Vérifiez que les arguments du fichier de configuration sont correctement définis.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : GPCModule creation failed.	Echec de la création du module GPCModule en raison d'une mémoire insuffisante.	Essayez de libérer plus de mémoire.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to start the OS- <i>nom_passerelle</i> Writer. <i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to start the <i>nom_passerelle</i> -OS Reader.	Echec du démarrage du module de lecture ou d'écriture de la passerelle ObjectServer.	Vérifiez que le module est installé à l'emplacement correct et que les droits sur les fichiers sont correctement définis.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to shutdown <i>nom_passerelle</i> Writer.	Echec de l'arrêt du module d'écriture de la passerelle.	Consultez le fichier journal du programme d'écriture pour plus d'informations.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to construct path to <i>nom_passerelle</i> Read/Write Module.	Echec de la génération du chemin d'accès à l'application du module de programme de lecture ou d'écriture.	Vérifiez que le module est installé à l'emplacement correct et que les droits sur les fichiers sont correctement définis.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to find the <i>nom_passerelle</i> Read/Write Module [x].	Fichier binaire du module introuvable.	Vérifiez que le module est installé à l'emplacement correct et que les droits sur les fichiers sont correctement définis.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Incorrect permissions on the <i>nom_passerelle</i> module binary [x].	Les droits du fichier du module ne sont pas définis correctement.	Vérifiez que le module est installé à l'emplacement correct et que les droits sur les fichiers sont correctement définis.

Tableau 72. Messages d'erreur de passerelle communs (suite)

Erreur	Description	Action
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to create the Serial Cache Mutex.	Le programme d'écriture de la passerelle n'est pas parvenu à créer la structure de protection de données nécessaire pour le cache de numéro de série interne en raison de ressources insuffisantes. Cet incident est généralement provoqué par une mémoire insuffisante.	Essayez de libérer plus de mémoire.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to create the Conn Mutex.	Le programme d'écriture de la passerelle n'est pas parvenu à créer la structure de protection de données nécessaire pour la connexion au serveur ObjectServer en raison de ressources insuffisantes.	Essayez de libérer plus de mémoire.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to start the <i>nom_passerelle</i> -to-OS service thread.	La passerelle n'a pas pu générer l'unité d'exécution de service.	Vérifiez que la passerelle peut accéder au serveur ObjectServer.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to send a shutdown request to the <i>nom_passerelle</i> Writer.	La passerelle ne s'est pas arrêtée correctement.	Consultez le journal du programme d'écriture pour plus d'informations.
Failed to install SIGCHLD handler. Failed to install SIGPIPE handler.	Echec de la passerelle pendant l'installation du gestionnaire.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
No <nom_mappe> attribute for <i>nom_passerelle</i> writer <i>nom_programme_écriture</i> .	La passerelle n'a pas trouvé le nom de la mappe.	Vérifiez le fichier de configuration.
<nom_mappe> attribute is not a name for <i>nom_passerelle</i> writer <i>nom_programme_écriture</i> .	Nom de programme d'écriture incorrect indiqué.	Vérifiez le fichier de configuration.
A MAP called <mappe> does not exist for <i>nom_passerelle</i> writer <i>nom_programme_écriture</i> .	La passerelle n'a pas trouvé la mappe spécifiée.	Vérifiez le fichier de configuration.
MAP <mappe> is invalid for <i>nom_passerelle</i> writer <i>nom_programme_écriture</i> .	La mappe indiquée est invalide.	Vérifiez le fichier de configuration.
Map <mappe> is not the journal map and cannot contain the <journal map name> map item in <i>nom_passerelle</i> Writer <i>nom_programme_écriture</i> .	Si cette mappe n'est pas la mappe de journal, la définition de l'attribut JOURNAL_MAP_NAME est incorrecte.	Vérifiez l'attribut JOURNAL_MAP_NAME dans le fichier de configuration de la passerelle.
<i>Nom_passerelle</i> Writer: Failed to send <i>nom_passerelle</i> Event to the <i>nom_passerelle</i> Writer module.	La passerelle n'a pas pu envoyer un événement donné.	Vérifiez les fichiers journaux pour plus d'informations.
<i>Nom_passerelle</i> Writer: Failed to wait for return from the <i>nom_passerelle</i> Writer module.	Une erreur s'est produite pendant la récupération de l'instruction de réussite.	Vérifiez les fichiers journaux pour plus d'informations.
<i>Nom_passerelle</i> Writer: Failed to read the status return message from the <i>nom_passerelle</i> Writer module.	La passerelle n'a pas pu extraire le statut d'un module.	Vérifiez les fichiers journaux pour plus d'informations.
<i>Nom_passerelle</i> Writer: Failed to send event to <i>nom_passerelle</i> .	Le module n'a pas pu envoyer l'événement à la passerelle.	Vérifiez les fichiers journaux pour plus d'informations.

Tableau 72. Messages d'erreur de passerelle communs (suite)

Erreur	Description	Action
<i>Nom_passerelle</i> Writer: <i>nom_passerelle</i> Writer Module experienced Fatal Error.	Une erreur fatale s'est produite.	Vérifiez les fichiers journaux pour plus d'informations.
<i>Nom_passerelle</i> Writer: Failed to send event to <i>nom_passerelle</i> . Unknown type.	La passerelle a reçu un type imprévu.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Failed to build serial index.	La passerelle n'a pas pu créer les index.	Vérifiez que la colonne Serial existe dans la table alerts.status du serveur ObjectServer.
Incorrect data type for the Serial column.	La passerelle n'a pas reçu le type de données correct.	Vérifiez que le type de données de la colonne Serial de la table alerts.status du serveur ObjectServer est un entier.
<i>Nom_passerelle</i> Writer: Failed to build server serial index.	La passerelle n'a pas reçu l'index de série du serveur.	Vérifiez que la colonne ServerSerial existe dans la table alerts.status du serveur ObjectServer.
Incorrect data type for the Server Serial column.	La passerelle n'a pas reçu le type de données correct.	Vérifiez que le type de données de la colonne ServerSerial de la table alerts.status du serveur ObjectServer est un entier.
<i>Nom_passerelle</i> Writer: Failed to build server name index.	La passerelle n'a pas reçu l'index de nom du serveur.	Vérifiez que la colonne ServerName existe dans la table alerts.status du serveur ObjectServer.
Incorrect data type for the Server Name column.	La passerelle n'a pas reçu le type de données correct.	Vérifiez que le type de données de la colonne ServerName de la table alerts.status du serveur ObjectServer est une chaîne.
<i>Nom_passerelle</i> Writer: Failed to find field <numéro_zone> in <i>nom_passerelle</i> Event.	La passerelle n'a pas trouvé le numéro de zone recherché.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Invalid field name for expansion on action SQL [<zone>].	La passerelle a reçu un nom de zone invalide.	Pour plus d'informations sur SQL ObjectServer, consultez le <i>Guide d'administration d'IBM Tivoli Netcool/OMNIBus</i> .
<i>Nom_passerelle</i> Writer: Unenclosed field expansion request in action SQL [<action sql>].	La passerelle n'a pas trouvé de crochets d'inclusion.	Vérifiez le fichier action.sql.
<i>Nom_passerelle</i> Writer: Failed to turn counter-part notification back-on. Fatal error in <i>nom_passerelle</i> -to-OS Feedback. <i>Nom_passerelle</i> Writer: Failed to turn counter-part notification off. <i>Nom_passerelle</i> -to-OS Feedback failed.	La passerelle n'a pas pu envoyer de commande notify.	Il s'agit d'une erreur interne. Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Failed to send SQL command to ObjectServer. <i>Nom_passerelle</i> -to-OS Feedback failed.	La passerelle n'a pas pu envoyer la commande SQL au serveur ObjectServer.	Consultez le fichier journal du serveur ObjectServer.

Tableau 72. Messages d'erreur de passerelle communs (suite)

Erreur	Description	Action
Failed to find the column <nom_colonne> in map <nom_mappe>.	La passerelle n'a pas trouvé la colonne indiquée.	Vérifiez que le nom de colonne indiqué est correct dans le fichier de configuration et qu'il est affiché dans la table alerts.status du serveur ObjectServer.
Nom_passerelle Writer: Failed to lock the cache mutex.	Le programme d'écriture n'a pas pu verrouiller la connexion retour du serveur ObjectServer afin d'accéder à la connexion et de renvoyer les modifications des données de ticket d'incident pour l'alerte associée.	L'échec de verrouillage peut être dû à des ressources insuffisantes ou résulter du système d'unités d'exécution sous-jacent qui empêche l'interblocage entre plusieurs unités d'exécutions concurrentes pour la ressource.
Failed to find cached problem ticket for serial <numéro série> using map <nom_mappe>.	La passerelle n'a pas trouvé le numéro de ticket d'incident en cache spécifié.	Vérifiez que le ticket spécifié a été créé par la passerelle.
Nom_passerelle Writer: Failed to unlock the cache mutex.	Après avoir accédé à la mémoire cache, une tentative de déverrouillage du verrou de protection des structures de données à échoué. Ce message indique que la passerelle est dans une position qui entraînera un interblocage.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
Nom_passerelle Writer: Cache add error.	La passerelle n'a pas pu ajouter le numéro de série au cache de numéros de série en raison de ressources insuffisantes.	Essayez de libérer plus de mémoire.
Nom_passerelle Writer nom_programme_écriture: Failed to create nom_passerelle Event for journal update.	La passerelle n'a pas pu créer la mise à jour de l'événement de journal.	Consultez le fichier journal du programme d'écriture.
Nom_passerelle Writer nom_programme_écriture: Failed to send journal update event to nom_passerelle.	La passerelle n'a pas pu envoyer la mise à jour de l'événement de journal.	Consultez le fichier journal du programme d'écriture.
<nom attribut> attribute is not a string for Nom_passerelle writer nom_programme_écriture.	Le type de données d'un attribut du programme d'écriture était incorrect.	Vérifiez la définition du programme d'écriture dans le fichier de configuration.
No <nom attribut> attribute for nom_passerelle writer nom_programme_écriture given.	La passerelle n'a pas trouvé l'attribut.	Ajoutez l'attribut à la définition du programme d'écriture dans le fichier de configuration.
Nom_passerelle Writer nom_programme_écriture: Failed to find the <attribut homologue> attribute for the writer. This is necessary due to bi-directional nature.	Une tentative de trouver l'attribut homologue nécessaire a échoué.	Vérifiez le fichier de configuration.
Nom_passerelle Writer nom_programme_écriture: Is not a name for an Object Server reader.	La passerelle a trouvé un type de données incorrect.	Vérifiez le fichier de configuration.
Nom_passerelle Writer nom_programme_écriture: Reader <programme_lecture> was not found for counter part.	Le programme de lecture est introuvable.	Vérifiez la configuration de l'homologue dans le fichier de configuration.

Tableau 72. Messages d'erreur de passerelle communs (suite)

Erreur	Description	Action
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to send SKIP Command.	Cette commande n'a pas pu désactiver IDUC sur une connexion bidirectionnelle.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
Connection to feedback server failed.	La passerelle n'a pas pu établir de connexion.	Consultez le fichier journal du serveur ObjectServer.
Failed to set the death call on the feedback connection.	La passerelle n'a pas pu définir la propriété nécessaire.	Il s'agit d'une erreur interne. Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
Writer counterpart error.	La passerelle n'a pas trouvé l'attribut homologue pour le programme d'écriture de la passerelle.	Vérifiez la configuration de l'homologue dans le fichier de configuration.
<i>Nom_passerelle</i> Writer: Failed to stat() the action SQL file " <i>nom_fichier</i> ".	La passerelle n'a pas pu envoyer de commande stat sur le fichier pour définir sa taille.	Vérifiez les droits d'accès du fichier d'action spécifié.
<i>Nom_passerelle</i> Writer: Empty action SQL file " <i>nom_fichier</i> ".	Le fichier " <i>nom_fichier</i> " est vide.	Vérifiez le fichier d'actions SQL.
<i>Nom_passerelle</i> Writer: Failed to open the action SQL file " <i>nom_fichier</i> ".	La passerelle n'a pas pu ouvrir le fichier.	Vérifiez les droits du fichier.
<i>Nom_passerelle</i> Writer: Failed to read the action SQL file " <i>nom_fichier</i> ".	La passerelle n'a pas pu lire le fichier.	Vérifiez les droits du fichier.
<i>Nom_passerelle</i> Writer: No Action SQL find in file " <i>nom_fichier</i> ".	Il n'y a aucune action SQL dans le fichier.	Vérifiez le fichier.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to read the conversions table.	La passerelle n'a pas pu lire la table de conversions.	Vérifiez les droits du fichier.
<i>Nom_passerelle</i> Writer: Failed to find PM %s in cache for return PMO event.	La passerelle a reçu un événement retour Problem Management Open de la passerelle pour un ticket d'incident. Aucun enregistrement n'a été trouvé ou récupéré lors d'une tentative de recherche du numéro de ticket d'incident dans le cache du programme d'écriture afin de déterminer le numéro de série de l'alerte associée au ticket.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Open Feedback Failed.	La passerelle n'a pas pu générer l'instruction SQL open action ou envoyer la commande d'action SQL au serveur.	Consultez le fichier SQL du serveur ObjectServer.
<i>Nom_passerelle</i> Writer: No Update action SQL for <i>nom_passerelle</i> Update event.	Il n'y a aucune instruction SQL d'action de mise à jour.	Vérifiez le fichier de configuration.

