Netcool/OMNIbus
Version 8.1

Guide d'administration

SC43-0823-01

Netcool/OMNIbus
Version 8.1

Guide d'administration

SC43-0823-01

Important
Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations figurant à la section «Remarques», à la page 493.

Deuxième édition - Novembre 2014

Réf. US : SC27-6265-01

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 CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE.

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

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

Vous pouvez également consulter les serveurs Internet suivants :

- http://www.fr.ibm.com (serveur IBM en France)
- http://www.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

La présente 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
Public visé ix
Publications ix
Accessibilité xi
Formation technique Tivoli xi
Informations de support xi
Conventions utilisées dans la présente publication . xi

Chapitre 1. Configuration du serveur ObjectServer. 1
Traitement des alertes dans le serveur ObjectServer . 1
Utilisation des propriétés et options de ligne de commande du serveur ObjectServer 1
Propriétés et options de ligne de commande du serveur ObjectServer. 4
Exécution du serveur ObjectServer en mode sécurisé 23
Mises à jour d'outil client via IDUC 25
Spécification de l'intervalle de mise à jour IDUC 26
Spécification du port IDUC 26
Configuration du serveur ObjectServer pour le support multiculturel 27
Protection du serveur ObjectServer contre les afflux d'événements. 28
Stockage des données et utilisation du point de contrôle 30
Stockage des données à l'aide de magasins de mémoire 31
Introduction à l'utilisation des points de contrôle 31
Utilitaire de vérification de point de contrôle nco_check_store 32
Modification des limites logicielle et matérielle du magasin de mémoire table_store 34
Utilisation de nco_postmsg pour envoyer des alertes aux serveurs ObjectServer 35
Propriétés et options de ligne de commande nco_postmsg 39
Exemples nco_postmsg et instructions INSERT obtenues 42

Chapitre 2. Configuration d'un serveur proxy 45
Démarrage du serveur proxy 45
Démarrage d'un serveur proxy en utilisant le contrôle de processus 46
Démarrage d'un serveur proxy en utilisant les services (Windows). 46
Démarrage manuel du serveur proxy. 46
Propriétés et options de ligne de commande du serveur proxy 47
Connexion au serveur proxy. 50
Exécution du serveur proxy en mode sécurisé . . . 50

Chapitre 3. Configuration d'un serveur pare-feu 53
Configuration d'un serveur pare-feu standard . . . 54
Configuration d'un serveur pare-feu multiple . . . 56
Configuration de reprise en ligne du serveur pare-feu 57
Démarrage du serveur pare-feu. 59
Démarrage d'un serveur pare-feu à l'aide du contrôle de processus 59
Démarrage d'un serveur pare-feu à l'aide des services Windows 59
Démarrage manuel du serveur pare-feu 59
Propriétés et options de ligne de commande du serveur pare-feu 60
Fichier de définition des hôtes sécurisés 64
Langage de commande d'un serveur de pont pare-feu 66
SHOW PROPS et GET CONFIG 66
GET PROP 67
SHOW DATAFLOWS 67
SET LOG LEVEL TO 68
SHUTDOWN. 68

Chapitre 4. Utilisation de Netcool/OMNIbus Administrator pour configurer les serveurs ObjectServer . 69
Démarrage de Netcool/OMNIbus Administrator . . 69
Remarques relatives au support multiculturel . . 69
Démarrage de Netcool/OMNIbus Administrator 70
Connexion à un serveur ObjectServer. 73
Connexion à un agent de processus 74
Utilisation des composants Tivoli Netcool/OMNIbus 76
Connexions SSL 76
Sélection d'objets ObjectServer à configurer. . . 78
Définition des préférences dans Netcool/OMNIbus Administrator 79
Sortie de Netcool/OMNIbus Administrator. . . 83
Gestion des autorisations avec des utilisateurs, des groupes, des rôles et des filtres de restriction . . . 83
Configuration des rôles 84
Configuration des groupes 89
Configuration d'utilisateurs 94
Configuration des filtres de restriction 99
Configuration des menus, outils et invites. . . . 102
Personnalisation des menus 103
Configuration des outils. 107
Configuration des invites 111
Configuration des automatisations 115
Configuration des déclencheurs 116
Configuration de procédures 130
Configuration des signaux 139
Configuration de l'apparence visuelle de la liste d'événements 143
Création et édition des conversions 143

© Copyright IBM Corp. 1994, 2014

Suppression de conversions 144
Création et édition de couleurs de gravité d'événement pour les listes d'événements Windows 144
Création et édition de visuels de colonne . . . 145
Suppression de visuels de colonne 146
Création et édition de classes 147
Suppression de classes 147
Configuration de bases de données, fichiers, propriétés, connexions et canaux ObjectServer . . 148
Configuration des bases de données. 148
Affichage et modification des propriétés du serveur ObjectServer 158
Configuration des fichiers du serveur ObjectServer. 160
Surveillance des connexions au serveur ObjectServer. 162
Configuration des canaux 163
Utilisation de l'interface interactive SQL en mode interface graphique 164

Chapitre 5. SQL ObjectServer 167
Objets du serveur ObjectServer 167
Héritage de droits d'accès d'un objet 169
Bases de données initialisées par le système . . 169
Tables système 171
Spécification des types de données des colonnes 171
Spécification des propriétés facultatives des colonnes 174
Affichage des détails de l'index 175
Automatisations ObjectServer 176
Déclencheurs 176
Procédures 192
Variables utilisateur implicites dans les procédures et les déclencheurs. 195
Automatisations Tivoli Netcool/OMNIbus standard 196
Automatisation des événements affectés par le service. 202
Exemples d'automatisation 203
Administration des utilisateurs, des groupes et des rôles 206
Commandes SQL ObjectServer 207
Mots réservés 211
Conventions de dénomination SQL ObjectServer 213
Syntaxe SQL. 214
Opérateurs 214
Fonctions. 221
Expressions 227
Conditions 228
interface interactive SQL 229
Exécution des commandes SQL 229
Exécution des commandes SQL à partir de fichiers 231
Chiffrement des mots de passe dans les scripts 232
Options de ligne de commande de l'interface interactive SQL. 232

Chapitre 6. Configuration de la notification des événements accélérés 235
Configuration d'une sonde pour indiquer des événements pour l'accélération 236
Configuration d'une passerelle de notification d'événement accéléré 236
Configuration de la table alerts.status pour recevoir l'indicateur AEN 237
Configuration des canaux afin de diffuser les données d'événement. 238
Création et édition de canaux 238
Copie et collage de canaux 242
Suppression d'un canal 242
Envoi de messages aux destinataires de canal 242
Déconnexion des clients de notification d'événement accéléré 243
Arrêt des clients de notification d'événement accéléré 244
Configuration des déclencheurs pour la prise en charge de la notification d'événement accéléré . . 245

Chapitre 7. Utilisation du contrôle de processus pour gérer des processus et des procédures externes 247
Connexion des agents de processus 247
Résolution du nom d'hôte au démarrage . . . 249
Composants du contrôle de processus 249
Agents de processus 249
Processus. 250
Services 250
Utilitaires de contrôle de processus 251
Création et démarrage d'un système réseau de contrôle de processus. 252
Création de groupes d'utilisateurs UNIX pour le système de contrôle de processus. 253
Exigences des comptes Windows pour le système de contrôle de processus. 254
Configuration des informations de communication du serveur pour les agents de processus. 254
Mise à niveau du fichier de configuration de contrôle de processus par défaut 255
Démarrage manuel des agents de processus . . 256
Considérations concernant le sécurité de agent de processus. 263
Démarrage automatique des agents de processus sous UNIX. 266
Démarrage automatique des agents de processus sous Windows 268
Gestion de la configuration du système de contrôle de processus. 268
Configuration et gestion du contrôle de processus à partir de la ligne de commande 269
Définition de processus, de services et d'hôtes pour le contrôle de processus 269
Gestion du contrôle de processus à l'aide des utilitaires de contrôle de processus 280
Utilisation de Netcool/OMNIbus Administrator pour gérer le contrôle de processus 289
Connexion à un agent de processus 289

Affichage et configuration des informations d'état d'un agent de processus 291
Affichage des processus et services d'un agent de processus. 292
Configuration des services d'un agent de processus. 294
Configuration des processus 297
Copie et collage d'un service ou d'un processus entre les hôtes de l'agent de processus 304
Exécution d'une action externe 306
Arrêt d'un agent de processus. 307
Utilisation du contrôle de processus pour exécuter des procédures externes dans les automatisations . 307

Chapitre 8. Optimisation des performances 309
Indicateurs clés de performances Tivoli Netcool/OMNIbus 310
Indicateurs clés de performances du serveur ObjectServer. 310
Indicateurs clés de performances de la sonde 314
Indicateurs clés des performances de la passerelle. 315
Valeurs recommandées pour l'optimisation des performances 316
Exécution du serveur ObjectServer avec le profilage activé. 316
Collecte des informations statistiques sur les déclencheurs 318
Vérification et modification de l'architecture du système 321
Activation du groupe de déclencheurs stats_triggers 321
Vérification et modification des fichiers de configuration des sondes 322
Configuration de la détection du flot d'événements 322
Gestion du volume d'informations dans la table alerts.details. 323
Utilisation d'un agent de surveillance pour contrôler et gérer les ressources Tivoli Netcool/OMNIbus 323
Vérification et modification de vos requêtes SQL, puis création d'une sélection d'index bien conçus et efficaces. 324
Suivi des tendances de performances à des intervalles réguliers 326
Surveillance automatique de Tivoli Netcool/OMNIbus 326
Installation du tableau de bord dans un environnement d'équilibrage de charge. . . . 328
Seuils de surveillance automatique 329
Instructions de requête SQL 333
Consultation des résultats des optimisations de requête automatiques. 333
Optimisation manuelle des requêtes. 336
Instructions d'indexation 336
Exemple d'utilisation d'index avec les requêtes SQL 338
Exemple d'utilisation d'index avec des déclencheurs ou des procédures 339
Valeurs recommandées lors de la création de déclencheurs 340

Annexe A. Aide-mémoire des commandes d'application 345

Annexe B. Informations de référence sur les commandes SQL ObjectServer 353
ADD COLUMN 353
ALTER COLUMN. 354
ALTER FILE. 354
ALTER GROUP 355
ALTER ROLE 355
ALTER SYSTEM 356
ALTER TABLE 357
ALTER TRIGGER 358
ALTER TRIGGER GROUP 359
ALTER USER 360
CASE WHEN 361
CHECK STATEMENT 361
CREATE DATABASE. 362
CREATE FILE 362
CREATE GROUP 363
CREATE INDEX 364
CREATE PROCEDURE (procédures externes). . . 365
CREATE PROCEDURE (procédures SQL) 367
CREATE RESTRICTION FILTER 370
CREATE ROLE. 371
CREATE SIGNAL 371
CREATE TABLE 372
CREATE TRIGGER (déclencheur de base de données) 373
CREATE TRIGGER (déclencheur de signal) . . . 375
CREATE TRIGGER (déclencheur temporel) . . . 377
CREATE TRIGGER GROUP 379
CREATE USER. 379
CREATE VIEW. 380
DELETE 381
DESCRIBE 382
DROP COLUMN 383
DROP DATABASE 385
DROP FILE 386
DROP GROUP 386
DROP INDEX 386
DROP PROCEDURE 387
DROP RESTRICTION FILTER 387
DROP ROLE 387
DROP SIGNAL. 387
DROP TABLE 388
DROP TRIGGER 388
DROP TRIGGER GROUP 388
DROP USER. 389
DROP VIEW 389
EXECUTE PROCEDURE 389
FOR 390
FOR EACH ROW 390
GRANT 392
GRANT ROLE 394
IF THEN ELSE 395
IDUC EVTFT 395

IDUC FLUSH 396
IDUC SNDMSG 397
INSERT 397
RAISE SIGNAL 398
REVOKE 398
REVOKE ROLE 400
SELECT (scalaire) 401
SELECT (aggregate) 403
SELECT (GROUP BY) 405
SET. 406
SET DATABASE et USE DATABASE 407
SVC 407
UPDATE 408
WRITE INTO 410

Annexe C. Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires . . 411

Annexe D. Tables du serveur ObjectServer. 415
Tables d'alertes 415
Table alerts.status 415
Table alerts.details 424
Table alerts.journal 425
Table alerts.iduc_messages 426
Table alerts.application_types 426
Table master.class_membership 427
Tables de services 427
Table service.status 428
Tables de catalogues système 428
Table catalog.memstores 428
Table catalog.databases 429
Table catalog.tables 429
Table catalog.base_tables 430
Table catalog.views 430
Table catalog.files 431
Table catalog.restrictions 431
Table catalog.columns 431
Table catalog.primitive_signals 432
Table catalog.primitive_signal_parameters . . . 432
Table catalog.trigger_groups 433
Table catalog.triggers 433
Table catalog.database_triggers 434
Table catalog.signal_triggers 434
Table catalog.temporal_triggers 435
Table catalog.procedures. 435
Table catalog.sql_procedures 435
Table catalog.external_procedures 435
Table catalog.procedure_parameters 436
Table catalog.connections 437
Table catalog.properties 437
Table catalog.security_permissions 438
Table catalog.profiles 438
Table catalog.indexes 439
Tables de statistiques 439
Table catalog.profiles 440
Table master.stats 440
Table catalog.trigger_stats 441
Table catalog.channel_stats 442
Tables de prise en charge d'outils client 443
Table alerts.resolutions 443
Table alerts.conversions 443
Table alerts.col_visuals 444
Table alerts.colors 444
Tables d'outils de bureau 445
Table tools.actions 445
Table tools.action_access 446
Table tools.menus 446
Table tools.menu_items 447
Table tools.prompt_defs 447
Table tools.menu_defs 448
Tables du serveur ObjectServer de bureau 448
Table master.national 448
Table master.servergroups 449
Tables de sécurité pour la compatibilité amont . . 449
Tables IDUC. 449
Table iduc_system.channel 449
Table iduc_system.channel_interest 450
Table iduc_system.channel_summary 450
Table iduc_system.channel_summary_cols . . . 450
Table iduc_system.iduc_stats 451
Tables d'événements affectés par un service . . . 451
Table precision.service_affecting_event 451
Table precision.service_details 451
Table precision.entity_service 452
Tables de registre 452
Table registry.oslc 452
Table registry.probes 453
Tables de surveillance automatique 455
Table master.sm_activity 455
Table master.sm_thresholds 456

Annexe E. Conseils pour les messages d'erreur 459
Messages d'erreur générés par les composants côté serveur 461
ALD 461
APM 466
AUT 466
ETC 467
GOB 469
GTK 471
IVM 472
OBX 475
PRO 480
REG 480
REP 481
STK 488
Messages d'erreur générés par le composant interface graphique Web. 490
HEMIN 490

Remarques 493
Marques 496

Index 497

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.

© Copyright IBM Corp. 1994, 2014

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 manuel *Guide d'administration d'IBM Tivoli Netcool/OMNIbus* fournit des informations détaillées sur les outils d'administration et les fonctions de Tivoli Netcool/OMNIbus. De plus, ce manuel est destiné à être utilisé comme un guide de référence pour vous aider dans la conception et la configuration de votre environnement.

Public visé

La présente publication s'adresse aux administrateurs chargés de configurer Tivoli Netcool/OMNIbus.

Publications

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

Votre bibliothèque Tivoli Netcool/OMNIbus

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

- *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*,

  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 comme 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*,

© Copyright IBM Corp. 1994, 2014

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

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

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

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. La valeur de NCHOME est la suivante :

- UNIX Linux `$NCHOME` prend par défaut la valeur `/opt/IBM/tivoli/netcool`
- Windows `%NCHOME%` prend par défaut la valeur `C:\IBM\Tivoli\Netcool`

Lorsqu'un répertoire ou une commande commence par la variable *NCHOME*, les informations s'appliquent à tous les systèmes d'exploitation pris en charge.

D'autres produits qui utilisent la variable d'environnement NCHOME (par exemple, 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
`hpux11hpia`	Systèmes HP-UX Itanium
`linux2x86`	Systèmes Red Hat Linux et SUSE
`linux2s390`	Linux for System z
`solaris2`	Systèmes Solaris
`win32`	Systèmes Windows

Emplacement de 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 de OMNIHOME comme suit :

- UNIX Linux Définissez $OMNIHOME sur $NCHOME/omnibus
- Windows Définissez %OMNIHOME% sur %NCHOME%\omnibus

Répertoire d'origine de l'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 d'origine de l'interface graphique Web est distinct de ceux de Jazz for Service Management

Répertoires d'origine de Jazz for Service Management

REP_INSTALL_WAS
Fait référence à l'emplacement d'installation de WebSphere Application Server. 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 d'installation de Jazz for Service Management. 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 qui est utilisé pour Jazz for Service Management. Cet emplacement se trouve dans le sous-répertoire `/profile` du répertoire d'origine de Jazz for Service Management :

UNIX Linux *REP_INSTALL_JazzSM*`/profile`

Windows *REP_INSTALL_JazzSM*`\profile`

REP_INSTALL_DASH
Fait référence à l'emplacement d'installation de Concentrateur des services d'application du tableau de bord. 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, voir le centre de documentation Jazz for Service Management à l'adresse http://www-01.ibm.com/support/knowledgecenter/SSEKCU/welcome.

Conventions pour les graphiques de repérage

La documentation Tivoli Netcool/OMNIbus contient des graphiques de repérage permettant d'indiquer que certaines parties d'une rubrique ou d'une instruction s'appliquent uniquement dans certaines conditions. Le tableau suivant décrit la signification de chaque graphique.

Tableau 2. Graphiques de repérage de Tivoli Netcool/OMNIbus

Graphique	Description
Web GUI	Le texte ou l'instruction s'applique uniquement au composant interface graphique Web.
UNIX	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 : AIX Solaris
Linux	Le texte ou l'instruction s'applique uniquement aux systèmes d'exploitation Linux.
Windows	Le texte ou l'instruction s'applique uniquement aux systèmes d'exploitation Windows.
32-bit	Le texte ou l'instruction s'applique uniquement aux systèmes d'exploitation 32 bits.
64-bit	Le texte ou l'instruction s'applique uniquement aux systèmes d'exploitation 64 bits.
FIPS 140-2	Le texte ou l'instruction s'applique uniquement à l'utilisation ou à la configuration du chiffrement FIPS 140-2.
Default	Le texte ou l'instruction décrit le comportement par défaut ou s'applique uniquement aux configurations par défaut.
Fix Pack 1	Le texte ou l'instruction s'applique uniquement au numéro de groupe de correctifs (fixpack) indiqué par le graphique. Les fonctions ou améliorations décrites dans le texte sont uniquement disponibles une fois que vous avez installé le groupe de correctifs indiqué. **Remarque :** Les groupes de correctifs du composant serveur et de l'interface graphique Web sont publiés séparément. Les groupes de correctifs de l'interface graphique Web sont indiqués comme suit : Web GUI Fix Pack 1
Administrator	Le texte ou l'instruction s'applique uniquement aux administrateurs de l'interface graphique Web, c'est-à-dire les utilisateurs ayant les rôles ncw_user et ncw_admin.
C	Le texte ou l'instruction s'applique uniquement au langage de programmation C.
Java	Le texte ou l'instruction s'applique uniquement 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 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. Configuration du serveur ObjectServer

Au moins un serveur ObjectServer est requis pour stocker et gérer les informations d'alerte dans une installation Tivoli Netcool/OMNIbus. Vous pouvez configurer le serveur ObjectServer à l'aide de ses propriétés et options de ligne de commande. Vous pouvez également le configurer pour qu'il n'accepte que les connexions sécurisées. De plus, vous pouvez configurer le support IDUC (Insert, Delete, Update, or Control), le support multiculturel, ainsi que la reprise en ligne et la reprise par restauration automatiques.

Traitement des alertes dans le serveur ObjectServer

Les alertes sont stockées en tant que lignes ou entrées dans la table alerts.status du serveur ObjectServer. Chaque alerte possède une zone Identificateur qui identifie de manière unique la source de l'incident. L'identificateur est généré par la sonde en fonction de la spécification figurant dans le fichier de règles de la sonde.