Tableau 72. Messages d'erreur de passerelle communs (suite)

Erreur	Description	Action
<i>Nom_passerelle</i> Writer: Failed to find PM %s in cache for return PMU event.	La passerelle a reçu un événement retour Problem Management Update de la passerelle pour un ticket d'incident. Lors d'une tentative de recherche du numéro de ticket d'incident dans le cache du programme d'écriture afin de déterminer le numéro de série de l'alerte associée au ticket, aucun enregistrement n'a été trouvé ou récupéré.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Update Feedback Failed.	La passerelle n'a pas pu générer l'instruction SQL open action ou envoyer la commande d'action SQL au serveur.	Consultez le fichier journal du serveur ObjectServer.
<i>Nom_passerelle</i> Writer: Failed to find PM %s in cache for return PMJ event.	La passerelle a reçu un événement retour Problem Management Journal de la passerelle pour un ticket d'incident. Lors d'une tentative de recherche du numéro de ticket d'incident dans le cache du programme d'écriture afin de déterminer le numéro de série de l'alerte associée au ticket, aucun enregistrement n'a été trouvé ou récupéré.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Journal Feedback Failed.	La passerelle n'a pas pu générer l'instruction SQL open action ou envoyer la commande d'action SQL au serveur.	Consultez le fichier journal du serveur ObjectServer.
<i>Nom_passerelle</i> Writer: Failed to find PM %s in cache for return PMC event.	La passerelle a reçu un événement retour Problem Management Close de la passerelle pour un ticket d'incident. Lors d'une tentative de recherche du numéro de ticket d'incident dans le cache du programme d'écriture afin de déterminer le numéro de série de l'alerte associée au ticket, aucun enregistrement n'a été trouvé ou récupéré.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Close Feedback Failed.	La passerelle n'a pas pu générer l'instruction SQL open action ou envoyer la commande d'action SQL au serveur.	Consultez le fichier journal du serveur ObjectServer.
Received error code <code> from Reader/Writer Module - [<message>].	La passerelle a reçu un message d'erreur.	Vérifiez les fichiers journaux du module.
<i>Nom_passerelle</i> Writer: Failed to read <i>nom_passerelle</i> event from <i>nom_passerelle</i> Reader Module.	La passerelle n'a pas pu lire l'événement envoyé par le module de programme de lecture de la passerelle.	Vérifiez les fichiers journaux du programme de lecture.
<i>Nom_passerelle</i> Writer: Received event of type <type d'événement> which was unexpected.	La passerelle a reçu un type d'événement inconnu.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.

Tableau 72. Messages d'erreur de passerelle communs (suite)

Erreur	Description	Action
<i>Nom_passerelle</i> Writer: Received invalid known message from Reader/Writer Module for this system.	La passerelle a reçu un message connu invalide.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Received unknown message from Reader/Writer Module.	La passerelle a reçu un message inconnu invalide.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Failed to block on data feed from <i>nom_passerelle</i> Reader Module.	La passerelle n'a pas pu se bloquer en raison d'une demande d'arrêt. Ce message est affiché pendant l'arrêt de la passerelle.	Consultez votre contrat de support pour savoir comment contacter le centre d'assistance.
<i>Nom_passerelle</i> Writer: Fatal thread termination. Stopping gateway.	Une unité d'exécution a quitté de manière inattendue.	Vérifiez les fichiers journaux de la passerelle.
<nom attribut> attribute is not a string for <i>nom_passerelle</i> writer <i>nom_programme_écriture</i> - IGNORED	Un nom d'attribut n'est pas reconnu. La passerelle l'ignore.	Vérifiez les fichiers journaux de la passerelle.
<nom attribut> attribute must be set to TRUE or FALSE for writer <i>nom_programme_écriture</i> .	Un nom d'attribut n'a pas été défini sur TRUE ou FALSE.	Vérifiez le fichier de configuration de la passerelle.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to shutdown <i>nom_passerelle</i> Reader/Writer Modules.	La passerelle n'a pas pu arrêter les modules des programmes de lecture et d'écriture.	Consultez le fichier journal du module.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to disconnect feedback connection.	Echec de la déconnexion du canal de retour.	Consultez le fichier journal du serveur ObjectServer.
Failed to create <i>nom_passerelle</i> event structure for a problem management open event in writer <i>nom_programme_écriture</i> .	Le programme d'écriture de la passerelle n'a pas pu allouer de structure d'événement de passerelle pour un événement de gestion de problème ouvert en raison de ressources de mémoire insuffisantes.	Essayez de libérer plus de mémoire.
<i>Nom_passerelle</i> Writer: FEEDBACK FAILED!!	La passerelle n'a pas pu stocker le numéro du problème.	Consultez le fichier journal du serveur ObjectServer.
Failed to create journal for <i>nom_passerelle</i> writer <i>nom_programme_écriture</i> (from INSERT)	La passerelle n'a pas pu créer le journal.	Consultez le fichier journal du programme d'écriture.
Failed to create <i>nom_passerelle</i> event structure for a problem management update event in writer <i>nom_programme_écriture</i> .	Le programme d'écriture de la passerelle n'a pas pu allouer de structure d'événement de passerelle pour un événement de mise à jour de gestion de problème en raison de ressources de mémoire insuffisantes.	Essayez de libérer plus de mémoire.
<i>Nom_passerelle</i> Writer <i>nom_programme_écriture</i> : Failed to delete problem ticket from cache for serial <serial number>.	La passerelle n'a pas pu supprimer le numéro de série de la mémoire cache.	Il s'agit d'une erreur interne. Vous pouvez l'ignorer.

Tableau 72. Messages d'erreur de passerelle communs (suite)

Erreur	Description	Action
Failed to create <i>nom_passerelle</i> event structure for a PMC event in writer <i>nom_programme_écriture</i> .	Le programme d'écriture de la passerelle n'a pas pu allouer de structure d'événement de passerelle pour un événement de gestion de problème fermé en raison de ressource mémoire insuffisantes.	Essayez de libérer plus de mémoire.

Annexe C. Expressions régulières

Tivoli Netcool/OMNIBus prend en charge l'utilisation des expressions régulières dans les demandes de recherche que vous réalisez sur le serveur ObjectServer. Les expressions régulières sont des séquences d'*atomes* composées de caractères normaux et de métacaractères.

Un atome est un caractère unique ou un canevas d'un ou de plusieurs caractères entre parenthèses. Les caractères normaux incluent les lettres majuscules et minuscules et les nombres. Les métacaractères sont des caractères non alphabétiques qui possèdent une signification spécifique dans des expressions régulières.

Deux types de bibliothèques d'expressions régulières sont disponibles pour le serveur ObjectServer :

- NETCOOL : cette bibliothèque est utile pour le traitement des caractères à un seul octet.
- TRE : cette bibliothèque permet d'utiliser la syntaxe d'expression régulière étendue POSIX 1003.2 et assure la prise en charge des langues à caractères à un seul octet ou à plusieurs octets. Lorsque le codage UTF-8 est activé sous Windows, seuls les caractères du plan Unicode 0, le BMP (Basic Multilingual Plane), sont pris en charge dans les modèles d'expression régulière. Tous les caractères hors du BMP, qui se trouvent dans le modèle, génèrent une erreur. Les chaînes correspondantes du modèle d'expression régulière peuvent contenir des caractères UTF-8.

Remarque : L'utilisation de la bibliothèque TRE peut donner lieu à une nette diminution des performances du système. Il est possible d'obtenir des performances optimales avec la bibliothèque NETCOOL.

Vous pouvez utiliser la propriété **RegexpLibrary** du serveur ObjectServer pour spécifier la bibliothèque à utiliser pour la correspondance des expressions régulières. La bibliothèque d'expressions régulières NETCOOL est activée par défaut.

Bibliothèque d'expressions régulières NETCOOL

Si votre système prend en charge les langages de caractères codés sur un octet, la bibliothèque d'expressions régulières NETCOOL vous permet d'exécuter des requêtes de recherche sur vos données. La bibliothèque NETCOOL fournit de meilleures performances système que la bibliothèque d'expressions régulières TRE.

Remarque : Lorsqu'une expression régulière est écrite en langage SQL, l'analyseur syntaxique SQL traite d'abord les littéraux chaîne avant de les transmettre à la bibliothèque d'expressions régulières. L'analyseur syntaxique SQL traite les séquences d'échappement avec barres obliques inversées. Par la suite, les barres obliques inversées simples ne sont pas visibles par la bibliothèque d'expressions régulières. Par conséquent, lorsque vous écrivez une expression régulière en langage SQL, utilisez des barres obliques inversées doubles comme caractère d'échappement pour les caractères réservés.

Par exemple, comme caractère d'échappement pour les parenthèses dans la chaîne '1_(22)', utilisez l'expression suivante : '1_\\(22\\)'.

La bibliothèque d'expressions régulières NETCOOL prend en charge l'utilisation de caractères et de métacaractères normaux. Le tableau suivant décrit le jeu de métacaractères pris en charge par la bibliothèque d'expressions régulières NETCOOL.

Tableau 73. Métacaractères

Métacaractère	Description	Exemples
*	Correspond à zéro ou plusieurs instances de l'atome précédent. Correspond à autant d'instances que possible.	goo* correspond à my godness, my goodness et my goodness, mais pas à my gdness.
+	Correspond à une ou plusieurs instances de l'atome précédent. Correspond à autant d'instances que possible.	goo+ correspond à my goodness et my goodness, mais pas à my godness.
?	Correspond à zéro ou plusieurs instances de l'atome précédent.	goo? correspond à my godness, my goodness et my goodness, mais pas à my gdness. colou?r correspond à color et colour. end-?user correspond à enduser et end-user.
\$	Correspond à la fin de la chaîne.	end\$ correspond à the end, mais pas à the ending.
^	Correspond au début de la chaîne.	^severity correspond à severity level 5, mais pas à The severity is 5.
.	Correspond à n'importe quel caractère unique.	b.at correspond à baat, bBat et b4at, mais pas à bat ou bB4at.
[abcd]	Correspond à n'importe quel caractère compris entre les crochets.	[nN][o0] correspond à no, n0, No et N0. gr[ae]y correspond aux deux orthographes du mot 'grey', à savoir gray et grey.
[a-d]	Correspond à n'importe quel caractère compris dans la plage de caractères séparées par un trait d'union (-).	[0-9] correspond à n'importe quel chiffre décimal. [ab3-5] correspond à a, b, 3, 4 et 5. ^[A-Za-z]+\$ correspond à n'importe quelle chaîne contenant uniquement des caractères en minuscules ou en majuscules.
[^abcd] [^a-d]	Correspond à n'importe quel caractère sauf ceux se trouvant entre les crochets ou compris dans la plage de caractères séparés par un tiret (-).	[^0-9] correspond à n'importe quelle chaîne qui ne contient pas de caractères numériques.
()	Indique que les caractères entre parenthèses doivent être traités comme un modèle de caractère.	A(boo)+Z correspond à AbooZ, AboobooZ, et AboobooZ, mais pas à AboZ ou AboooZ. Jan(uary)? correspond à Jan et January.

Tableau 73. Métacaractères (suite)

Métacaractère	Description	Exemples
	Correspond à l'un des atomes se trouvant d'un côté ou de l'autre de la barre verticale.	A(B C)D correspond à ABD et ACD, mais pas à AD, ABCD, ABBD, ou ACCD. (AB CD) correspond à AB et CD, mais pas à ABD et ACD.
\	Indique que le métacaractère suivant doit être traité comme un caractère régulier. Les métacaractères répertoriés dans ce tableau nécessitent un caractère d'échappement avec barre oblique inversée en tant que préfixe pour désactiver leur signification particulière.	* correspond au caractère *. \. correspond au caractère \. \. correspond au caractère .. \[[0-9]*\] correspond à un crochet d'ouverture, suivi de n'importe quel chiffre ou d'un espace, puis d'un crochet de fermeture.

Concepts associés:

«Bibliothèque d'expressions régulières TRE»

Utilisez la bibliothèque d'expressions régulières TRE pour exécuter des demandes de recherche sur les langages à caractère à un seul octet et à plusieurs octets.

Bibliothèque d'expressions régulières TRE

Utilisez la bibliothèque d'expressions régulières TRE pour exécuter des demandes de recherche sur les langages à caractère à un seul octet et à plusieurs octets.