Lorsqu'une alerte est réacheminée vers le serveur ObjectServer, le système recherche une zone Identifier correspondante dans la table alerts.status. Si aucune entrée contenant le même identificateur n'est détectée, une nouvelle entrée d'alerte est insérée. L'entrée contient des informations détaillées sur le problème. Par exemple, la zone FirstOccurrence indique l'heure à laquelle le problème est survenu pour la première fois.

Si une entrée contenant le même identificateur est détectée, un dédoublonnage a lieu. Le serveur ObjectServer accuse réception de l'occurrence de l'entrée en double en ajoutant un nombre à l'entrée existante. Par défaut, le serveur ObjectServer met à jour la zone LastOccurrence de l'entrée existante avec l'heure de la nouvelle alerte et incrémente la zone Tally.

Remarque : La version 7.0, ou toute version ultérieure, du déclencheur de dédoublonnage est prioritaire sur le paramètre Update on Deduplication (Mettre à jour lors du dédoublonnage) si la zone Identifier est explicitement désignée par le déclencheur.

Le serveur ObjectServer peut aussi répondre automatiquement à des alertes indiquées à l'aide de l'automatisation.

Vous pouvez exporter des informations d'alerte vers d'autres applications via une passerelle ou utiliser une passerelle bidirectionnelle du serveur ObjectServer pour offrir une prise en charge de la reprise en ligne sur un autre ObjectServer.

Utilisation des propriétés et options de ligne de commande du serveur ObjectServer

Le serveur ObjectServer lit son fichier de propriétés à son démarrage. Si une propriété n'est pas spécifiée dans ce fichier, la valeur par défaut est utilisée à moins qu'une option de ligne de commande ne soit utilisée pour la remplacer.

© Copyright IBM Corp. 1994, 2014

A propos du fichier de propriétés du serveur ObjectServer

L'emplacement par défaut du fichier de propriétés est `$NCHOME/omnibus/etc/`*`nom_serveur`*`.props`. Dans le fichier de propriétés du serveur ObjectServer, la propriété et sa valeur correspondante sont séparées par deux points (:). Les valeurs de chaîne sont entourées par des guillemets simples et droits. Par exemple :

```
Name: 'NCOMS'
```

Chaque propriété du serveur ObjectServer possède une valeur par défaut. Dans un fichier de propriétés non modifié, les valeurs par défaut sont attribuées à toutes les propriétés. Le haut du fichier contient une liste de référence de propriétés avec leurs valeurs, types de données et descriptions par défaut. Ces propriétés se voient ajouter le signe dièse (#) au début de la ligne. Sous la liste de ces propriétés, une autre liste de propriétés avec leurs propriétés par défaut est fournie à des fins d'édition, si nécessaire.

Remarque : Si vous exécutez le serveur ObjectServer au codage UTF-8 sous Windows, vous devez également enregistrer le fichier de propriétés du serveur ObjectServer au codage UTF-8.

Spécification des propriétés du serveur ObjectServer

Vous pouvez modifier les propriétés du serveur ObjectServer de l'une des manières suivantes :

- Modifiez le fichier de propriétés et la valeur des propriétés souhaitées. Modifiez les entrées sous la liste de propriétés commentée. Lorsque vous apportez des modifications aux propriétés du fichier de propriétés, les modifications ne prennent pas effet tant que vous n'avez pas redémarré le serveur ObjectServer.
- Exécutez la commande ALTER SYSTEM SET à partir de l'interface interactive SQL.
- Dans l'interface Netcool/OMNIbus Administrator, utilisez le bouton de menu **System (Système)** et l'option **Properties (Propriétés)** pour afficher les propriétés et modifier leurs valeurs.

  Lorsque vous utilisez la commande ALTER SYSTEM SET ou Netcool/OMNIbus Administrator pour modifier les propriétés du serveur ObjectServer, les modifications apportées à certaines propriétés ne prennent pas effet tant que vous n'avez pas redémarré le serveur ObjectServer. Pour obtenir des informations sur l'affichage des propriétés nécessitant un redémarrage du serveur ObjectServer, voir la liste de **conseils** qui suit.

Chaque fois que vous modifiez les valeurs des propriétés du serveur ObjectServer à l'aide de la commande ALTER SYSTEM SET ou de Netcool/OMNIbus, une liste de propriétés reflétant les changements opérés est ajoutée au bas du fichier de propriétés. Par conséquent, plusieurs entrées d'une même propriété peuvent coexister dans un fichier. La dernière entrée d'une propriété est prioritaire sur les entrées précédentes.

Astuces relatives à l'affichage des informations sur les propriétés du serveur ObjectServer :

- Vous pouvez interroger la table catalog.properties afin de visualiser les informations sur les propriétés du serveur ObjectServer. Par exemple, pour extraire une liste des propriétés qui ne peuvent pas être modifiées, entrez la requête suivante dans l'interface interactive SQL :

  ```
  Sélectionnez PropName dans catalog.properties, où IsModifyable=FALSE;
  ```

- Les modifications apportées à certaines propriétés ne prennent pas effet tant que vous n'avez pas redémarré le serveur ObjectServer. Pour extraire la liste de ces propriétés, entrez la requête suivante :

  ```
  Sélectionnez PropName dans catalog.properties, où IsImmediate=FALSE;
  ```

- Vous pouvez également visualiser les informations des propriétés du serveur ObjectServer dans l'interface Netcool/OMNIbus Administrator.

Spécification des options de ligne de commande du serveur ObjectServer

Lors de l'exécution du serveur ObjectServer à l'aide de la commande **nco_objserv**, vous pouvez spécifier un ensemble d'options de ligne de commande. Vous pouvez remplacer la valeur par défaut et le fichier de propriétés depuis la ligne de commande.

Les options de ligne de commande du serveur ObjectServer utilisent le format suivant :

```
nco_objserv [ -option [ valeur ] ... ]
```

Dans cette commande, *-option* est l'option de ligne de commande et la *valeur* est la valeur sur laquelle vous définissez l'option. Il n'est pas nécessaire de spécifier une valeur pour chaque option.

Vous pouvez ajouter des options de ligne de commande aux commandes **nco_objserv** du fichier de configuration de l'agent du processus.

Tâches associées:

«Affichage et modification des propriétés du serveur ObjectServer», à la page 158
Les propriétés du serveur ObjectServer permettent de déterminer son comportement. Vous pour les afficher et les modifier à l'aide de Netcool/OMNIbus Administrator. Vous ne pouvez pas ajouter des propriétés, mais uniquement éditer celles qui existent.

«Définition de processus, de services et d'hôtes pour le contrôle de processus», à la page 269
Pour exécuter le contrôle de processus, les processus, services et hôtes doivent être définis dans un fichier de configuration d'agent de processus. Au démarrage de l'agent de processus, ce dernier lit ce fichier afin d'établir les paramètres de configuration.

Référence associée:

«ALTER SYSTEM», à la page 356
La commande `ALTER SYSTEM` permet de modifier les paramètres par défaut et actuels du serveur ObjectServer en paramétrant les propriétés, d'arrêter le serveur ObjectServer, de supprimer des connexions utilisateur ou de sauvegarder le serveur ObjectServer.

«Table catalog.properties», à la page 437
La table catalog.properties contient des informations sur les propriétés du serveur ObjectServer.

«SELECT (scalaire)», à la page 401
La commande scalaire `SELECT` extrait les colonnes et lignes d'une table en fonction de critères spécifiques.

«SELECT (aggregate)», à la page 403
Une commande `SELECT` d'agrégation calcule le nombre de lignes et renvoie une valeur unique.

«SELECT (GROUP BY)», à la page 405
Vous pouvez utiliser une commande SELECT avec une clause GROUP BY pour regrouper en une seule ligne toutes les lignes ayant des valeurs identiques dans une colonne ou une combinaison de colonnes spécifiques. Vous pouvez également rechercher la valeur agrégée de chaque groupe de valeurs de colonne.

Propriétés et options de ligne de commande du serveur ObjectServer

Utilisez les propriétés ou les options de ligne de commande ObjectServer pour configurer les paramètres du serveur ObjectServer. Pour éviter les erreurs, ajoutez autant de propriétés que possible au fichier de propriétés plutôt que des options de ligne de commande. D'autres utilitaires sont fournis que vous pouvez utiliser pour chiffrer les valeurs de propriété.

Les propriétés et les options de ligne de commande du serveur ObjectServer sont décrites dans le tableau suivant. La propriété **Ipc.Timeout** est documentée dans ce tableau, même s'il s'agit d'une propriété de client ObjectServer tel que les passerelles.

Remarque : La propriété **Ipc.Timeout** décrite ici n'est pas une propriété d'ObjectServer mais est utilisée par plusieurs clients d'ObjectServer tels que les passerelles.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer

Propriété	Option de ligne de commande	Description
ActingPrimary TRUE | FALSE	Non applicable	Cette propriété est utilisée dans une configuration de reprise en ligne, dans laquelle des serveurs ObjectServer principal et de sauvegarde sont connectés par une passerelle bidirectionnelle ObjectServer, utilisée pour répliquer les données d'événement entre les deux serveurs ObjectServer. La propriété est mise à jour par des automatisations uniquement et ne doit *pas* être mise à jour manuellement dans le fichier de propriétés. **ActingPrimary** est définie sur TRUE sur le serveur ObjectServer de sauvegarde durant la période où ce dernier agit en tant que serveur principal. Sinon, elle est définie sur FALSE sur le serveur ObjectServer de sauvegarde. **ActingPrimary** est toujours définie sur TRUE sur le serveur ObjectServer principal.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
AlertSecurityModel *entier*	-alertsecuritymodel *entier*	Cette propriété détermine si la sécurité au niveau des lignes de groupe est appliquée dans la liste d'événements. Par défaut, la sécurité au niveau des lignes de groupe est désactivée (0). Dans ce cas : • Un membre du groupe Normal peut modifier une ligne affectée à ce groupe ou à l'utilisateur nobody. • Un membre du groupe Administrator peut modifier une ligne affectée à ce groupe, à l'utilisateur nobody ou à un membre du groupe Normal. Si la propriété **AlertSecurityModel** est activée (1), seuls les utilisateurs du groupe détenteur de la ligne peuvent modifier cette ligne. Dans ce cas, un membre du groupe Normal ou Administrator peut modifier une ligne affectée à un groupe auquel il appartient. Un membre du groupe System peut toujours modifier une ligne. Pour plus d'informations sur le système et les groupes par défaut, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.
AllowConnections TRUE | FALSE	-disallowconnections	Indique si les utilisateurs non root peuvent se connecter au serveur ObjectServer. Si l'option est paramétrée sur FALSE, aucune nouvelle connexion au serveur ObjectServer n'est autorisée. La valeur par défaut est TRUE.
AllowISQL TRUE | FALSE	-disallowisql	Indique si les connexions au serveur ObjectServer sont autorisées via l'interface SQL interactive. Si l'option est définie sur FALSE, aucun utilisateur ne peut se connecter via **nco_sql**. L'option par défaut est TRUE. Si l'option est paramétrée sur TRUE, elle peut être activée pour chaque utilisateur à l'aide de Netcool/OMNIbus Administrator.
AllowISQLWrite TRUE | FALSE	-disallowisqlwrite	Indique si les modifications apportées au serveur ObjectServer sont autorisées via l'interface SQL interactive. Si l'option est définie sur FALSE, aucun utilisateur ne peut modifier le serveur ObjectServer via **nco_sql**. La valeur par défaut est TRUE. Si l'option est définie sur TRUE, elle peut être activée pour chaque utilisateur à l'aide de Netcool/OMNIbus Administrator.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
AllowTimedRefresh TRUE | FALSE	-allowtimedrefresh TRUE | FALSE	Cette propriété indique si l'utilisateur peut activer l'actualisation temporisée dans l'onglet **Actualiser** de la fenêtre Préférences de liste d'événements. Si l'option est définie sur TRUE, les préférences de la liste d'événements peuvent être définies pour autoriser la mise à jour des informations d'alerte à un intervalle spécifié, au lieu d'attendre la notification des mises à jour depuis le serveur ObjectServer. La valeur par défaut est FALSE. Si l'option est définie sur FALSE, la case d'actualisation temporisée est grisée dans l'onglet **Actualiser** de la fenêtre Préférences de liste d'événements. L'actualisation temporisée est désactivée.
Auto.Debug TRUE | FALSE	-autodebug TRUE | FALSE	Si l'option est paramétrée sur TRUE, le débogage de l'automatisation est activé. La valeur par défaut est FALSE.
Auto.Enabled TRUE | FALSE	-autoenabled TRUE | FALSE	Si l'option est paramétrée sur TRUE, les automatisations sont activées. La valeur par défaut est TRUE.
Auto.StatsInterval *entier*	-autostatsinterval *entier*	Indique l'intervalle en secondes auquel le système d'automatisation collecte les statistiques. La valeur par défaut est 60. Les statistiques sont regroupées à moins que l'option de ligne de commande -autoenabled ne soit définie sur FALSE. Dans ce cas, toutes les automatisations sont désactivées.
BackupObjectServer TRUE | FALSE	-backupserver	Fournit une fonction de reprise par restauration avec clients de bureau, sondes, serveur proxy et passerelle ObjectServer. La valeur par défaut est FALSE ; les clients de bureau, le serveur proxy et les passerelles sont supposés être connectés à un serveur ObjectServer principal. Lorsque l'option est paramétrée sur TRUE, les clients de bureau, les sondes, le serveur proxy et les passerelles sont informés qu'ils sont connectés au serveur ObjectServer de sauvegarde dans une paire de reprise par restauration. Si c'est le cas, les clients de bureau, les sondes, le serveur proxy et les passerelles vérifient automatiquement la récupération du serveur ObjectServer principal dans la paire de reprise en ligne et effectuent une reprise par restauration au redémarrage du serveur.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
ClientHeartbeatDisable TRUE | FALSE	-clienthbdisable TRUE | FALSE	Désactive le système de pulsation client si l'option est paramétrée sur TRUE. Cela pousse un client connecté à dépasser le délai si le serveur ObjectServer est occupé, par exemple, pendant une resynchronisation ou une automatisation de passerelle. Le paramètre par défaut FALSE active la pulsation et évite tout délai d'attente client non valide ou obsolète. Si le serveur ObjectServer est actif mais occupé, ce paramètre pousse le serveur ObjectServer à envoyer un message contextuel à un client connecté avec des détails du type de traitement en cours.
ClientHeartbeatRate *non signé*	-clienthbrate *non signé*	Définit le taux en secondes d'une pulsation client. Ce taux définit le temps que doit attendre un client pour obtenir une réponse du serveur ObjectServer avant un dépassement de délai. La valeur par défaut est 10.
ConfigCryptoAlg *chaîne*	N/D	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 : • 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**.
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**.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
Connections *entier*	-connections *entier*	Définit le nombre maximal de connexions disponibles pour les clients de bureau, les sondes et les passerelles. La valeur maximale est 1024. La valeur par défaut est 256. Le système peut utiliser jusqu'à deux connexions. Si l'ObjectServer atteint la valeur spécifiée, aucune connexion n'est possible tant que certaines des connexions précédemment établies n'ont pas été supprimées.
DeleteLogFile *chaîne*	-DELETES *chaîne*	Chemin et nom du fichier journal de suppression où toutes les commandes de suppression sont enregistrées si la consignation de la suppression est activée. Par défaut, les suppressions sont consignées dans le fichier $NCHOME/omnibus/log/*nom_serveur*_deletes_file.log*n*.
DeleteLogging TRUE | FALSE	-DELETE_LOGGING TRUE | FALSE	Lorsque l'option a pour valeur TRUE, la consignation de suppression est activée. La valeur par défaut est FALSE.
DeleteLogLevel *entier*	-DELETES_LEVEL *entier*	Le niveau de consignation détermine la quantité d'informations envoyées au fichier journal de suppression. Les paramètres possibles sont les suivants : • <0 : pas de consignation • 0 : type de client (ID d'application, par exemple, ctisql pour **nco_sql**) et exécution de langage SQL. Il s'agit du niveau de consignation par défaut. • 1 : heure, ID utilisateur, type de client et exécution SQL.
DeleteLogSize *entier*	-DELETES_SIZE *entier*	Taille maximale du fichier journal de suppression. Lorsque le fichier journal *nom_serveur*_deletes_file.log1 atteint la taille spécifiée, il est renommé *nom_serveur*_deletes_file.log2 et un nouveau fichier journal, *nom_serveur*_deletes_file.log1, est créé. Lorsque le nouveau fichier atteint sa taille maximale, l'ancien fichier est supprimé et le processus est répété. La sortie d'une commande de suppression unique n'est jamais divisée entre fichiers journaux. Par conséquent, la taille des fichiers journaux peut être supérieure à la taille spécifiée. La taille du fichier journal par défaut est de 1024 Ko.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
DTMaxTopRows *entier*	-dtmaxtoprows *entier*	Nombre maximum de lignes qu'un administrateur peut spécifier pour restreindre le nombre de lignes qu'un utilisateur de liste d'événements peut visualiser. La valeur par défaut est 100.
Granularity *entier*	-granularity *entier*	Contrôle l'intervalle de mise à jour en secondes des diffusions IDUC vers les bureaux et les passerelles. Réduisez cette valeur pour augmenter la fréquence de mise à jour vers les clients. L'intervalle par défaut est 60 secondes. Si vous devez effectuer des actualisations IDUC plus fréquentes, configurez le taux d'actualisation côté client, par exemple dans le fichier de configuration des passerelles. Si la propriété **Profile** est paramétrée sur TRUE, à la fin de chaque période de granularité, l'ObjectServer écrit un récapitulatif dans le journal de profilage du temps passé sur chaque connexion client.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
GWDeduplication *entier*	-gwdeduplication *entier*	Cette propriété contrôle le comportement en cas de tentative de réinsertion d'un événement existant dans le serveur ObjectServer. Les valeurs possibles sont les suivantes : 0 : lorsque l'option possède cette valeur (par défaut) : • Si le client n'est pas une passerelle, le nombre figurant dans la zone Tally est incrémenté. • Les zones LastOccurrence, Summary et AlertKey sont mises à jour avec les nouvelles valeurs et les zones StateChange et InternalLast sont mises à jour avec l'heure actuelle. • Si la nouvelle gravité est supérieure à 0 (effacer) et que l'ancienne gravité était égale à 0, l'alerte est également mise à jour avec la nouvelle valeur de gravité. 1 : si le client est une passerelle, le nouvel événement remplace l'ancien. 2 : si le client est une passerelle, le nouvel événement est ignoré . 3 : lorsque l'option possède cette valeur : • Pour tous les clients, le nombre figurant dans la zone Tally est incrémenté. • Les zones LastOccurrence, Summary et AlertKey sont mises à jour avec les nouvelles valeurs et les zones StateChange et InternalLast sont mises à jour avec l'heure actuelle. • Si la nouvelle gravité est supérieure à 0 (effacer) et que l'ancienne gravité était égale à 0, l'alerte est également mise à jour avec la nouvelle valeur de gravité.
Non applicable	-help	Affiche l'aide sur les options de ligne de commande et quitte.