La bibliothèque d'expressions régulières TRE prend en charge l'utilisation de la syntaxe d'expression régulière étendue POSIX 1003.2 sous la forme :

- de métacaractères
- de quantificateurs minimaux ou non gourmands
- d'expressions entre crochets
- de constructions pour le support multiculturel
- de séquences d'échappement avec barres obliques inversées

Restriction : Une nette diminution des performances du système peut être observée lors de l'utilisation de cette bibliothèque.

Référence associée:

«Bibliothèque d'expressions régulières NETCOOL», à la page 345

Si votre système prend en charge les langages de caractères codés sur un octet, la bibliothèque d'expressions régulières NETCOOL vous permet d'exécuter des requêtes de recherche sur vos données. La bibliothèque NETCOOL fournit de meilleures performances système que la bibliothèque d'expressions régulières TRE.

Métacaractères

Les métacaractères sont des caractères non alphabétiques qui possèdent une signification spécifique dans des expressions régulières.

L'ensemble de métacaractères pouvant être utilisé dans des syntaxes d'expressions régulières étendues est le suivant :

* + ? \$ ^ . () | \ {} [

Le tableau suivant décrit tous ces métacaractères, sauf le métacaractère de crochet [. Le métacaractère [permet de construire des expressions entre crochets.

Tableau 74. Métacaractères

Métacaractère	Description	Exemples
*	Correspond à zéro, une ou plusieurs instances de l'atome précédent. Correspond à autant d'instances que possible.	goo* correspond à my godness, my goodness et my goodness, mais pas à my gdness.
+	Correspond à une ou plusieurs instances de l'atome précédent. Correspond à autant d'instances que possible.	goo+ correspond à my goodness et my goodness, mais pas à my godness.
?	Correspond à zéro, une ou plusieurs instances de l'atome précédent.	goo? correspond à my godness, my goodness et my goodness, mais pas à my gdness. colou?r correspond à color et colour. end-?user correspond à enduser et end-user.
\$	Correspond à la fin de la chaîne.	end\$ correspond à the end, mais pas à the ending.
^	Correspond au début de la chaîne. Le métacaractère ^ peut également être utilisé dans des expressions entre crochets.	^severity correspond à severity level 5, mais pas à The severity is 5.
.	Correspond à n'importe quel caractère unique.	b.at correspond à baat, bBat et b4at, mais pas à bat ou bB4at.
()	Indique que les caractères entre parenthèses doivent être traités comme un modèle de caractère.	A(boo)+Z correspond à AbooZ, AboobooZ, et AboobooZ, mais pas à AboZ ou AboooZ. Jan(uary)? correspond à Jan et à January.
	Correspond à l'un des atomes se trouvant d'un côté ou de l'autre de la barre verticale.	A(B C)D correspond à ABD et ACD, mais pas à AD, ABCD, ABBD, ou ACCD. (AB CD) correspond à AB et CD, mais pas à ABD et ACD.

Tableau 74. Métacaractères (suite)

Métacaractère	Description	Exemples
\	Indique que le métacaractère suivant doit être traité comme un caractère régulier. Les métacaractères répertoriés dans cette section nécessitent un caractère d'échappement avec barre oblique inversée en tant que préfixe pour désactiver leur signification particulière. Le métacaractère \ peut également être utilisé pour construire des séquences d'échappement avec barre oblique inversée.	* correspond au caractère *. \. correspond au caractère \. \. correspond au caractère ..
{m , n}	Met en correspondance m instances avec n instances de l'atome précédent, où m est le minimum et n, le maximum. Correspond à autant d'instances que possible. Remarque : m et n sont des entiers décimaux non signés compris entre 0 et 255.	f{1,2}ord correspond à ford et fford. N/{1,3}A correspond à N/A, N//A et N///A, mais pas à NA ou N///A.
{m , }	Met en correspondance m instances ou plus de l'atome précédent.	Z{2,} correspond à deux répétitions ou plus de Z.
{m}	Correspond exactement à m instances de l'atome précédent.	a{3} correspond à aaa. 1{2} correspond à 11.

Référence associée:

«Expressions entre crochets», à la page 350

Les expressions entre crochets permettent de mettre en correspondance un seul caractère ou un élément de classement.

«Séquences d'échappement avec barres obliques inversées», à la page 353

Lors de la construction des expressions régulières, la barre oblique inversée peut être utilisée de manières différentes.

Quantificateur minimal ou non gourmand

Les expressions régulières sont généralement considérées comme *gourmandes* car une expression contenant des répétitions tente de mettre en correspondance autant de caractères que possible. L'astérisque (*), le signe plus (+), le point d'interrogation (?) et les accolades ({}) sont des métacaractères qui exposent un comportement 'répétitif' et tentent de mettre en correspondance autant d'instances que possible.

Pour mettre en correspondance une sous-expression et aussi peu de caractères que possible, un point d'interrogation (?) peut être ajouté à ces métacaractères pour les rendre *minimaux* ou *non gourmands*. Le tableau suivant décrit les quantificateurs non gourmands.

Tableau 75. Quantificateurs minimaux/non gourmands

Quantificateur	Description	Exemples
*?	Correspond à zéro instances ou plus de l'atome précédent. Correspond à aussi peu d'instances que possible.	Prenons la chaîne d'entrée Netcool Tool Library : <ul style="list-style-type: none"> Le premier groupe de <code>^(.*1).*</code> correspond à Netcool Tool . Le premier groupe de <code>^(.*?1).*</code> correspond à Netcool.
+	Correspond à une ou plusieurs instances de l'atome précédent. Correspond à aussi peu d'instances que possible.	Prenons la chaîne d'entrée little : <ul style="list-style-type: none"> <code>.*?l</code> correspond à l. <code>^.+l</code> correspond à littl.
??	Correspond à zéro ou une instance de l'atome précédent. Correspond à aussi peu d'instances que possible.	<code>.*?b</code> correspond à ab de abc, et b de bbb. <code>.?b</code> correspond à ab de abc, et bb de bbb.
{m , n} ?	Met en correspondance m instances avec n instances de l'atome précédent, où m est le minimum et n, le maximum. Correspond à aussi peu d'instances que possible. Remarque : m et n sont des entiers décimaux non signés compris entre 0 et 255.	Prenons la chaîne d'entrée Netcool Tool Cool Fool Library : <ul style="list-style-type: none"> <code>^((.*?ool)*).*</code> correspond à Netcool Tool Cool Fool. <code>^((.*?ool)+).*</code> correspond à Netcool Tool Cool Fool. <code>^((.*?ool)+?) .*</code> correspond à Netcool. <code>^((.*?ool){2,5}).*</code> correspond à Netcool Tool Cool Fool. <code>^((.*?ool){2,5}?) .*</code> correspond à Netcool Tool. <code>^((.*?ool){2,5}) [FL].*</code> correspond à Netcool Tool Cool Fool. <code>^((.*?ool){2,5}?) [FL].*</code> correspond à Netcool Tool Cool.
{m ,} ?	Met en correspondance m instances ou plus de l'atome précédent. Correspond à aussi peu d'instances que possible.	Prenons la chaîne d'entrée Netcool Tool Cool Fool Library : <ul style="list-style-type: none"> <code>^((.*?ool){2,}).*</code> correspond à Netcool Tool Cool Fool. <code>^((.*?ool){2,}?) .*</code> correspond à Netcool Tool. <code>^((.*?ool){2,}) [FL].*</code> correspond à Netcool Tool Cool Fool. <code>^((.*?ool){2,}?) [FL].*</code> correspond à Netcool Tool Cool.

Expressions entre crochets

Les expressions entre crochets permettent de mettre en correspondance un seul caractère ou un élément de classement.

Le tableau suivant décrit comment utiliser les expressions entre crochets.

Tableau 76. Expressions entre crochets

Expression	Description	Exemples
[abcd]	Correspond à n'importe quel caractère compris entre les crochets.	[nN] [oO] correspond à no, nO, No et NO. gr[ae]y correspond aux deux orthographes du mot 'grey', à savoir gray et grey.

Tableau 76. Expressions entre crochets (suite)

Expression	Description	Exemples
[a-d]	Correspond à n'importe quel caractère compris dans la plage de caractères séparées par un trait d'union -).	[0-9] correspond à n'importe quel chiffre décimal. [ab3-5] correspond à a, b, 3, 4 et 5. [0-9]{4} correspond à n'importe quelle chaîne à 4 chiffres. ^[A-Za-z]+\$ correspond à n'importe quelle chaîne contenant uniquement des caractères en minuscules ou en majuscules. \[[0-9]*\] correspond à un crochet d'ouverture, suivi de n'importe quel chiffre ou d'un espace, puis d'un crochet de fermeture.
[^abcd] [^a-d]	Correspond à n'importe quel caractère sauf ceux se trouvant entre les crochets ou compris dans la plage de caractères séparés par un tiret (-).	[^0-9] correspond à n'importe quelle chaîne qui ne contient pas de caractères numériques.
[.ab.]	Correspond à un élément de classement de plusieurs caractères.	[.ch.] correspond à une séquence de classement de plusieurs caractères ch (si le support de langue nationale actuel prend en charge cette séquence de classement).
[=a=]	Met en correspondance tous les éléments de classement possédant le même ordre de tri principal que cet élément, y compris l'élément lui-même.	[=e=] correspond à e et à toutes les variantes de e dans l'environnement local actuel.

Notez les points suivants :

- Le caret (^) a uniquement une signification spécifique lorsqu'il est inclus en tant que premier caractère après le crochet d'ouverture ([). Sinon, il est traité comme un caractère normal.
- Le trait d'union (-) est traité comme un caractère normal uniquement dans les conditions suivantes :
 - Le trait d'union est le premier ou le dernier caractère entre crochets, par exemple [ab-] ou [-xy].
 - Le trait d'union est le seul caractère (le premier et le dernier), c'est-à-dire [-].
- Pour mettre en correspondance un crochet de fermeture et une expression entre crochets, le crochet de fermeture doit être le premier caractères entre les crochets. Par exemple, [] [xy] correspond à], [, x et y.
- Les autres métacaractères sont traités comme des caractère normaux entre crochets et ne doivent pas être ignorés. Par exemple, [ca\$] correspond à c, a ou \$.

Référence associée:

«Métacaractères», à la page 348

Les métacaractères sont des caractères non alphabétiques qui possèdent une signification spécifique dans des expressions régulières.

Constructions pour le support multiculturel

L'ordre de tri des caractères (et de toutes leurs variantes) dépend de l'environnement local, de sorte que des expressions régulières différentes sont généralement obligatoires pour mettre en correspondance des caractères de la même classe, dans différents environnements locaux. Pour faciliter le support multiculturel, un ensemble de noms prédéfinis inclus entre [: et :] permet de représenter des caractères de la même classe.

L'ensemble de noms valides dépend de la valeur de la variable d'environnement LC_CTYPE de l'environnement local actuel, mais certains noms affichés dans le tableau suivant sont valides dans tous les environnements locaux.

Tableau 77. Constructions multiculturelles

Construction	Description
[:alnum:]	Correspond à n'importe quel caractère alphanumérique.
[:alpha:]	Correspond à n'importe quel caractère alphabétique.
[:blank:]	Correspond à n'importe quel caractère blanc - en d'autres termes, espace et tabulation.
[:cntrl:]	Correspond à n'importe quel caractère de commande. Ces caractères sont non imprimables.
[:digit:]	Correspond à n'importe quel chiffre décimal.
[:graph:]	Correspond à n'importe quel caractère imprimable, sauf un espace.
[:lower:]	Correspond à un caractère alphabétique en minuscules.
[:print:]	Correspond à n'importe quel caractère imprimable, même les espaces.
[:punct:]	Correspond à n'importe quel caractère imprimable qui n'est pas un espace ou un caractère alphanumérique, c'est-à-dire, de la ponctuation.
[:space:]	Correspond à n'importe quel caractère blanc.
[:upper:]	Correspond à un caractère alphabétique en majuscules.
[:xdigit:]	Correspond à un chiffre hexadécimal.

Exemple : constructions multiculturelles

[:lower:]AB] correspond aux lettres en minuscules et aux A et B en majuscules.

[:space:] [:alpha:]] correspond à n'importe quel caractère blanc ou alphabétique.

[:alpha:]] correspond à [A-Za-z] dans l'environnement local Anglais (en), mais inclut des lettres accentuées et supplémentaires dans d'autres environnements locaux.

Séquences d'échappement avec barres obliques inversées

Lors de la construction des expressions régulières, la barre oblique inversée peut être utilisée de manières différentes.

La barre oblique inversée (\) peut être utilisée pour :

- Désactiver la signification particulière de métacaractères afin qu'ils puissent être traités comme des caractères normaux.
- Inclure des caractères non imprimables dans une expression régulière.
- Donner une signification spéciale à certains caractères normaux.
- Indiquer des backreferences. Les *backreferences* vous permettent d'indiquer qu'une sous-expression correspondante précédente est de nouveau mise en correspondance ultérieurement.