Iduc.ListeningHostname *chaîne*	-listeninghostname *chaîne*	Définit le nom d'hôte des clients à utiliser pour localiser le serveur ObjectServer. Sur les systèmes utilisant le serveur de noms de domaine, le nom retourné par une machine hôte en interne peut être différent du nom utilisé pour celle-ci sur le réseau. Par exemple, un ordinateur nommé sfo peut être nommé sfo.bigcorp.org sur le réseau. Pour permettre aux clients de se connecter au serveur ObjectServer sur sfo, définissez cette propriété sur sfo.bigcorp.org. Le nom par défaut est le nom de l'ordinateur local, rapporté par la commande **hostname**.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
Iduc.ListeningPort *entier*	-listeningport *entier*	Définit le port de la connexion de communication IDUC. Il s'agit du port sur lequel le serveur ObjectServer envoie des mises à jour à chaque liste d'événements et chaque passerelle. Si ce port n'est pas spécifié, il est sélectionné aléatoirement par le serveur ObjectServer à partir des numéros de port inutilisés et disponibles. La valeur par défaut est 0 (port suivant disponible). Vous pouvez également spécifier le port dans le fichier /etc/services de la machine hôte.
Ipc.Timeout *entier*	-ipctimeout *entier*	Définit le temps d'attente en secondes d'une application client ou d'un utilitaire pour une réponse d'un serveur ObjectServer, lorsqu'il tente d'envoyer des alertes. Si ce temps est dépassé ou si le serveur ObjectServer est inactif, une erreur de communication survient et l'alerte est envoyée au fichier de mémoire cache. La valeur par défaut est 60 secondes. **Remarque :** Cette propriété n'est pas une propriété d'ObjectServer mais est utilisée par plusieurs clients d'ObjectServer tels que les passerelles.
LogFilePoolSize *entier*	-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 paramétrée sur TRUE. La taille du pool est comprise entre 2 et 99. Le nombre est défini par défaut sur 10.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
LogFileUsePool TRUE | FALSE	-logfileusepool	Indique si un pool de fichiers journaux doit être utilisé pour la consignation des messages. Si l'option est paramétrée sur TRUE, le système de consignation ouvre le nombre de fichiers spécifiés pour le pool au démarrage, puis les maintient ouverts pendant son exécution. Pour définir le nombre de fichiers du pool, utilisez la propriété **LogFilePoolSize**. Lorsqu'un fichier du pool atteint sa taille maximale (tel que cela est spécifié par la propriété **MaxLogFileSize**), le système de consignation écrit dans le fichier suivant. Une fois que tous les fichiers du pool ont atteint leur taille maximale, le système de consignation tronque le premier fichier du pool et commence à y réécrire. Les fichiers du pool sont nommés en suivant le format *nom_serveur*.log_*ID*, où *ID* est un nombre à deux chiffres commençant à 01 et se terminant au nombre maximal spécifié pour la propriété **LogFilePoolSize**. Lorsque le système de consignation démarre pour utiliser un fichier de pool, le système écrit dans le numéro de fichier disponible le plus bas, quel que soit le fichier dans lequel il écrivait lors de sa dernière exécution. La valeur par défaut est FALSE. Lorsque l'option est définie sur FALSE, le fichier par défaut *nom_serveur*.log est renommé *nom_serveur*.log_OLD et un nouveau fichier journal démarre lorsque la taille maximale est atteinte. Si le fichier ne peut pas être renommé, par exemple en raison d'un verrou en lecture dans le fichier _OLD, et que la propriété **LogFileUseStdErr** est définie sur FALSE, le système de consignation démarre automatiquement à l'aide d'un pool de fichiers journaux.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
LogFileUseStdErr TRUE | FALSE	-logfileusestderr	Indique si stderr doit être utilisé comme flux de sortie pour les messages de consignation. La valeur par défaut est TRUE. Si ce paramètre a pour valeur TRUE et que le serveur ObjectServer est exécuté à partir de la ligne de commande, les messages sont consignés dans la console si le serveur ObjectServer ne peut pas écrire dans le fichier journal par défaut. Si ce paramètre est défini sur TRUE et que le serveur ObjectServer s'exécute en tant que service Windows, les messages sont écrits dans un fichier nommé %NCHOME%\omnibus\log\nco_objserv*.err si le serveur ObjectServer ne peut pas écrire dans le fichier journal par défaut. Si ce paramètre est défini sur FALSE et que le serveur ObjectServer ne peut pas écrire dans le fichier journal par défaut, les messages sont écrits dans un pool de fichiers journaux, tel que cela est défini par la propriété **LogFileUsePool**. **Remarque :** Le paramètre de propriété **LogFileUsePool** est prioritaire sur le paramètre **LogFileUseStdErr**.
MaxLogFileSize *entier*	-maxlogfilesize *entier*	Indique la taille maximale que peut atteindre le fichier journal en Ko. La valeur par défaut est 1024 Ko. Une fois la taille spécifiée atteinte, le fichier *nom_serveur*.log est renommé *nom_serveur*.log_OLD et un nouveau fichier journal démarre. Lorsque le nouveau fichier atteint la taille maximale, il est renommé et le processus redémarre.
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 : message fatal uniquement • error : messages fatal et error • warn : messages fatal, error, et warn • info : messages fatal, error, warn, et info • debug : messages fatal, error, warn, info et debug **Conseil :** La valeur de la *chaîne* peut être en majuscules et/ou en minuscules.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
MessageLog *chaîne*	-messagelog *chaîne*	Indique le chemin d'accès à l'emplacement de consignation des messages. Le chemin d'accès par défaut est $NCHOME/omnibus/log/NCOMS.log. Sous Windows, si le système ne peut pas écrire dans le fichier journal spécifié (par exemple, en cas d'erreur fatale), il écrit l'erreur dans un fichier nommé %NCHOME%\omnibus\log\nco_objserv*.err. **Remarque :** Le serveur ObjectServer doit être exécuté en tant que service. Si ce n'est pas le cas, il ne peut pas écrire les erreurs dans un fichier.
Non applicable	-name *chaîne*	Définit le nom du serveur ObjectServer, qui doit être unique. Il s'agit du nom qui est configuré dans l'éditeur de serveur. La valeur par défaut est NCOMS.
NHttpd.AccessLog *chaîne*	-nhttpd_accesslog *chaîne*	Indique le nom et l'emplacement du fichier journal dans lequel le serveur consigne toutes les demandes qu'il traite. Le fichier par défaut est $OMNIHOME/log/NCOMS_http_access.log.
NHttpd.Authentication Domain *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.
Nhttpd.BasicAuth *chaîne*	-nhttpd_basicauth *chaîne*	Indique la combinaison de mot de passe-utilisateur autorisée pour les connexions à l'interface HTTP, où le mot de passe est un hachage AES du mot de passe. La valeur par défaut est ''.
NHttpd.ConfigFile *chaîne*	-nhttpd_configfile *chaîne*	Indique le chemin d'accès au fichier de configuration JSON. La valeur par défaut est $OMNIHOME/etc/libnhttpd.json qui active les paramètres mimeType et les en-têtes HTTP dans les fichiers de réponse HTTP. Pour plus d'informations sur le fichier de configuration libnhttpd.json, voir Configuration de l'interface HTTP ObjectServer avancée
NHttpd.DocumentRoot *chaîne*	-nhttpd_docroot *chaîne*	Indique la racine de document du service Web imbriqué. La valeur par défaut est $OMNIHOME/etc/restos/docroot.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
NHttpd.EnableFileServing TRUE | FALSE	-nhttpd_enablefs TRUE | FALSE	Cette propriété permet d'activer la prise en charge de fichier par défaut par l'ObjectServer. Cela permet à l'ObjectServer de faire office de serveur HTTP simple prenant en charge les fichiers du système de fichiers local. La valeur par défaut est FALSE.
NHttpd.EnableHTTP TRUE | FALSE	-nhttpd_enablehttp TRUE | FALSE	Permet l'utilisation du port HTTP. La valeur par défaut est FALSE.
NHttpd.ExpireTimeout *non signé*	-nhttpd_exptimeout *non signé*	Indique la durée d'inactivité, en secondes, d'une connexion HTTP 1.1 avant qu'elle soit supprimée. La valeur par défaut est 15.
NHttpd.ListeningHostname *chaîne*	-nhttpd_hostname *chaîne*	Indique le nom d'hôte ou l'adresse IP d'écoute pouvant être utilisée comme partie nom d'hôte d'un URI de l'interface HTTP ou HTTP de l'ObjectServer. La valeur par défaut est localhost.
NHttpd.ListeningPort *entier*	-nhttpd_port *entier*	Indique le port sur lequel le serveur ObjectServer écoute les demandes HTTP. La valeur par défaut est 0.
NHttpd.NumWorkThreads *entier*	-nhttpd_numworkthrs *entier*	Indique le nombre d'unités d'exécution dans le pool d'unités d'exécution de gestion client. La valeur par défaut est 5.
NHttpd.SSLEnable TRUE | FALSE	-nhttpd_sslenable TRUE | FALSE	Permet l'utilisation du support SSL. La valeur par défaut est FALSE.
NHttpd.SSLListeningPort *entier*	-nhttpd_sslport *entier*	Indique le port sur lequel le serveur ObjectServer écoute les demandes HTTPS. La valeur par défaut est 0.
NHttpd.SSLCertificate *entier*	-nhttpd_sslcert *entier*	Indique le nom du certificat SSL du serveur. La valeur par défaut est ''.
NHttpd.SSLCertificatePwd *chaîne*	-nhttpd_sslcertpwd *chaîne*	**Remarque :** Cette propriété est obsolète dans Tivoli Netcool/OMNIbus V8.1. Si vous définissez cette propriété, elle n'aura aucun impact.
NRestOS.Enable TRUE | FALSE	-nrestosenable TRUE | FALSE	Active l'interface HTTP et l'interface OSLC pour le serveur ObjectServer. La valeur par défaut est FALSE, c'est-à-dire que les interfaces sont désactivées.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
NRestOS.OSLCRDFMsgFormat *chaîne*	nrestososlcrmf *chaîne*	Définissez cette propriété sur la chaîne MIGRATION pour redévelopper les utilitaires basés sur l'interface OSLC de l'ObjectServer afin que la liste de références des ressources membres soit générée dans une instance de ressource **Collection** plutôt que dans une instance de ressource **ResponseInfo** dans le contenu RDF/XML de la fonction de requête d'événement, de détail et de journal. La valeur MIGRATION signifie que la liste de références des ressources membres est générée à la fois dans une instance de ressource **Collection** et dans une instance de ressource **ResponseInfo**. Redéveloppez vos utilitaires OSLC pour générer la liste de références des ressources membres uniquement dans l'instance de ressource **Collection**. Une fois supprimé le code qui génère la liste dans l'instance de ressource **ResponseInfo**, réinitialisez cette propriété.
NRestOS.OSLCResource ConfigFile *chaîne*	-nrestososlcrescfg *chaîne*	Chemin d'accès au fichier de configuration OSLC. Ce fichier JSON définit la façon dont les colonnes du schéma ObjectServer sont mappées avec des propriétés du domaine des événements. Le chemin d'accès par défaut est $OMNIHOME/etc/restos/resourcecfg.json.
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.
PA.Name *chaîne*	-pa *chaîne*	Définit le nom de l'agent de contrôle du processus. Ce nom doit se composer d'un maximum de 29 lettres majuscules et ne peut pas commencer par un entier. Lorsqu'une procédure externe est exécutée à partir d'une automatisation, le serveur ObjectServer peut démarrer un processus externe. Pour démarrer le processus, le serveur ObjectServer contacte un agent de contrôle de processus. Le nom par défaut de l'agent de contrôle de processus est NCO_PA.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