Remarque : La barre oblique inversée ne peut pas être le dernier caractère d'une expression régulière.

Lorsqu'une expression régulière est écrite en langage SQL, l'analyseur syntaxique SQL traite d'abord les littéraux chaîne avant de les transmettre à la bibliothèque d'expressions régulières. L'analyseur syntaxique SQL traite les séquences d'échappement avec barres obliques inversées. Par la suite, les barres obliques inversées simples ne sont pas visibles par la bibliothèque d'expressions régulières. Par conséquent, lorsque vous écrivez une expression régulière en langage SQL, utilisez des barres obliques inversées doubles comme caractère d'échappement pour les caractères réservés.

Par exemple, comme caractère d'échappement pour les parenthèses dans la chaîne '1_(22)', utilisez l'expression suivante : '1_\\(22\\)'

Le tableau suivant décrit comment indiquer des séquences d'échappement avec barres obliques inversées pour des caractères non imprimables et des backreferences. Ce tableau explique également comment utiliser les séquences d'échappement avec barres obliques inversées pour conférer une signification particulière à certains caractères normaux.

Tableau 78. Séquences d'échappement avec barres obliques inversées

Séquence d'échappement avec barres obliques inversées	Description
\\a	Correspond au caractère de sonnerie (code ASCII 7).
\\e	Correspond au caractère d'échappement (code ASCII 27).
\\f	Correspond au caractère avance page (code ASCII 12).
\\n	Correspond au caractère de retour à la ligne ou de saut de ligne (code ASCII 10).
\\r	Correspond au caractère de retour chariot (code ASCII 13).
\\t	Correspond au caractère de tabulation horizontale (code ASCII 9).
\\v	Correspond au caractère de tabulation verticale.
\\<	Correspond au début d'un mot ou d'un identificateur, défini en tant que limite entre des caractères non alphanumériques et des caractères alphanumériques (y compris les traits de soulignement). Cette séquence ne correspond à aucun caractère, juste au contexte.
\\>	Correspond à la fin d'un mot ou d'un identificateur.

Tableau 78. Séquences d'échappement avec barres obliques inversées (suite)

Séquence d'échappement avec barres obliques inversées	Description
<code>\b</code>	Correspond à une limite de mot. En d'autres termes, cette séquence correspond à la chaîne vide au début ou à la fin d'une séquence alphanumérique. Permet une recherche 'de mots entiers uniquement'.
<code>\B</code>	Correspond à une limite différente d'un mot. En d'autres termes, cette séquence correspond à la chaîne vide ne se trouvant pas au début ou à la fin d'un mot.
<code>\d</code>	Correspond à n'importe quel chiffre décimal. Equivalut à <code>[0-9]</code> et <code>[[:digit:]]</code> .
<code>\D</code>	Correspond à n'importe quel caractère différent d'un chiffre. Equivalut à <code>[^0-9]</code> et <code>[^[:digit:]]</code> .
<code>\s</code>	Correspond à n'importe quel caractère blanc. Equivalut à <code>[\t\n\r\f\v]</code> ou <code>[[:space:]]</code> .
<code>\S</code>	Correspond à n'importe quel caractère différent d'un caractère blanc. Equivalut à <code>[^ \t\n\r\f\v]</code> et <code>[^[:space:]]</code> .
<code>\w</code>	Correspond à un caractère de mot. En d'autres termes, n'importe quel caractère alphanumérique ou trait de soulignement. Equivalut à <code>[a-zA-Z0-9_]</code> ou <code>[[:alnum:]]</code> .
<code>\W</code>	Correspond à n'importe quel caractère non alphanumérique. Equivalut à <code>[^a-zA-Z0-9_]</code> ou <code>[^[:alnum:]]</code> .
<code>\[1-9]</code>	Une barre oblique inversée suivie d'un seul chiffre décimal n différent de zéro est appelée une <i>backreference</i> . Cette séquence correspond au même jeu de caractères que celui mis en correspondance par la sous-expression n entre parenthèses.

Exemple de constructions avec des barres obliques inversées

`\bcat\b` correspond à `cat` mais pas à `cats` ou `bobcat`.

`\d\s` correspond à un chiffre suivi d'un caractère blanc.

`\d\s` correspond à n'importe quel chiffre ou caractère blanc.

`.[XY]).[XY].` correspond à `aXbXc` et à `aYbYc`, mais également à `aXbYc` et `aYbXc`. Néanmoins, `.[XY]).\1.` ne correspondra qu'à `aXbXc` et `aYbYc`.

Référence associée:

«Métacaractères», à la page 348

Les métacaractères sont des caractères non alphabétiques qui possèdent une signification spécifique dans des expressions régulières.

Annexe D. Tables et types de données ObjectServer

Cet annexe contient des informations relatives à la table de base de données Object Server.

Table alerts.status

La table alerts.status contient des informations de statut sur les problèmes détectés par les sondes.

Remarque : Vous pouvez afficher uniquement les colonnes de type CHAR, VARCHAR, INCR, INTEGER et TIME dans la liste d'événements. N'ajoutez pas de colonnes d'un autre type dans la table alerts.status.

Le tableau suivant décrit les colonnes de la table alerts.status.

Tableau 79. Colonnes de la table alerts.status

Nom de colonne	Type de données	Obligatoire	Description
Identifiant	varchar(255)	Oui	<p>Contrôle le dédoublement du serveur ObjectServer. La zone Identificateur contrôle la fonction de dédoublement du serveur ObjectServer et prend également en charge la compatibilité avec l'automatisation GenericClear en s'assurant que les événements de résolution sont correctement insérés dans le serveur ObjectServer et qu'ils ne sont pas dédoublement avec leurs événements de problème respectifs.</p> <p>L'identificateur ci-dessous identifie correctement les événement répétés dans un environnement type :</p> <pre>@Identifiant=@Node+ " "+@AlertKey+" "+@AlertGroup+ " "+@Type+ " "+@Agent+ "+@Manager</pre> <p>Il est possible que l'ajout d'informations supplémentaires à la zone Identificateur soit nécessaire pour s'assurer que le dédoublement et la compatibilité avec l'automatisation GenericClear sont corrects. Par exemple, lorsqu'une interruption caractéristique du protocole SNMP contient une valeur d'énumération de statut dans une de ses liaisons variables, le numéro d'interruption spécifique et la valeur de la varbind appropriée doivent être ajoutés à la zone Identificateur de la manière suivante :</p> <pre>@Identifiant=@Node + " + @AlertKey+" "+@AlertGroup+ " "+@Type+ " "+@Agent+ "+@Manager+ " "+\$specific-trap+ "+\$2</pre>
Serial	incr	Oui	Numéro de série Tivoli Netcool/OMNIBus de la ligne.
Node	varchar(64)	Oui	<p>Identifie l'entité gérée à partir de laquelle provient l'alarme. Il peut s'agir d'un nom d'hôte ou d'unité, d'un nom de service ou d'une autre entité.</p> <p>Pour les unités réseau IP ou les hôtes, la colonne Node contient le nom résolu de l'unité ou de l'hôte. Si le nom ne peut pas être résolu, la colonne Node doit contenir l'adresse IP de l'hôte ou de l'unité.</p> <p>En ce qui concerne les périphériques réseau et les hôtes non IP, les alarmes doivent contenir des informations similaires à celles de l'unité ou de l'hôte IP. En d'autres termes, la colonne Node doit contenir le nom de l'unité ou de l'hôte qui autorise une communication directe ou qui peut être résolu pour autoriser une communication directe avec l'unité ou l'hôte.</p>

Tableau 79. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
NodeAlias	varchar(64)	Non	<p>Alias du noeud. Pour les périphériques réseau ou les hôtes, il doit s'agir de l'adresse logique (couche 3) de l'entité. Pour les périphériques IP ou les hôtes, il doit s'agir de l'adresse IP.</p> <p>En ce qui concerne les unités ou les hôtes non IP, plusieurs schémas d'adressage sont possibles. Lorsque vous sélectionnez une valeur pour la zone NodeAlias, cette valeur devrait autoriser la communication directe avec l'unité ou l'hôte. Par exemple, une unité gérée par TL-1. La zone NodeAlias peut être remplie par une table de consultation ou une politique Netcool/Impact, avec l'adresse IP et le numéro de port du serveur de terminaux via lesquels l'unité TL-1 peut être atteinte.</p>
Manager	varchar(64)	Oui	<p>Nom descriptif de la sonde qui a collecté et réacheminé l'alarme vers le serveur ObjectServer. Il permet également d'indiquer l'hôte sur lequel la sonde s'exécute. Idéalement, il est défini dans le fichier de propriétés de la sonde. Cependant, le fichier de règles devrait vérifier qu'il est défini correctement et le modifier si nécessaire.</p> <p>Par exemple, la syntaxe suivante peut être utilisée pour définir la zone Manager : @Manager="MTTrapd Probe on" + hostname()</p>
Agent	varchar(64)	Non	<p>Nom descriptif du sous-gestionnaire qui a généré l'alerte.</p> <p>Les sondes qui traitent les interruptions SNMP doivent définir la zone Agent sur le nom du fournisseur ou sur le corps des valeurs standard qui ont défini l'interruption. Elles doivent également fournir une description de la base d'informations de gestion ou de son nom de définition, où l'interruption est définie. Cela doit se présenter au format suivant : vendor-MIB description</p> <p>Par exemple : Cisco-Accounting Control, Cisco-Health Monitor, IETFBRIDGEMIB, ATMF-ATM-FORUM-MIB</p> <p>Facultativement, les informations spécifiques au fournisseur, par exemple les nombres modèles des unités, peuvent être ajoutées à la zone Agent pour les implémentations spécifiques au fournisseur des interruptions standards.</p> <p>La sonde Syslog devrait définir la zone Agent sur le nom du fournisseur qui a défini le message reçu. Elle devrait également fournir toute description logique de la famille de messages à laquelle le message reçu appartient.</p> <p>Par exemple, Cisco définit les messages reçus des unités basées sur IOS dans la documentation séparée des messages reçus du Firewall PIX. Le format des messages est également légèrement différent. Par conséquent, la zone Agent des messages Syslog reçus de Cisco sera définie sur Cisco-IOS ou Cisco- PIX Firewall.</p> <p>L'instance de TSM TL-1 devrait définir la zone Agent sur le nom du fournisseur qui a défini le message reçu. Elle devrait également fournir toute description logique de la famille de messages à laquelle le message reçu appartient.</p>
AlertGroup	varchar(255)	Non	<p>Nom descriptif du type de panne indiqué par l'alerte. Par exemple : Interface Status ou CPU Utilization).</p> <p>La zone AlertGroup doit contenir la même valeur pour les événements de problème connexe et de résolution.</p> <p>Par exemple, l'interruption SNMP 2 (linkDown) et l'interruption SNMP 3 (linkUp) doivent toutes deux contenir la même valeur AlertGroup de Link Status.</p> <p>La zone AlertGroup d'un message TL-1 sera définie sur la valeur du type d'alarme du message.</p>

Tableau 79. Colonnes de la table *alerts.status* (suite)

Nom de colonne	Type de données	Obligatoire	Description
AlertKey	varchar(255)	Oui	<p>Clé descriptive qui indique l'instance d'objet géré référencée par l'alerte. Par exemple, la partition de disque indiquée par une alerte complète du système de fichiers ou le port de commutation indiqué par une alerte d'utilisation.</p> <p>Les sondes traitant les interruptions SNMP devraient définir la zone AlertKey sur l'une des valeurs suivantes (par ordre de préférence) :</p> <ul style="list-style-type: none"> Instance SNMP de l'objet géré représentée par l'alarme. Pour l'obtenir, il faut normalement extraire l'instance de l'ID objet d'une des liaisons variables de l'interruption. De plus, il est possible qu'elle fasse également partie d'une combinaison composée d'une ou plusieurs valeurs de liaisons variables de l'interruption. Par exemple, la première liaison variable d'une interruption linkDown contient la valeur ifIndex (numéro d'interface) de l'interface qui a échoué. AlertKey peut être définie de deux manières : <ul style="list-style-type: none"> @AlertKey = extract(\$OID1, "\.[0-9]+\$") @AlertKey = \$1 Description textuelle de l'instance dérivée du nom ou de la description de l'interruption. Par exemple, une unité ayant deux alimentations électriques (A et B) peut envoyer deux interruptions caractéristiques séparées, sans liaisons variables, pour indiquer le statut en échec de l'une de ses alimentations électriques. L'instance d'alimentation électrique appropriée aurait besoin d'être dérivée des définitions d'interruption de la base d'informations de gestion, puis encodée dans le fichier de règles : <pre>switch(\$specific-trap) { case "1": @AlertKey = "A" case "2": @AlertKey = "B" default: }</pre> Combinaison mélangée de valeurs de liaisons variables et d'informations dérivées du nom ou de la description de l'interruption. Toute information sur l'instance indisponible pour les deux valeurs précédentes alors qu'elle est obligatoire pour s'assurer que le dédoublonnage et la compatibilité GenericClear sont corrects, est appropriée. <p>La sonde Syslog devrait définir AlertKey sur une description textuelle de l'instance dérivée du texte contenu dans le message de journal. Idéalement, il s'agit d'un nom textuel de la même entité gérée. Par exemple :</p> <pre>Nov 20 13:12:57 device.customer.net 195.180.208.193 19986: 37w0d: %LINK-3-UPDOWN: Interface FastEthernet0/13, changed state to down</pre> <p>Dans cet exemple, AlertKey serait défini sur FastEthernet0/13 à l'aide de la syntaxe suivante :</p> <pre>@AlertKey = extract(\$Details, "Interface (.*), changed")</pre> <p>En règle générale, la zone AlertKey d'un message TL-1 est définie sur la valeur de l'emplacement de l'alarme du message.</p>