PA.Password *chaîne*	-papassword *chaîne*	Indique le mot de passe de l'utilisateur se connectant à un agent de contrôle de processus pour exécuter des procédures externes dans des automatisations. La valeur par défaut est ''. **Remarque :** Si vous exécutez des procédures externes, le démon de contrôle de processus doit être en mesure d'authentifier l'utilisateur. Par conséquent, une combinaison valide pouvant être authentifiée par le démon doit être spécifiée dans les valeurs de *chaîne* des propriétés **PA.Username** et **PA.Password**. Sinon, la connexion à l'agent de contrôle de processus et l'exécution de la procédure externe échoueront.
PA.Username *chaîne*	-pausername *chaîne*	Indique le nom d'utilisateur permettant de se connecter à un agent de contrôle pour exécuter des procédures externes dans des automatisations. Une valeur doit être spécifiée lorsque l'agent de contrôle de processus s'exécute en mode sécurisé. La valeur par défaut est root. Sous Windows, spécifiez le nom d'utilisateur d'un compte local, compte de domaine ou compte UPN valide. Sous UNIX, tout utilisateur souhaitant accéder au système de contrôle de processus doit être membre d'un groupe d'utilisateurs UNIX (par défaut, ncoadmin) que vous identifiez en tant que groupe administratif pour cet objectif. Notez que si vous utilisez des modules PAM (Pluggable Authentication Modules) dans le cadre de l'authentification, les utilisateurs n'ont pas besoin d'être membres d'un groupe d'utilisateurs UNIX tel que ncoadmin pour accéder au système de contrôle de processus.
PasswordEncryption *chaîne*	-pwdenc *chaîne*	Cette propriété définit le schéma de chiffrement utilisé pour coder les mots de passe utilisateur stockés dans le serveur ObjectServer. Les valeurs possibles sont les suivantes : • DES : chiffrement Data Encryption Standard (DES). Seuls les huit premiers caractères d'un mot de passe DES sont lus. Les autres caractères sont ignorés. • AES : chiffrement Advanced Encryption Standard (AES128). Seuls les 16 premiers caractères d'un mot de passe AES128 sont lus. Les autres caractères sont ignorés. En mode FIPS 140–2, l'option AES est mandatée par le système. Pour les installations non FIPS 140-2, la valeur par défaut est DES. En mode FIPS 140-2, la valeur par défaut est AES.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
PasswordFormat *chaîne*	-pwdformat *chaîne*	La propriété définit le format des mots de passe utilisateur. Elle ne fonctionne que quand la propriété **RestrictPasswords** est paramétrée sur TRUE. Indiquez la valeur de cette propriété sous la forme d'un ensemble de valeurs de type entier séparées par des signes deux-points. Chaque valeur définit une exigence de mot de passe. Le format est le suivant : *long_min*:*nbre_alpha*:*nbre_num*:*nbre_ponct* où : • *lon_min* est la longueur du mot de passe. • *nbre_alpha* est le nombre minimum de caractères alphabétiques. • *nbre_num* est le nombre minimum de caractères numériques. • *nbre_ponct* est le nombre minimum de caractères de ponctuation. La valeur par défaut de 8:1:1:0 indique qu'un mot de passe doit comporter au moins huit caractères et contenir au moins un caractère alphabétique et au moins un caractère numérique. Le mot de passe n'a pas besoin de contenir de caractères de ponctuation. **Remarque :** Les exigences de caractère alphabétique, numérique et de ponctuation doivent être satisfaites au sein du nombre de caractères spécifié par la longueur minimale du mot de passe. La valeur par défaut de 8:1:1:0 doit comporter au moins un caractère alphabétique et un caractère numérique dans les 8 premiers caractères de la chaîne de mot de passe. Par exemple, si la propriété a pour valeur 8:1:1:0 et qu'un utilisateur indique le mot de passe abcdefgh590675, ce dernier est rejeté car les 8 premiers caractères ne contiennent aucun caractère numérique. Une fois cette propriété définie, l'ObjectServer valide tous les mots de passe nouveaux ou modifiés par rapport à cette exigence et les mots de passe ne remplissant pas cette exigence sont rejetés. Les mots de passe existants ne sont pas validés.
ProfileStatsInterval *entier*	-profilestatsinterval *entier*	Spécifie l'intervalle en secondes auquel lequel le profileur écrit des informations dans le fichier journal du profil. La valeur par défaut est 60 secondes.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
Profile TRUE | FALSE	-profile	Contrôle le profilage du serveur ObjectServer. Si l'option a pour valeur TRUE, la durée nécessaire aux clients pour exécuter le langage SQL est consignée dans la table catalog.profiles. La valeur par défaut est TRUE. Les statistiques du profil sont également consignées dans le fichier $NCHOME/omnibus/log/*nom_serveur*_profiler_report.log*n* à chaque intervalle des statistiques de profil.
Props.CheckNames TRUE | FALSE	Non applicable	Lorsque l'option a pour valeur TRUE, le serveur ObjectServer ne s'exécute pas si une propriété spécifiée est non valide. La valeur par défaut est TRUE.
Non applicable	-propsfile *chaîne*	Définit le nom du fichier de propriétés ObjectServer. Le nom par défaut est *nom_serveur*.props, où *nom_serveur* est défini par l'option -name.
RegexpLibrary *chaîne*	-regexplib *chaîne*	Définit la bibliothèque d'expression régulière à utiliser pour le serveur ObjectServer. Les valeurs possibles sont les suivantes : NETCOOL et TRE. La valeur par défaut NETCOOL est utile pour le traitement d'un caractère codé à un seul octet et fournit des performances système optimales. Pour utiliser la syntaxe d'expression régulière étendue sur les langues codées sur un seul octet ou sur plusieurs octets, définissez la valeur sur TRE. Ce paramètre réduit les performances système.
RestrictionFiltersAND TRUE | FALSE	-restrictfiltersand	Contrôle la concaténation des filtres de restriction des utilisateurs et des groupes. Cette propriété est définie pour le système et non par l'utilisateur. Les modifications apportées au paramètre de propriété prend uniquement effet après le redémarrage du serveur ObjectServer. Les valeurs de cette propriété sont les suivantes : • TRUE : tous les filtres de restriction sont combinés à l'aide de l'opérateur AND. L'option par défaut est TRUE. Exemple : user_rf AND groupe1 AND groupe2 • FALSE : tous les filtres de restriction sont combinés à l'aide de l'opérateur OR. Exemple : user_rf OR groupe1 OR groupe2
RestrictionUpdateCheck TRUE | FALSE	-norestrictionupdatecheck	Lorsque cette option a pour valeur FALSE, les utilisateurs pour lesquels des filtres de restriction sont appliqués peuvent mettre à jour des alertes qui n'apparaissent pas dans leur vue après la mise à jour. La valeur par défaut est TRUE.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
RestrictPasswords TRUE | FALSE	-restrictpasswords TRUE | FALSE	Lorsque cette option a pour valeur TRUE, le format spécifié par la propriété **PasswordFormat** est appliqué. La valeur par défaut est FALSE.
RestrictProxySQL TRUE | FALSE	-restrictproxysql	Lorsque l'option a pour valeur TRUE, les connexions à partir d'un serveur proxy sont restreintes dans les commandes SQL du serveur ObjectServer qui peuvent être exécutées. Les seules modifications pouvant être apportées aux données du serveur ObjectServer sont les insertions situées dans les tables alerts.status et alerts.details. La valeur par défaut est FALSE. Si l'option est paramétrée sur FALSE, les connexions d'un serveur proxy peuvent exécuter n'importe quelle commande SQL du serveur ObjectServer.
Sec.AuditLevel *chaîne*	-secauditlevel *chaîne*	Indique le niveau d'audit de sécurité effectué. Les valeurs possibles sont debug, info, warn et error. La valeur par défaut est warn. Les niveaux debug et info génèrent des messages de réussite et d'échec d'authentification, tandis que les niveaux warn et error génèrent des messages exclusifs aux échecs d'authentification.
Sec.AuditLog *chaîne*	-secauditlog *chaîne*	Indique le fichier dans lequel les informations d'audit sont écrites. La valeur par défaut est $NCHOME/omnibus/log/*nom_serveur*/audit.log.
Sec.ExternalAuthentication *chaîne*	-authenticate *chaîne*	Contrôle le module d'authentification utilisé pour authentifier les utilisateurs. Les valeurs sont les suivantes : • none : utilisez cette valeur pour effectuer une authentification d'utilisateur dans les noms d'utilisateur et les mots de passe stockés dans le serveur ObjectServer. Ce paramètre désactive l'authentification externe. • LDAP : utilisez cette valeur pour authentifier en externe les utilisateurs dont les données d'identification sont stockées dans un référentiel de protocole LDAP (Lightweight Directory Access Protocol). • PAM : utilisez cette valeur pour authentifier les utilisateurs via le module PAM Tivoli Netcool/OMNIbus fourni ou pour authentifier en externe les utilisateurs à l'aide d'une configuration de module PAM tierce (telle que le système d'exploitation ou une configuration de module PAM définie pour utiliser des données d'identification LDAP). Sous UNIX et Linux, la valeur par défaut est PAM. Sous Windows, la valeur par défaut est none.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
SecureMode TRUE | FALSE	-secure	Définit le mode de sécurité du serveur ObjectServer. Si l'option a pour valeur TRUE, le serveur ObjectServer authentifie les requêtes de connexion à la sonde, à la passerelle et au serveur ObjectServer avec un nom d'utilisateur et un mot de passe. Les autres requêtes de connexion au client sont toujours authentifiées avec un nom d'utilisateur et un mot de passe.
Store.LocalizedSort TRUE | FALSE	-storelocalesort	Définit si le tri localisé est activé ou non. La valeur par défaut est FALSE (désactive le tri localisé en faveur des comparaisons de chaîne (libc) standard)). Par exemple, Å est traité comme une variante de A et le tri placera ces deux caractères à proximité l'un de l'autre. Utilisez ce paramètre par défaut pour des performances système optimales. Définissez la valeur sur TRUE pour activer le tri localisé. Par exemple, dans un environnement local danois, Å est traité comme une lettre distincte triée juste après Z. Notez que le fait de spécifier ce paramètre entraîne une baisse des performances du système.
Store.LocalizedSortCaseSensitive *chaîne*	-storecasesort *chaîne*	Contrôle la sensibilité de la casse dans le tri localisé. Cette propriété ne s'applique que lorsque l'option **Store.LocalizedSort** a pour valeur TRUE. Les valeurs sont les suivantes : • OFF : le tri est insensible à la casse et les majuscules et les minuscules sont classées en fonction de leur poids de tri tertiaire. • UPPERFIRST : les caractères majuscules sont triés avant les caractères minuscules. Par exemple, Account est trié avant account. • LOWERFIRST : les caractères minuscules sont triés avant les caractères majuscules. Par exemple, account est trié avant Account. La valeur par défaut est OFF.
UniqueLog TRUE | FALSE	-uniquelog	Si l'option a pour valeur TRUE, le fichier journal est uniquement nommé en ajoutant l'ID de processus du serveur ObjectServer dans le nom du fichier journal par défaut. Par exemple, si le serveur ObjectServer NCOMS s'exécute en tant que processus 1234, le fichier journal est nommé NCHOME/omnibus/log/ NCOMS.1234.log. La valeur par défaut est FALSE. Si la propriété **MessageLog** a pour valeur stderr oustdout, la propriété **UniqueLog** est ignorée.