Tableau 79. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
Severity	integer	Oui	<p>Indique le niveau de gravité de l'alerte, qui indique comment la capacité perçue de l'objet géré a été affectée. La couleur de l'alerte dans la liste d'événements est contrôlée par la valeur de gravité :</p> <p>0 : Supprimer. Le niveau de gravité Supprimer indique qu'une ou plusieurs alarmes précédemment signalées ont été effacées. Les alarmes ont été supprimées soit manuellement par un opérateur réseau, soit automatiquement par un processus qui a déterminé que la situation d'erreur n'est plus d'actualité. Les processus automatiques, par exemple les processus GenericClear Automation, suppriment généralement toutes les alarmes d'un objet géré (AlertKey) qui ont le même type d'alarme et/ou la même cause probable (groupe d'alertes).</p> <p>1 : Indéterminé. Le niveau de gravité Indéterminé signale que le niveau de gravité n'a pas pu être déterminé. De plus, toute alarme solutionnant un problème est initialement définie comme indéterminée jusqu'à ce qu'elle ait été corrélée avec une alarme signalant un problème (par exemple, via GenericClear Automation), lorsque toutes les alarmes corrélées sont définies sur Supprimer.</p> <p>2 : Avertissement. Le niveau de gravité Avertissement signale la détection d'erreurs potentielles ou imminentes ayant des conséquences sur le service. Si nécessaire, effectuez une recherche plus poussée sur la nature de l'erreur pour éviter que celle-ci ne s'aggrave.</p> <p>3 : Mineur. Le niveau de gravité Mineur signale l'existence d'une situation d'erreur qui n'a aucune conséquence sur le service. Il est recommandé de prendre des actions correctives pour éviter que cette situation ne s'aggrave. Ce niveau de gravité peut être signalé lorsque, par exemple, la condition de l'alarme détectée ne dégrade actuellement pas la capacité de l'objet géré.</p> <p>4 : Majeur. Le niveau de gravité Majeur signale qu'une condition affectant le service s'est développée et qu'une action corrective doit être prise de toute urgence. Ce niveau de gravité peut être signalé lorsqu'il y a, par exemple, une dégradation sévère de la capacité de l'objet géré et que ces capacités doivent être restaurées dans leur totalité.</p> <p>5 : Critique. Le niveau de gravité Critique signale la présence d'une condition affectant le service et recommande qu'une action corrective soit prise de toute urgence. Ce niveau de gravité peut être signalé lorsque, par exemple, un objet géré est hors service et que ses capacités doivent être restaurées.</p>

Tableau 79. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
Summary	varchar(255)	Oui	<p>Contient du texte qui décrit la condition de l'alarme et l'instance de l'objet géré affectée.</p> <ul style="list-style-type: none"> Assurez-vous que les informations présentées dans la zone Summary sont concises mais suffisamment détaillées. La zone Summary doit contenir, entre parenthèses, une description de l'instance de l'objet géré fournie par les données de l'alarme disponibles. Par exemple, une interruption linkDown à partir d'une unité Cisco contiendra la valeur <code>ifDescr</code> dans la seconde liaison variable. Le texte récapitulatif d'un tel événement serait similaire à : <code>"Link Down (FastEthernet0/13)"</code> Pour les alarmes qui sont en relation avec des seuils contenant les valeurs comparées ou de seuil, sélectionnez un des formats suivants en vous basant sur les données disponibles : <ul style="list-style-type: none"> Aucune valeur fournie : <code>"Link Utilization High (BRI2/0:1)"</code> Nom de valeur comparée fourni : <code>"Link Utilization High: inOctets Exceeded Threshold (BRI2/0:1)"</code> Nom de valeur comparée et valeur fournies : <code>"Link Utilization High: inOctets, 7100, Exceeded Threshold (BRI2/0:1)"</code> Nom de seuil fourni : <code>"Link Utilization High: inOctetsMax Exceeded (BRI2/0:1)"</code> Valeur de seuil fournie : <code>"Link Utilization High: inOctetsMax, 7000, Exceeded (BRI2/0:1)"</code> Valeur comparée et valeur de seuil fournies : <code>"Link Utilization High: 7100 Exceeded 7000 (BRI2/0:1)"</code> Noms et valeurs fournis : <code>"Link Utilization High: inOctets, 7100, Exceeded inOctetsMax,7000 (BRI2/0:1)"</code>
StateChange	time	Oui	Horodatage du serveur ObjectServer géré automatiquement qui indique la dernière insertion ou modification de l'alerte à partir de n'importe quelle source.
FirstOccurrence	time	Oui	Heure en secondes (à partir du 1er janvier 1970 à minuit) indiquant l'heure de création de cette alerte ou le début de l'interrogation au niveau de la sonde.
LastOccurrence	time	Oui	Heure de dernière mise à jour de cette alerte au niveau de la sonde.
InternalLast	time	Oui	Heure de dernière mise à jour de cette alerte au niveau du serveur ObjectServer.
Poll	integer	Non	Fréquence d'interrogation de cette alerte en secondes.

Tableau 79. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
Type	integer	Non	<p>Le type d'alarme, où "type" fait référence à l'état de problème ou de résolution de l'alarme. Cette zone est importante pour assurer la corrélation correcte des événements à l'aide de GenericClear Automation. Les valeurs suivantes sont valides pour la zone Type :</p> <p>0 : Type non défini</p> <p>1 : Incident</p> <p>2 : Résolution</p> <p>3 : Incident Netcool/Visionary</p> <p>4 : Résolution Netcool/Visionary</p> <p>7 : Nouvelle alarme Netcool/ISMs</p> <p>8 : Ancienne alarme Netcool/ISMs</p> <p>11 : Plus grave</p> <p>12 : Moins grave</p> <p>13 : Information</p> <p>Certains scénarios ne peuvent être catégorisés ni comme Problème, ni comme Résolution. Par exemple, les événements dont le niveau de gravité augmente mais qui ne représentent pas actuellement un échec et les événements dont le niveau de gravité diminue mais qui ne signalent pas actuellement que l'échec a été complètement résolu. Dans ce cas, la zone Type doit être définie sur Problème, Plus grave ou Moins grave pour entretenir la compatibilité avec GenericClear Automation.</p> <p>Par exemple, la logique de fichier de règles suivante est utilisée pour gérer des interruptions associées à l'état de connexion homologue BGP :</p> <pre>switch (\$bgpPeerState) { case "1": ### inactif @Severity = 4 @Type = 1 case "2": ### connecté @Severity = 2 @Type = 12 case "3": ### actif @Severity = 2 @Type = 12 case "4": ### ouvert envoyé @Severity = 2 @Type = 12 case "5": ### confirmé ouvert @Severity = 2 @Type = 12 case "6": ### établi @Severity = 1 @Type = 2 default: @Severity = 2 @Type = 1 }</pre>
Tally	integer	Oui	Comptage géré automatiquement du nombre d'insertions et de mises à jour de l'alerte à partir de n'importe quelle source. Ce comptage est affecté par le dédoublement.
Class	integer	Oui	Classe d'alerte utilisée pour identifier la sonde ou le fournisseur à partir desquels l'alerte a été générée. Contrôle l'applicabilité d'outils contextuels de liste d'événements.
Grade	integer	Non	<p>Indique l'état d'escalade de l'alerte :</p> <p>0 : Non escaladée</p> <p>1 : Escaladée</p>
Location	varchar(64)	Non	Indique l'emplacement physique de l'unité, de l'hôte ou du service pour lesquels l'alerte a été générée.
OwnerUID	integer	Oui	ID utilisateur de l'utilisateur en charge de la gestion de cette alerte. La valeur par défaut est 65534, soit l'identificateur de l'utilisateur nobody.

Tableau 79. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
OwnerGID	integer	Non	ID groupe du groupe en charge de la gestion de cette alerte. La valeur par défaut est 0, soit l'identificateur de l'utilisateur public.
Acknowledged	integer	Oui	Indique si l'alerte a un accusé réception : 0 : Non 1 : Oui Il est possible d'accuser réception des alertes manuellement via un opérateur de réseau ou automatiquement via un processus de corrélation ou de flux de travaux.
Flash	integer	Non	Active l'option de clignotement de la liste d'événements.
EventId	varchar(255)	Non	ID événement (par exemple, SNMPTRAP-link down). Plusieurs événements peuvent avoir le même ID événement. L'événement ID est rempli par le fichier de règles de la sonde et utilisé par IBM Tivoli Network Manager IP Edition.
ExpireTime	integer	Oui	Nombre de secondes écoulées entre le moment où cette alerte a été détectée la dernière fois par le serveur ObjectServer (LastOccurrence) et le moment où elle a été automatiquement supprimée. Utilisé par l'automatisation Temps d'expiration.
ProcessReq	integer	Non	Indique si l'alerte doit être traitée par IBM Tivoli Network Manager IP Edition. Cette colonne est renseignée par le fichier de règles de la sonde et utilisée par IBM Tivoli Network Manager IP Edition.
SuppressEscl	integer	Oui	Utilisée pour supprimer ou escalader l'alerte : 0 : Normal 1 : Escaladée 2 : Niveau d'escalade 2 3 : Niveau d'escalade 3 4 : Supprimée 5 : Masquée 6 : Maintenance Le niveau de suppression est sélectionné manuellement par les opérateurs de la liste d'événements.
Customer	varchar(64)	Non	Nom du client affecté par cette alerte.
Service	varchar(64)	Non	Nom du service affecté par cette alerte.
PhysicalSlot	integer	Non	Numéro d'emplacement indiqué par l'alerte.
PhysicalPort	integer	Non	Numéro de port indiqué par l'alerte.
PhysicalCard	varchar(64)	Non	Numéro de carte ou description indiqué(e) par l'alerte.
TaskList	integer	Oui	Indique si un utilisateur a ajouté l'alerte à la liste de tâches : 0 : Non 1 : Oui Les opérateurs peuvent ajouter des alertes à la liste de tâches de la liste d'événements.
NmosSerial	varchar(64)	Non	Numéro de série de l'événement qui supprime l'événement en cours. Colonne renseignée par IBM Tivoli Network Manager IP Edition.
NmosObjInst	integer	Non	Colonne renseignée par IBM Tivoli Network Manager IP Edition lors du traitement de l'alerte.
NmosCauseType	integer	Non	Etat de l'alerte. Cette colonne est renseignée par IBM Tivoli Network Manager IP Edition sous forme de valeur entière : • 0 : Inconnue • 1 : Cause première • 2 : Symptôme

Tableau 79. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
Nmos Domain Name	varchar(64)	Non	Nom du domaine d'IBM Tivoli Network Manager IP Edition qui gère l'événement. Par défaut, cette colonne est renseignée uniquement pour les événements générés par des sondages d'IBM Tivoli Network Manager IP Edition. Pour remplir cette colonne pour d'autres sources d'événement que des sondes, vous devez modifier les fichiers de règles.
Nmos EntityId	integer	Non	ID numérique unique qui identifie l'entité de topologie d'IBM Tivoli Network Manager IP Edition à laquelle l'événement est associé. Cette colonne ressemble à la colonne NmosObjInst, mais est plus granulaire. Par exemple, la valeur NmosEntityId peut représenter l'ID d'une interface dans une unité.
Nmos Managed Etat	integer	Non	Statut géré de l'entité de réseau pour laquelle l'événement a été mis au premier plan. Cette colonne peut s'appliquer à des événements d'IBM Tivoli Network Manager IP Edition et de n'importe quelle sonde. Elle permet de filtrer les événements d'interfaces qui ne sont pas considérés comme pertinents.
NmosEvent Map	varchar(64)	Non	Contient la version 3.9 ou une version ultérieure d'IBM Tivoli Network Manager IP Edition, le nom eventMap et la priorité facultative de l'événement qui indique comment IBM Tivoli Network Manager IP Edition doit traiter l'événement. Le numéro de priorité facultatif peut être concaténé à la fin de la valeur, suivi par un point (.). Si la priorité n'est pas indiquée, elle est définie sur 0. Les exemples suivants montrent la configuration d'un mappage d'événement avec une priorité explicite de 900, et d'un autre mappage où la priorité est définie par défaut sur 0 : <ul style="list-style-type: none">• ItnmLinkdownIfIndex.900• PrecisionMonitorEvent
LocalNode Alias	varchar(64)	Oui	Alias de l'entité de réseau indiquée par l'alerte. Pour les périphériques réseau ou les hôtes, il s'agit de l'adresse logique (couche 3) de l'entité ou d'une autre adresse logique qui permet une communication directe avec le périphérique. A utiliser lors de l'identification de l'instance d'un objet géré.
LocalPriObj	varchar(255)	Non	Objet principal référencé par l'alerte. A utiliser lors de l'identification de l'instance d'un objet géré.
LocalSecObj	varchar(255)	Non	Objet secondaire référencé par l'alerte. A utiliser lors de l'identification de l'instance d'un objet géré.
LocalRoot Obj	varchar(255)	Oui	Objet qui équivaut à l'objet principal référencé dans l'alarme. A utiliser lors de l'identification de l'instance d'un objet géré.
Remote Node Alias	varchar(64)	Oui	Adresse réseau de l'entité réseau distante. A utiliser lors de l'identification de l'instance d'un objet géré.
RemotePri Obj	varchar(255)	Non	Objet principal d'une entité de réseau distante référencée par une alarme. A utiliser lors de l'identification de l'instance d'un objet géré.
RemoteSec Obj	varchar(255)	Non	Objet secondaire d'une entité de réseau distante référencée par une alarme. A utiliser lors de l'identification de l'instance d'un objet géré.
Remote RootObj	varchar(255)	Oui	Objet qui équivaut à l'objet principal de l'entité distante référencée dans l'alarme. A utiliser lors de l'identification de l'instance d'un objet géré.

Tableau 79. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
X733 EventType	integer	Non	Indique le type d'alerte : 0 : Non défini 1 : Communications 2 : Qualité de service 3 : Erreur de traitement 4 : Equipement 5 : Environnemental 6 : Violation d'intégrité 7 : Violation opérationnelle 8 : Violation physique 9 : Violation de service de sécurité 10 : Violation de domaine horaire
X733 Probable Cause	integer	Non	Indique la cause probable de l'alerte.
X733 Specific Prob	varchar(64)	Non	Identifie des informations supplémentaires relatives à la cause potentielle de l'alerte. Cette colonne est utilisée par les fichiers de règles des sondes pour indiquer un ensemble d'identificateurs à utiliser lors de l'identification de l'instance d'un objet géré.
X733 CorrNotif	varchar(255)	Non	Liste de toutes les notifications auxquelles cette notification est liée.
ServerName	varchar(64)	Oui	Nom du serveur ObjectServer d'origine. Cette colonne est utilisée par les passerelles pour contrôler la propagation des alertes entre les serveurs ObjectServer.
ServerSerial	integer	Oui	Numéro de série de l'alerte sur le serveur ObjectServer d'origine (si elle ne provient pas de cet ObjectServer). Cette colonne est utilisée par les passerelles pour contrôler la propagation des alertes entre les serveurs ObjectServer.
URL	varchar(1024)	Non	Adresse URL facultative qui fournit un lien vers des informations supplémentaires dans l'unité du fournisseur ou l'ENMS.
Extended Attr	varchar(4096)	Non	Contient des paires nom-valeur (d'attributs étendus Tivoli Enterprise Console) ou d'autres informations supplémentaires pour lesquelles il n'existe aucune colonne dédiée dans la table alerts.status. N'utilisez cette colonne que via les fonctions SQL nvp_get, nvp_set et nvp_exists. Exemple de chaîne nom-valeur : Region="EMEA";host="sf01392w"; Error="errno=32: "Broken pipe" Dans cet exemple, l'attribut Region est défini sur la valeur EMEA, l'attribut host, sur la valeur sf01392w et l'attribut Error, sur la valeur errno=32: "Broken pipe". Notez que les guillemets sont ignorés en les dédoublant, comme indiqué dans la valeur de l'attribut Error. Dans les paires nom-valeur, la valeur est toujours incluse entre guillemets (" ") et les guillemets imbriqués sont ignorés en les dédoublant. Le séparateur entre les paires nom-valeur est un point-virgule (;). Aucun blanc n'est autorisé avant ou après le signe égal (=) ou un point-virgule. Remarque : La colonne peut uniquement contenir 4 096 octets. Le nombre de caractères sera donc inférieur à 4 096 si des caractères codés sur plusieurs octets sont utilisés.

Tableau 79. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
OldRow	integer	Non	<p>Conserve l'état local de la ligne dans chaque ObjectServer lors de la resynchronisation dans une paire de reprise en ligne. Cette colonne ne doit pas être ajoutée aux fichiers de mappage de la passerelle.</p> <p>La valeur de la colonne OldRow est modifiée en 1 dans le serveur ObjectServer de destination pour la durée de la resynchronisation si la propriété Gate.ResyncType de la passerelle est définie sur Minimal.</p> <p>La valeur par défaut est 0.</p>
ProbeSubSecondId	integer	Non	<p>Pour ces alertes que la sonde envoie dans le même intervalle d'une seconde et qui ont, par conséquent, la même valeur LastOccurrence, une valeur incrémentielle commençant à 1 est ajoutée à la zone ProbeSubSecondId pour différencier l'heure de dernière occurrence. La valeur par défaut est 0.</p>
MasterSerial	integer	Non	<p>Identifie le serveur ObjectServer maître si cette alerte est traitée dans un environnement du serveur ObjectServer.</p> <p>Cette colonne est ajoutée lorsque vous exécutez l'utilitaire d'initialisation de la base de données nco_dbinit avec l'option -desktopserver.</p> <p>Remarque : MasterSerial doit être la dernière colonne de la table alerts.status si vous utilisez un environnement du serveur ObjectServer.</p>
BSM_Identity	varchar(1024)	Non	<p>L'identificateur unique de la ressource à partir duquel l'événement est généré est utilisé pour établir une corrélation entre l'événement et cette ressource dans IBM Tivoli Business Service Manager (TBSM).</p>

Table alerts.details

La table alerts.details contient les attributs détaillés des alertes dans le système.

Le tableau suivant décrit les colonnes de la table alerts.details.

Tableau 80. Colonnes de la table alerts.details

Nom de colonne	Type de données	Description
KeyField	varchar(255)	<p>Chaîne d'organisation interne permettant une unicité.</p> <p>La valeur Keyfield est composée d'une valeur Identifieur, de quatre chiffres et d'un numéro de séquence commençant par 1, par exemple :</p> <p><i>Identifieur####1</i></p> <p>où <i>Identifieur</i> est un type de données de varchar(255), utilisé pour relier les détails aux entrées de la table alerts.status.</p> <p>Si la valeur Identifieur dépasse une certaine longueur, il est possible que la valeur Keyfield dépasse sa limite définie de 255 caractères, ce qui provoquerait une troncature du numéro de séquence. Les valeurs Keyfield ne seraient par conséquent plus uniques et la duplication non voulue pourrait provoquer l'échec des insertions dans la table alerts.details.</p> <p>Conseil : Pour éviter un débordement de KeyField (et assurer une unicité), la longueur de la valeur Identifieur doit être suffisamment inférieure à 255 pour permettre l'ajout des quatre chiffres et d'un numéro de séquence (d'un ou plusieurs chiffres).</p>

Tableau 80. Colonnes de la table alerts.details (suite)

Nom de colonne	Type de données	Description
Identifieur	varchar(255)	Identificateur permettant de lier les détails aux entrées dans la table alerts.status. Cette valeur permet de calculer la valeur Keyfield et doit être obligatoirement inférieure à une certaine longueur pour assurer l'unicité de chaque valeur Keyfield calculée. Pour obtenir des instructions sur la longueur maximale de la valeur Identifieur, voir l'astuce dans la ligne KeyField précédente.
AttrVal	integer	Booléen, lorsque cette valeur est définie sur False (0), seule la colonne Detail est valide. Sinon, les colonnes Name et Detail sont toutes deux valides.
Sequence	integer	Numéro de séquence utilisé pour trier les entrées dans la fenêtre de la liste d'événements Données sur l'événement.
Name	varchar(255)	Nom de l'attribut stocké dans la colonne Detail.
Detail	varchar(255)	Valeur d'attribut.

Table alerts.journal

La table alerts.journal fournit un historique des travaux effectués sur les alertes.

Le tableau suivant décrit les colonnes de la table alerts.journal.

Tableau 81. Colonnes de la table alerts.journal

Nom de colonne	Type de données	Description
KeyField	varchar(255)	Clé principale de la table.
Serial	integer	Numéro de série de l'alerte à laquelle est liée cette entrée de journal.
UID	integer	Identificateur de l'utilisateur qui a fait cette entrée.
Chrono	time	Heure et date d'exécution de cette entrée.
Text1	varchar(255)	Premier bloc de texte de l'entrée de journal.
Text2	varchar(255)	Deuxième bloc de texte de l'entrée de journal.
Text3	varchar(255)	Troisième bloc de texte de l'entrée de journal.
Text4	varchar(255)	Quatrième bloc de texte de l'entrée de journal.
Text5	varchar(255)	Cinquième bloc de texte de l'entrée de journal.
Text6	varchar(255)	Sixième bloc de texte de l'entrée de journal.
Text7	varchar(255)	Septième bloc de texte de l'entrée de journal.
Text8	varchar(255)	Huitième bloc de texte de l'entrée de journal.
Text9	varchar(255)	Neuvième bloc de texte de l'entrée de journal.
Text10	varchar(255)	Dixième bloc de texte de l'entrée de journal.
Text11	varchar(255)	Onzième bloc de texte de l'entrée de journal.
Text12	varchar(255)	Douzième bloc de texte de l'entrée de journal.
Text13	varchar(255)	Treizième bloc de texte de l'entrée de journal.
Text14	varchar(255)	Quatorzième bloc de texte de l'entrée de journal.
Text15	varchar(255)	Quinzième bloc de texte de l'entrée de journal.
Text16	varchar(255)	Seizième bloc de texte de l'entrée de journal.

Table service.status

La table service.status permet de contrôler les fonctions supplémentaires obligatoires pour la prise en charge d'IBM Tivoli Composite Application Manager for Internet Service Monitoring.

Le tableau suivant décrit les colonnes de la table service.status.

Tableau 82. Colonnes de la table service.status

Nom de colonne	Type de données	Description
Name	varchar(255)	Nom du service.
CurrentState	integer	Indique l'état du service : 0 : Bon 1 : Mauvais 2 : Marginal 3 : Inconnu
StateChange	time	Indique la dernière fois où l'état de service a été modifié.
LastGoodAt	time	Indique la dernière fois où le service était à l'état Bon (0).
LastBadAt	time	Indique la dernière fois où le service était à l'état Mauvais (1).
LastMarginalAt	time	Indique la dernière fois où le service était à l'état Marginal (2).
LastReportAt	time	Heure du dernier rapport d'état de service.

Table registry.probes

La table registry.probes est utilisée pour effectuer le suivi des informations d'exécution dynamiques relatives aux sondes. Lorsqu'une sonde se connecte au serveur ObjectServer, elle enregistre des informations la concernant dans la table registry.probes. La sonde contrôle les données entrées dans la table.

Dans Tivoli Netcool/OMNIbus version 7.4, la table registry.probes est une table virtuelle. Dans la version 8.1, tous les nouveaux serveurs ObjectServer utilisent une table registry.probes persistante. Les tables virtuelles existantes sont transformées en tables persistantes quand vous appliquez la mise à jour de schéma de serveur ObjectServer `update74to81.sql`.

Remarque : Toutes les données contenues dans la table virtuelle registry.probes existante sont perdues lors de cette mise à jour.

Si vous possédez plusieurs instances d'une sonde s'exécutant sur un ordinateur et si chaque instance possède le même nom, seule une instance sera enregistrée dans la table registry.probes. Pour permettre l'enregistrement de toutes les instances d'une sonde s'exécutant sur le même ordinateur, vous devez utiliser des valeurs uniques pour la propriété **Name** de chaque sonde.

Si une sonde ne parvient pas à redémarrer après s'être précédemment connectée au serveur ObjectServer, son état n'est pas mis à jour dans la table registry.probes. La table continue de représenter l'état de la sonde au moment de sa dernière connexion au serveur ObjectServer.