Tableau 3. Propriétés et options de ligne de commande du serveur ObjectServer (suite)

Propriété	Option de ligne de commande	Description
N/D	Windows `-utf8enabled TRUE | FALSE`	Contrôle le codage des données communiquées ou gérées par cette application sous Windows. Définissez la valeur `-utf8enabled` sur `TRUE` pour exécuter l'application au codage UTF-8. La valeur par défaut est `FALSE` ; ce qui implique l'utilisation de la page de codes par défaut du système. **Important :** Bien qu'une propriété **`UTF8Enabled`** soit disponible, la définition de cette propriété sur `TRUE` en vue d'activer le codage UTF-8 n'a aucun effet. Pour une exécution sous Windows au codage UTF-8, vous devez toujours utiliser l'option de ligne de commande `-utf8enabled`. **Remarque :** Lors de l'exécution au codage UTF-8, ne définissez pas la propriété **`OldTimeStamp`** sur `TRUE`.
`WTPasswordCheck` *chaîne*	`-wtpasswordcheck` *chaîne*	Contrôle le mode de vérification des mots de passe par le serveur ObjectServer lorsqu'ils sont transmis par les clients. Cette propriété est utile lorsque les utilisateurs du serveur ObjectServer sont authentifiés en externe. Les valeurs possibles de la propriété sont les suivantes : • `plain` : les mots de passe sont supposés être en texte normal (s'applique uniquement aux clients Netcool/Webtop 1.3). • `encrypted` : les mots de passe sont supposés être codés (s'applique uniquement aux clients Netcool/Webtop 1.3). • `default` : les mots de passe sont initialement supposés être en texte normal. Si la vérification du serveur ObjectServer échoue, les mots de passe sont supposés être codés (s'applique uniquement aux clients Netcool/Webtop 1.3). • `allplain` : les mots de passe sont supposés être en texte normal. Si la connexion échoue avec le mot de passe en texte en clair, il n'y a pas de seconde tentative. (s'applique uniquement aux clients autres que Netcool/Webtop 1.3). • `allencrypted` : les mots de passe sont supposés être codés et décodés avant l'authentification. Si la connexion échoue avec le mot de passe en texte en clair, une seconde tentative est exécutée. (s'applique uniquement aux clients autres que Netcool/Webtop 1.3).
Non applicable	`-version`	Affiche les informations de version du serveur ObjectServer et quitte.

Remarque : La commande **`nco_objserv`** comprend des propriétés avancées qui ne doivent être utilisées que sous la direction du service de support logiciel IBM. Ces propriétés ne sont pas documentées.

Exécution du serveur ObjectServer en mode sécurisé

Vous pouvez exécuter le serveur ObjectServer en mode sécurisé. Lorsque vous spécifiez l'option de ligne de commande `-secure`, le serveur ObjectServer authentifie les connexions de la sonde, de la passerelle et du serveur proxy via un nom d'utilisateur et un mot de passe.

Lorsqu'une demande de connexion est envoyée, le serveur ObjectServer transmet un message d'authentification. La sonde, la passerelle ou le serveur proxy doit fournit le nom d'utilisateur et le mot de passe corrects.

Si vous ne spécifiez pas l'option `-secure`, les demandes de connexion à la sonde, à la passerelle et au serveur proxy ne sont pas authentifiées.

Remarque : Les connexions à partir d'autres clients, tels que la listes d'événements et l'interface interactive SQL, sont toujours authentifiées.

Lors de la connexion à un serveur ObjectServer sécurisé :

- Dans le fichier de propriétés de chaque sonde ou serveur proxy établissant une connexion, les propriétés **AuthUserName** et **AuthPassword** doivent être spécifiées.
- Les propriétés **Gate.Writer.Username**, **Gate.Writer.Password**, **Gate.Reader.Username** et **Gate.Reader.Password** doivent être spécifiées dans chaque passerelle unidirectionnelle utilisant un fichier de propriétés. Les propriétés **Gate.ObjectServerA.Username**, **Gate.ObjectServerA.Password**, **Gate.ObjectServerB.Username** et **Gate.ObjectServerB.Password** doivent être spécifiées dans chaque passerelle bidirectionnelle utilisant un fichier de propriétés. Les commandes AUTH_USER et AUTH_PASSWORD doivent être spécifiées dans le fichier de configuration de passerelle de chaque passerelle utilisant un fichier de configuration.

Si le nom d'utilisateur et le mot de passe sont incorrects, le serveur ObjectServer transmet un message d'erreur et refuse la connexion.

Vous pouvez choisir tout nom d'utilisateur valide pour la propriété **AuthUserName**, **Gate.Writer.Username**, **Gate.Reader.Username**, **Gate.ObjectServerA.Username** et **Gate.ObjectServerB.Username** ou la commande AUTH_USER.

Les détails de chiffrement du mot de passe en modes FIPS 140–2 mode et non-FIPS 140–2 sont décrits dans la table suivante.

Tableau 4. Chiffrement du mot de passe en modes FIPS 140–2 mode et non-FIPS 140–2

Mode	Action
mode FIPS 140–2	Lorsque le mode FIPS 140–2 est activé, les mots de passe peuvent être spécifiés en texte normal ou dans un format chiffré. Vous pouvez coder les mots de passe à l'aide du chiffrement des valeurs de propriété de la façon suivante : 1. Si vous ne disposez pas encore d'une clé pour le chiffrement du mot de passe, créez-en une en exécutant l'utilitaire **nco_keygen** qui se trouve dans le répertoire `$NCHOME/omnibus/bin`. 2. Exécutez l'utilitaire **nco_aes_crypt** pour coder le mot de passe avec la clé générée par l'utilitaire **nco_keygen**. L'utilitaire **nco_aes_crypt** se trouve également dans le répertoire `$NCHOME/omnibus/bin`. Notez que vous devez spécifier `AES_FIPS` comme algorithme à utiliser pour le chiffrement du mot de passe. 3. Ouvrez le fichier de propriétés dans lequel vous souhaitez ajouter le mot de passe chiffré et spécifiez cette sortie chiffrée pour le paramètre **AuthPassword**. **Remarque :** Vous devez également définir la propriété **ConfigKeyFile** dans le fichier de clés que vous avez spécifié lors de l'exécution de **nco_aes_crypt**, et définir la propriété **ConfigCryptoAlg** dans l'algorithme de chiffrement utilisé.
mode non-FIPS 140–2	Lorsque le mode FIPS 140–2 est désactivé, les mots de passe peuvent être spécifiés en texte normal ou dans un format chiffré. Cependant, le client transmet toujours des informations de connexion chiffrées indépendamment du chiffrement du mot de passe utilisé dans le fichier de propriétés. Vous pouvez coder les mots de passe à l'aide de l'utilitaire **nco_g_crypt** ou du chiffrement de la valeur de la propriété de la façon suivante : • Pour coder un mot de passe à l'aide de l'utilitaire **nco_g_crypt**, exécutez la commande suivante : `$NCHOME/omnibus/bin/nco_g_crypt` *mdp_texte_clair* Dans cette commande, *mdp_texte_clair* représente la forme non chiffrée du mot de passe. L'utilitaire **nco_g_crypt** prend le mot de passe non chiffré et l'affiche dans une version chiffrée. Ouvrez le fichier de propriétés dans lequel vous souhaitez ajouter le mot de passe chiffré et spécifiez cette sortie chiffrée pour le paramètre **AuthPassword**. • Pour coder un mot de passe à l'aide du chiffrement de la valeur de la propriété, vous avez besoin d'une clé générée à l'aide de l'utilitaire **nco_keygen**. Vous pouvez alors exécuter **nco_aes_crypt** pour coder le mot de passe avec la clé. Notez que vous pouvez spécifier `AES_FIPS` ou `AES` comme algorithme de chiffrement du mot de passe. 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. Ouvrez le fichier dans lequel vous souhaitez ajouter le mot de passe chiffré et spécifiez cette sortie chiffrée pour le paramètre **AuthPassword**. **Remarque :** Vous devez également définir la propriété **ConfigKeyFile** dans le fichier de clés que vous avez spécifié lors de l'exécution de **nco_aes_crypt**, et définir la propriété **ConfigCryptoAlg** dans l'algorithme de chiffrement utilisé.