Le tableau suivant décrit les colonnes de la table registry.probes :

Tableau 83. Colonnes de la table registry.probes

Nom de colonne	Type de données	Description
ApiReleaseID	VARCHAR(20)	Identificateur d'édition de la bibliothèque de l'API de la sonde.
ApiVersion	VARCHAR(20)	Version de la bibliothèque de l'API de la sonde.
ConnectionID	entier	<p>ID connexion affecté à la sonde lorsqu'elle se connecte au serveur ObjectServer. Il correspond à l'ID connexion stocké dans la table catalog.connections. Cette colonne est remplie par le déclencheur ObjectServer registry_new_probe.</p> <p>Lorsque l'ID de connexion est 0, la sonde n'est pas connectée au serveur ObjectServer.</p> <p>Dans les déploiements de reprise en ligne et multiniveaux, l'ID de connexion n'est défini que sur les serveurs ObjectServer principal et de collecte. Dans une configuration de reprise en ligne, l'ID de connexion du serveur ObjectServer de secours est toujours 0. Dans un déploiement multiniveau, l'ID de connexion est toujours 0 sur les serveurs ObjectServer d'affichage et d'agrégation.</p> <p>Remarque : Lorsqu'une sonde est connectée au serveur ObjectServer via un serveur proxy, l'ID connexion de la sonde peut varier dans le temps et peut donc être enregistré de façon incorrecte. Cela s'explique par le fait que le serveur proxy optimise ses connexions au serveur ObjectServer et déplace les connexions de sonde de façon dynamique. Toutefois, l'ID connexion stocké dans la table registry.probes reste le même. Il n'est pas mis à jour lorsqu'une sonde est déplacée vers une autre connexion sur le même serveur proxy.</p>
Hostname	VARCHAR(64)	Nom de domaine qualifié complet de l'ordinateur sur lequel la sonde s'exécute.
HTTP_port	entier	Numéro de port sur lequel l'interface HTTP de la sonde est à l'écoute. Les propriétés de sonde NHttpd.EnableHTTP et NHttpd.ListeningPort doivent être activées pour que ce port soit actif. Lorsque le port n'est pas actif, la valeur par défaut de cette zone est 0.
HTTPS_port	entier	Numéro de port sur lequel l'interface HTTPS de la sonde est à l'écoute. Les propriétés de sonde NHttpd.SSLEnable et NHttpd.SSLListingPort doivent être activées pour que ce port soit actif. Lorsque le port n'est pas actif, la valeur par défaut de cette zone est 0.
LastUpdate	time	Horodatage de la dernière mise à jour de la table registry.probes. Cette colonne est remplie par le déclencheur ObjectServer registry_new_probe.
Name	VARCHAR(128)	Valeur de la propriété Name dans le fichier de propriétés de la sonde.
PID	integer	Identificateur de processus en cours de la sonde (PID).
ProbeType	VARCHAR(128)	Représentation sous forme de chaîne du type de sonde se connectant au serveur ObjectServer, par exemple, «simnet» ou «tivoli_eif».

Tableau 83. Colonnes de la table registry.probes (suite)

Nom de colonne	Type de données	Description
RulesChecksum	CHAR(40)	Total de contrôle qui identifie la version du fichier de règles utilisé par la sonde. Si le fichier de règles comporte des fichiers Include ou Lookup, le total de contrôle est une combinaison des totaux de contrôle du fichier de règles et de ces fichiers Lookup et Include. Cette valeur change quand le contenu du fichier de règles change. Remarque : Lorsqu'un fichier de règles est mis en cache, la valeur du total de contrôle est préservée dans des métadonnées de commentaire dans le fichier mis en cache. Ne modifiez pas cette valeur dans le fichier mis en cache. Voici un exemple de métadonnées de commentaire contenant un total de contrôle : #METADATA CheckSum=f967b187bf66e08d4277ab688f87f696bea3bb22
StartTime	time	Heure de démarrage de la sonde. Cette information vous permet de déterminer si la sonde est en cours de démarrage ou de reconnexion.
Etat	entier	Indique l'état de la sonde : 0 : La sonde est arrêtée. 1 : La sonde est en cours d'exécution.

Types de données ObjectServer

Chaque valeur de colonne du serveur ObjectServer est associée à un type de données. Le type de données détermine la manière dont le serveur ObjectServer traite les données de la colonne.

Par exemple, l'opérateur plus (+) permet d'ajouter des valeurs entières ou de concaténer des valeurs de chaîne, mais n'a aucun effet sur les valeurs booléennes. Les types de données pris en charge par le serveur ObjectServer figurent dans le tableau suivant :

Tableau 84. Types de données ObjectServer

Type SQL	Description	Valeur par défaut	ID du serveur ObjectServer correspondant au type de données
INTEGER	Entier signé 32 bits.	0	0
INCR	Entier à incrémentation automatique non signé 32 bits. S'applique aux colonnes de table uniquement, et peut uniquement être mis à jour par le système.	0	5
UNSIGNED	Entier non signé 32 bits.	0	12
BOOLEAN	TRUE ou FALSE.	FALSE	13
REAL	Nombre en virgule flottante signé 64 bits.	0.0	14
TIME	Heure, enregistré en nombre de secondes depuis le 1er janvier 1970 à minuit. Il s'agit de l'heure UTC (temps universel coordonné).	Jeudi 1 janvier 1970 01:00:00	1

Tableau 84. Types de données ObjectServer (suite)

Type SQL	Description	Valeur par défaut	ID du serveur ObjectServer correspondant au type de données
CHAR(<i>entier</i>)	Chaîne de caractères à taille fixe, composée de <i>entier</i> caractères (8192 octets étant la valeur maximale). Le type char est identique dans l'opération à varchar, mais les performances sont meilleures pour les mises à jour de masse qui modifient la longueur de la chaîne.	' '	10
VARCHAR(<i>entier</i>)	Chaîne de caractères à taille variable, composée de <i>entier</i> caractères au maximum (8192 octets étant la valeur maximale). Le type varchar utilise moins d'espace de stockage que le type char et les performances sont meilleures pour les opérations de dédoublonnage, de numérisation, d'insertion et de suppression.	' '	2
INTEGER64	Entier signé 32 bits.	0	16
UNSIGNED64	Entier non signé 64 bits.	0	17

Remarque : Vous pouvez afficher uniquement les colonnes de type CHAR, VARCHAR, INCR, INTEGER et TIME dans la liste d'événements. N'ajoutez pas de colonnes d'un autre type dans la table alerts.status.

Remarques

Ces informations ont été développées pour les produits et les services proposés aux Etats-Unis.

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. Toutefois, il est de la responsabilité de l'utilisateur d'évaluer et de vérifier le fonctionnement 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 seul titre d'information et n'impliquent en aucun cas l'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 données de performances indiquées dans ce document ont été déterminées dans un environnement surveillé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les

résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

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'ÉTAT» SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DÉCLINE NOTAMMENT TOUTE RESPONSABILITÉ RELATIVE À CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DÉFAUT D'APTITUDE À L'EXÉCUTION D'UN TRAVAIL DONNÉ. DANIEL VEILLARD NE POURRA EN AUCUN CAS ÊTRE TENU RESPONSABLE POUR TOUT DOMMAGE, QUELLES QU'EN SOIT

LA CAUSE ET LES RESPONSABILITÉS, CONTRACTUELLES OU NON, OU PAR FAUTE LIÉE À 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

AIX, IBM, le logo IBM, ibm.com, Informix, Netcool, System z, Tivoli, et Tivoli Enterprise Console sont des marques d'International Business Machines Corporation aux États-Unis et/ou dans certains autres pays.

Adobe, Acrobat, Portable Document Format (PDF), PostScript ainsi que toutes les marques incluant Adobe sont des marques d'Adobe Systems Incorporated aux États-Unis et/ou dans certains autres pays.



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

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

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux États-Unis et/ou dans certains autres pays.

UNIX est une marque de The Open Group aux États-Unis et/ou dans certains autres pays.

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

Index

Caractères spéciaux

@Identifiant 10, 11
@Tally 11

A

accessibilité xi
administration à distance des sondes
 nco_http 109
arch
 répertoire du système
 d'exploitation xi
associateur 223
atomes
 description 345
attribut COUNTERPART dans les
 passerelles 177
attributs, accès aux valeurs du
 cache 256, 270
attributs, dynamique 256, 270
attributs, mappage 256, 270
attributs d'accès aux valeurs du
 cache 256, 270
attributs de mappage 256, 270
attributs dynamiques 256, 270
auto-surveillance
 sondes 87
auto-surveillance de sonde
 ressources 90

B

bibliothèque d'expressions régulières
 NETCOOL 345
Bibliothèque d'expressions régulières
 TRE 347

C

centre de documentation logicielle
 Tivoli ix
chiffrement
 mots de passe pour le serveur
 ObjectServer 17, 180
chiffrement de mots de passe 17, 180
commande
 FAILOVER SYNCH 263, 310
 GET CONFIG 263, 309
 nco_sql 262, 309
 SHOW PROPS 263, 310
commande de passerelle ADD
 ROUTE 199, 213
commande de passerelle CREATE
 FILTER 200, 212
commande de passerelle CREATE
 MAPPING 210

commande de passerelle DROP
 FILTER 213
commande de passerelle DROP
 MAPPING 211
commande de passerelle DUMP
 CONFIG 214
commande de passerelle LOAD
 CONFIG 214
commande de passerelle LOAD
 FILTER 212
commande de passerelle REMOVE
 ROUTE 213
commande de passerelle SAVE
 CONFIG 214
commande de passerelle SET
 CONNECTIONS 215
commande de passerelle SET DEBUG
 MODE 216
commande de passerelle SHOW
 MAPPING ATTRIBUTES 212
commande de passerelle SHOW
 MAPPINGS 211
commande de passerelle SHOW
 READERS 208
commande de passerelle SHOW
 ROUTES 214
commande de passerelle SHOW
 SYSTEM 215
commande de passerelle SHOW WRITER
 ATTRIBUTES 210
commande de passerelle SHOW WRITER
 TYPES 209
commande de passerelle SHOW
 WRITERS 209
commande de passerelle
 SHUTDOWN 215
commande de passerelle START
 READER 198, 206
commande de passerelle START
 WRITER 198, 208
commande de passerelle STOP
 READER 207
commande de passerelle STOP
 WRITER 198, 209
commande de passerelle
 TRANSFER 216
commande de publications ix
commande FAILOVER SYNCH 263, 310
commande GET CONFIG 263, 309
commande nco_sql 262, 309
commande SHOW PROPS 263, 310
commandes 262, 309
commandes de configuration
 passerelles 214
commandes de passerelle 205
configuration
 débits d'événements anormaux 78
 débordement d'événements 78
 propriétés communes des passerelles
 passerelle bidirectionnelle 250,
 303

configuration (*suite*)
 propriétés communes des passerelles
 (*suite*)
 passerelle unidirectionnelle 242,
 295
 statistiques de sonde 91
configuration de passerelle
 .conf 197
 fichier de commandes de
 démarrage 183
 fichier de définition de mappe 183
 fichier de définition de réplication de
 table 183
 fichier de propriétés 183
configuration de passerelles
 fichiers de configuration 183
 isql 204
 nco_sql 204
 passerelles 183
constructions multiculturelles
 expressions régulières 352
conventions, police xi
conventions typographiques xi
conversion des mises à jour en
 insertion 264, 313
corrélation d'événements 11

D

débits d'événements anormaux
 configuration 78
débogage
 fichiers de règles 65
 sondes 16, 327
dédoublonnage 11, 48

E

éléments
 dans le fichier de règles de sonde 22
éléments temporaires
 dans le fichier de règles de sonde 23
exécution des sondes
 exécution de sondes sous SUID
 root 99
 SETUID 99
exemple de fichier de définition de
 réplication de table 268, 316
expressions entre crochets
 expressions régulières 350
expressions régulières
 atomes 345
 bibliothèque NETCOOL 345
 bibliothèque TRE 347
 constructions multiculturelles 352
 expressions entre crochets 350
 métacaractères 348
 présentation 345
 propriété RegexpLibrary 345
 quantificateurs minimaux 349

- expressions régulières (*suite*)
 - quantificateurs non gourmands 349
 - séquences d'échappement avec barres obliques inversées 353

F

- fichier, définition de mappe 255, 257, 269, 271
- fichier, définition de réplication de table 264, 313
- fichier de commandes, démarrage 262, 309
- fichier de commandes de démarrage 262, 309
- fichier de définition de mappe 255, 257, 269, 271
- fichier de définition de réplication de table 264, 313
- fichier de définition de réplication de table, exemple 268, 316
- fichier de règles 66
 - fonctions 33
 - opérateurs 33
- fichier de règles de configuration du débordement 82
- fichier de règles de débordement 85
- fichiers de règles 65
- filtre de suppression, resynchronisation 264, 313
- filtre de suppression de resynchronisation 264, 313
- filtres
 - commandes 212
 - dans les passerelles 200
- flood.config.rules 82
- flood.rules 85
- flot d'événements
 - configuration 78
- fonction détails
 - dans le fichier de règles de sonde 48
- fonction exists
 - dans le fichier de règles de sonde 37
- fonction log
 - dans le fichier de règles de sonde 49
- fonction Rechercher/Remplacer
 - dans les fichiers de règles d'analyse 57
- fonction service
 - dans le fichier de règles de sonde 58
- fonction setlog
 - dans le fichier de règles de sonde 49
- fonction update
 - dans le fichier de règles de sonde 48
- fonctions
 - fichiers de règles 33
- fonctions arithmétiques
 - dans le fichier de règles de sonde 42
- fonctions d'heure
 - dans le fichier de règles de sonde 43
- fonctions de chaîne
 - dans le fichier de règles de sonde 38
- fonctions de conversion 255, 269
- fonctions de conversion du fichier de définition de mappe 255, 269