Un mot de passe codé avec **nco_g_crypt** est spécifié de la même manière qu'un mot de passe non codé lors de la connexion au serveur ObjectServer. Le serveur ObjectServer détecte automatiquement un mot de passe codé et exécute le déchiffrement nécessaire pour vérifier le mot de passe lors de l'authentification.

Avertissement : Les mots de passe codés avec **nco_g_crypt** peuvent être utilisés de la même manière que les mots de passe non codés pour accéder au serveur ObjectServer. Par conséquent, vous devez définir les autorisations appropriées sur les fichiers contenant des mots de passe codés pour éviter tout accès non autorisé. Sinon, les mots de passe qui ont été codés avec **nco_g_crypt** doivent être à nouveau codés avec **nco_aes_crypt** et les autorisations sur le fichier clé doivent être définies correctement.

Pour plus d'informations sur l'utilisation du chiffrement de la valeur de propriété, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*. Pour plus d'informations sur l'exécution des sondes et des passerelles en mode sécurisé, voir *Guide des sondes et des passerelles d'IBM Tivoli Netcool/OMNIbus*.

Référence associée:

«Exécution du serveur proxy en mode sécurisé», à la page 50
Vous pouvez exécuter le serveur proxy en mode sécurisé. Lorsque vous indiquez la propriété **SecureMode** ou l'option de ligne de commande `-secure`, le serveur proxy authentifie les connexions de sonde en demandant un nom d'utilisateur et un mot de passe.

«Propriétés et options de ligne de commande du serveur proxy», à la page 47
Le serveur proxy lit son fichier de propriétés lors du démarrage. Si une propriété n'est pas indiquée dans ce fichier, la valeur par défaut est utilisée, sauf si vous utilisez une option de ligne de commande pour la remplacer. L'emplacement par défaut du fichier de propriétés est `$NCHOME/omnibus/etc/`*`serveur proxy`*`.props`.

Mises à jour d'outil client via IDUC

Une quantité importante de données est transmise entre le serveur ObjectServer et un client bureau à chaque mise à jour d'une liste d'événements. Pour empêcher toute surcharge du serveur ObjectServer avec les requêtes des mises à jour de liste d'événements, le serveur ObjectServer envoie une invite au client bureau lorsqu'une mise à jour est nécessaire. Le bureau demande ensuite les données mises à jour depuis le serveur ObjectServer.

Cette invite est envoyée au bureau via un lien de communication utilisant un protocole de communication IDUC (Insert, Delete, Update ou Control). L'invite demande au bureau d'actualiser tous les écrans des listes d'événements. Le protocole IDUC met à jour les passerelles de la même manière.

Le client bureau se connecte au serveur ObjectServer via le port défini par le fichier d'interfaces pour établir le lien de communication IDUC. Le bureau reçoit le numéro de connecteur de la connexion IDUC sur laquelle il recevra les invites du serveur ObjectServer des mises à jour.

Spécification de l'intervalle de mise à jour IDUC

L'intervalle de mise à jour est contrôlé par la propriété **Granularity** du serveur ObjectServer ou par l'option de ligne de commande `-granularity`, définie par défaut à 60 secondes. La valeur par défaut est optimale pour la plupart des systèmes. Si vous réduisez l'intervalle, le temps de réponse des outils clients s'améliore, mais le trafic réseau et la charge du serveur ObjectServer augmentent également.

Si vous devez effectuer des actualisations IDUC plus fréquentes, configurez le taux d'actualisation côté client, par exemple dans le fichier de configuration des passerelles.

Spécification du port IDUC

Par défaut, au démarrage d'un serveur ObjectServer, un numéro de port disponible est choisi pour la connexion IDUC. Vous pouvez également spécifier le port IDUC à utiliser. Vous devez indiquer le port IDUC lors de l'accès à un serveur ObjectServer protégé par un pare-feu.

Pour définir ces ports, mettez à jour le fichier suivant pour votre système d'exploitation :

- UNIX Linux `/etc/services`
- Windows `%SystemRoot%\system32\drivers\etc\services`

Le fichier `services` possède une entrée pour chaque ObjectServer au format suivant :

`nco_`*`nom_serveur nnnn`*`/tcp`

Où *nom_serveur* représente le nom du serveur ObjectServer et *nnnn* le numéro de port.

Vous pouvez paramétrer le port sur tout numéro non utilisé en dehors de la plage 1 à 1024, qui est généralement réservée au système.

UNIX Linux Lorsque le fichier `/etc/services` est géré par Network Information Service (NIS), vous devez créer l'entrée dans le fichier de services NIS, puis copier la configuration mise à jour sur tous les ordinateurs hôte.

Vous pouvez également utiliser l'option `-listeningport` sur la ligne de commande du serveur ObjectServer pour spécifier le port IDUC.

Exemples d'entrées de serveurs ObjectServer intitulés NCOMS et DENCO :

```
nco_NCOMS 7070/tcp
nco_DENCO 7071/tcp
```

Configuration du serveur ObjectServer pour le support multiculturel

Tivoli Netcool/OMNIbus prend en charge divers codages de caractères à octet unique et multi-octets pour une utilisation dans différents paramètres nationaux.

Pourquoi et quand exécuter cette tâche

Les propriétés ObjectServer suivantes concernent le traitement des caractères codés sur plusieurs octets :

- **Store.LocalizedSort** : cette propriété permet d'activer le tri localisé. Cette fonction est désactivée par défaut pour obtenir des performances optimales.
- **Store.LocalizedSortCaseSensitive** : cette propriété permet de contrôler la sensibilité à la casse dans le tri localisé.
- **RegexpLibrary** : utilisez cette propriété pour activer l'utilisation de la bibliothèque d'expressions régulières étendue POSIX 1003.2 (TRE). La bibliothèque d'expressions régulières NETCOOL standard est activée par défaut pour obtenir des performances système optimales.

Vous pouvez définir ces propriétés en appliquant une des procédures ci-dessous :

- Définissez les propriétés dans le fichier de propriétés ObjectServer `$NCHOME/omnibus/etc/`*`nom_serveur`*`.props`, où *nom_serveur* est le nom spécifié pour le serveur ObjectServer lors de sa création.
- Modifiez les paramètres depuis la ligne de commande au démarrage du serveur ObjectServer à l'aide de la commande **nco_objserv**. L'option de ligne de commande pour la propriété **Store.LocalizedSort** est `-storelocalesort`. L'option de ligne de commande pour la propriété **Store.LocalizedSortCaseSensitive** est `-storecasesort`. L'option de ligne de commande pour la propriété **RegexpLibrary** est `-regexplib`.

Remarque : Si votre nom d'utilisateur et votre mot de passe sont contrôlés selon une source d'authentification externe, vous devez vous assurer que cette source prend également en charge des caractères multi-octets. Si les caractères multi-octets ne sont pas pris en charge, vous devez spécifier les noms d'utilisateur et les mots de passe à l'aide de caractères ASCII.

Les noms des objets ObjectServer suivants sont limités aux caractères ASCII et commencent par une lettre ou un trait de soulignement suivi de lettres, chiffres et traits de soulignement :

- Memstores
- Bases de données
- Tables
- Colonnes
- Filtres de restriction
- Droits
- Groupes de déclencheurs
- Déclencheurs
- Signaux
- Procédures
- Paramètres
- Variables
- Propriétés

- Objets de fichier

  Les noms d'objets de fichier sont limités aux caractères ASCII spécifiés, alors que les noms des chemins d'accès aux fichiers sont limités aux caractères pris en charge par le système d'exploitation. Le codage de caractères utilisé pour créer les fichiers est le codage utilisé par le serveur ObjectServer : il peut être différent de celui du client.

Pour plus d'informations sur le support multiculturel pour Tivoli Netcool/OMNIbus, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Référence associée:

«Propriétés et options de ligne de commande du serveur ObjectServer», à la page 4
Utilisez les propriétés ou les options de ligne de commande ObjectServer pour configurer les paramètres du serveur ObjectServer. Pour éviter les erreurs, ajoutez autant de propriétés que possible au fichier de propriétés plutôt que des options de ligne de commande. D'autres utilitaires sont fournis que vous pouvez utiliser pour chiffrer les valeurs de propriété.

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 5. 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 6. 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 6. 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
`if( elapsed >= 300 )`	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.
`'ObjectServer ' + getservername() + ' is currently in flood',`	Texte de l'événement qui est déclenché lorsque le serveur ObjectServer passe en mode contrôle des afflux d'événements. Ne modifiez pas l'événement ' + `getservername()` + '.
`'ObjectServer ' + getservername() + ' is ending flood control',`	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 -password 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`.

Référence associée:

«Table catalog.profiles», à la page 438
La table catalog.profiles contient des informations de temporisation pour les commandes SQL en cours d'exécution à partir des connexions client.

«Table catalog.trigger_stats», à la page 441
La table catalog.trigger_stats stocke des informations de temporisation sur les déclencheurs, y compris le nombre de fois où le déclencheur a été mis au premier plan et le nombre de fois où il a été mis en application. Ces statistiques sont recueillies sauf si le système d'automatisation est désactivé, c'est-à-dire, sauf si l'option de ligne de commande `-autoenabled` est définie sur `FALSE`.

Stockage des données et utilisation du point de contrôle

Les données du serveur ObjectServer sont stockées en mémoire pour garantir un accès rapide. Le serveur ObjectServer prend en charge la persistance des données à l'aide de points de contrôle et de journaux, pour copier les données en mémoire sur disque. Cela permet de récupérer les données après un arrêt planifié ou non planifié.

Stockage des données à l'aide de magasins de mémoire

Les *magasins de mémoire* sont des conteneurs gérés par le serveur ObjectServer et conservent les tables et les données du serveur ObjectServer en mémoire.

Le serveur ObjectServer utilise les magasins de mémoire décrits dans la table suivante.

Tableau 7. Magasins de mémoire

Nom du magasin de mémoire	