- fonctions de date
 - dans les fichiers de règles de sonde 43
- fonctions mathématiques
 - dans le fichier de règles de sonde 42
- formation
 - voir formation technique Tivoli xi
- formation technique Tivoli xi
- Formation technique Tivoli xi

G

- génération d'événements à distance
 - nco_probeeventfactory 116
- genevent 50, 53
- groupe d'utilisateurs ncoadmin 201

I

- identification et résolution des problèmes
 - passerelles 335
 - sondes 327
- IDUC 206
- inclure des fichiers
 - dans le fichier de règles de sonde 70
- indicateur ON INSERT ONLY dans les passerelles 199
- informations de support xi
- instruction de commutateur dans les fichiers de règles 32
- instructions IF dans les fichiers de règles 31
- interface interactive SQL
 - isql 203
 - nco_sql 203
- interface SQL interactive
 - isql 196
 - nco_sql 196

L

- langage de règles de sonde
 - mots réservés 60

M

- manuels ix
- mappage, passerelle ObjectServer 255, 269
- mappage de passerelles, ObjectServer 255, 269
- mappages
 - commandes 210
 - dans les passerelles 199
- messages d'erreur
 - passerelles 335
 - sondes 319
- métacaractères
 - expressions régulières 348
- mise à jour de propriété à distance
 - nco_setprobeprop 115
- mode de capture brute dans les sondes 16
- mode de reprise en ligne d'égal à égal
 - sondes 17

- mode sécurisé
 - pour les passerelles 180
 - pour les sondes 17
- mode stocker-et-transmettre
 - dans les passerelles 179
 - dans les sondes 12
- modification
 - propriétés de sonde 97
- mots réservés
 - langage de règles de sonde 60

N

- nco_aes_crypt 17
- nco_g_crypt 17, 180
- nco_objserv 98

O

- ObjectServer
 - de sauvegarde 221
 - types de données 368
- ObjectServer, principal et de sauvegarde 263, 310
- ObjectServer principal et de sauvegarde 263, 310
- opérateurs
 - fichier de règles 33
 - opérateurs arithmétiques
 - dans le fichier de règles de sonde 35
 - opérateurs de chaîne
 - dans le fichier de règles de sonde 35
 - opérateurs de comparaison
 - dans le fichier de règles de sonde 37
 - opérateurs de manipulation de bit
 - dans le fichier de règles de sonde 36
 - opérateurs logiques
 - dans le fichier de règles de sonde 37
 - opérateurs mathématiques
 - dans le fichier de règles de sonde 35
- option de ligne de commande
 - messagelevel 65
- option de ligne de commande
 - messagelog 65
- options de ligne de commande
 - passerelle bidirectionnelle
 - propriétés communes des passerelles 250, 303
 - passerelle unidirectionnelle
 - propriétés communes des passerelles 242, 295
 - passerelles 201
 - sondes 129
- outil de ligne de commande interactif
 - pour les passerelles
 - nco_g_icmd 196

P

- passerelle ObjectServer
 - bidirectionnelles 221
 - propriétés communes des passerelles 250, 303
 - exemple de mappage 257, 271
 - mappage 255, 269
 - unidirectionnelle 223

- passerelle ObjectServer (*suite*)
 - propriétés communes des passerelles 242, 295
- passerelle ObjectServer bidirectionnelle
 - configuration
 - propriétés communes des passerelles 250, 303
 - description 221
 - exemple 221
 - flux d'informations 221
- passerelle ObjectServer unidirectionnelle
 - configuration
 - propriétés communes des passerelles 242, 295
 - description 223
 - exemple 223
 - flux d'informations 223
- passerelles
 - attribut COUNTERPART 177
 - bidirectionnelles 174, 176
 - commande ADD ROUTE 213
 - commande CREATE FILTER 212
 - commande CREATE MAPPING 210
 - commande DROP FILTER 213
 - commande DROP MAPPING 211
 - commande DUMP CONFIG 214
 - commande LOAD CONFIG 214
 - commande LOAD FILTER 212
 - commande REMOVE ROUTE 213
 - commande SAVE CONFIG 214
 - commande SET CONNECTIONS 215
 - commande SET DEBUG MODE 216
 - commande SHOW MAPPING 211
 - commande SHOW MAPPING ATTRIBUTES 212
 - commande SHOW READERS 208
 - commande SHOW ROUTES 214
 - commande SHOW SYSTEM 215
 - commande SHOW WRITER ATTRIBUTES 210
 - commande SHOW WRITER TYPES 209
 - commande SHOW WRITERS 209
 - commande SHUTDOWN 215
 - commande START READER 206
 - commande START WRITER 208
 - commande STOP READER 207
 - commande STOP WRITER 209
 - commande TRANSFER 216
 - commandes de configuration 214
 - commandes de filtrage 212
 - commandes de mappage 210
 - commandes de passerelle 190, 196, 203
 - commandes de programme d'écriture 208
 - commandes de programme de lecture 206
 - commandes de route 213
 - commandes générales 215
 - description de filtre 200
 - description de mappage 199
 - description de route 179, 199
 - description du programme d'écriture 198
 - description du programme de lecture 178, 198

- passerelles (*suite*)
 - exécution de passerelles 219
 - fichier de commandes de démarrage 190
 - fichiers de définition de mappe 184
 - fichiers journaux 220
 - mappage 184
 - mappe 184
 - messages d'erreur 335
 - mode sécurisé 180
 - mode stocker-et-transmettre 179
 - modules de programme de lecture/écriture 178
 - options de ligne de commande 201
 - options de ligne de commande de passerelle 192
 - présentation 173
 - propriétés de passerelle 192
 - types 174
 - unidirectionnelle 174, 177
- passerelles bidirectionnelles 174, 176
- passerelles unidirectionnelles 174
- Probe Rules Utility
 - nco_rules_util 72
 - caractères d'échappement 76
 - nco_rules_util.props 72
 - variables d'environnement 76
- programme d'écriture 223
- programme de lecture 223
- programme de lecture/écriture 221
- programmes d'écriture
 - commandes 208
 - dans les passerelles 198
- programmes de lecture
 - commandes 206
 - dans les passerelles 178, 198
- propriété RegexpLibrary 345
- propriétés
 - dans le fichier de règles de sonde 24
 - sondes 129
- publications ix
- publications en ligne ix

Q

- quantificateurs minimaux
 - expressions régulières 349
- quantificateurs non gourmands
 - expressions régulières 349

R

- rechargement due fichier de règles
 - nco_probereloadrules 114
- registertarget 50
- registry.probes 2
- registry.probes, table 366
- répertoire du système d'exploitation
 - arch xi
- réplication, table 264, 313
- réplication de table 264, 313
- reprise en ligne 263, 310
- reprise en ligne de l'ObjectServer 263, 310
- routes
 - commandes 213

- routes (*suite*)
 - dans les passerelles 179, 199

S

- séquences d'échappement avec barres obliques inversées
 - expressions régulières 353
- Serveur ObjectServer
 - paire de reprise en ligne 221
- setdefaulttarget 50
- settarget 50
- sonde générique 16
- sonde Ping 7
- sondes
 - API 4
 - auto-surveillance 87, 91
 - base de données 4
 - capture brute 16
 - collecte de données de mesure 91
 - composants 5
 - configuration de l'auto-surveillance 88
 - CORBA 4
 - débits d'événements anormaux 78
 - débogage 16, 327
 - débogage de fichiers de règles 65
 - dédoublonnage dans le fichier de règles 48
 - détection des débordements d'événements 78
 - diverses 5
 - édition des propriétés 97
 - éléments dans les fichiers de règles 22
 - éléments temporaires dans le fichier de règles 23
 - enregistrement 2
 - fichier de propriétés 6
 - fichier de règles 8, 69
 - fichier exécutable 5
 - fichier journal 3
 - fonction détails dans le fichier de règles 48
 - fonction log dans le fichier de règles 49
 - fonction Rechercher/Remplacer dans les fichiers de règles 57
 - fonction service dans le fichier de règles 58
 - fonction setlog dans le fichier de règles 49
 - fonction update dans le fichier de règles 48
 - fonctionnement 12
 - fonctions arithmétiques dans les fichiers de règles 42
 - fonctions d'heure dans les fichiers de règles 43
 - fonctions de chaînes dans les fichiers de règles 38
 - fonctions de date dans les fichiers de règles 43
 - fonctions mathématiques dans les fichiers de règles 42
 - identification et résolution des problèmes 327

- sondes (*suite*)
 - inclure des fichiers dans les fichiers de règles 70
 - instruction de commutateur dans les fichiers de règles 32
 - instructions IF dans les fichiers de règles 31
 - mémoire cache des fichiers de règles 69
 - messages d'erreur 319
 - mode de reprise en ligne d'égal à égal 17
 - mode sécurisé 17
 - opérateurs arithmétiques dans les fichiers de règles 35
 - opérateurs de chaîne dans les fichiers de règles 35
 - opérateurs de comparaison dans les fichiers de règles 37
 - opérateurs de manipulation de bit dans les fichiers de règles 36
 - opérateurs logiques dans les fichiers de règles 37
 - opérateurs mathématiques dans les fichiers de règles 35
 - options de ligne de commande 129
 - périphérique 3
 - personnalisations 77
 - présentation 1
 - propriétés 7
 - propriétés dans les fichiers de règles 24
 - stocker et transmettre 12
 - suppression d'éléments dans les fichiers de règles 38
 - tables de consultation dans les fichiers de règles 45
 - tables de recherche 46
 - téléchargement de fichiers de règles 68
 - test de fichiers de règles 64
 - traitement du fichier de règles 22
 - types 3
 - utilisation d'une sonde spécifique 9
 - zone Identificateur 11
 - zones dans les fichiers de règles 22
- sondes API 4
- sondes CORBA 4
- sondes de base de données 4
- sondes de fichier journal 3
- sondes de périphériques 3
- sondes diverses 5
- statistiques de sonde
 - configuration 91
- suppression
 - éléments dans les fichiers de règles de sonde 38
- symbole \$
 - dans le fichier de règles de sonde 22
- symbole @
 - dans le fichier de règles de sonde 10, 22
 - dans les mappages de passerelles 199
- symbole %
 - dans le fichier de règles de sonde 24

T

- table alerts.details 364
- table alerts.journal 365
- table alerts.status 355
- table de consultation
 - dans le fichier de règles de sonde 45
- table service.status 366
- tables de recherche 46
- tables de serveurs ObjectServer
 - registry.probes 366
- tables du serveur ObjectServer
 - alerts.details 364
 - alerts.journal 365
 - service.status 366
- tables ObjectServer
 - alerts.status 355
- test
 - fichiers de règles 64
- traitement du fichier de règles 22
 - dédoublonnage 11
 - exemples de fichier de règles 62
 - fonction détails 48
 - fonction exists 37
 - fonction log 49
 - fonction Rechercher/Remplacer 57
 - fonction setlog 49, 58
 - fonction update 48
 - fonctions d'heure 43
 - fonctions de chaîne 38
 - fonctions de date 43
 - fonctions mathématiques 42
 - instruction de commutateur 32
 - instructions IF 31
 - opérateurs de chaîne 35
 - opérateurs de comparaison 37
 - opérateurs de manipulation de bit 36
 - opérateurs logiques 37
 - opérateurs mathématiques 35
 - suppression d'éléments 38
 - table de consultation 45
 - traitement multiprocessus 55
 - type de données BOOLEAN 368
 - type de données CHAR 368
 - type de données INCR 368
 - type de données INTEGER 368
 - type de données INTEGER64 368
 - type de données REAL 368
 - type de données SHORT 368
 - type de données UNSIGNED 368
 - type de données UNSIGNED64 368
 - type de données UTC 368
 - type de données VARCHAR 368
 - types de données 368

U

- utilisateurs concernés ix

V

- variables, notation xi
- variables d'environnement
 - NCHOME 98, 219
 - OMNIHOME 98, 219
- variables d'environnement, notation xi

Z

- zone Identificateur 11
- zone Tally 11
- zones
 - dans le fichier de règles de sonde 22
 - Identificateur 11
 - Tally 11



SC43-0824-01

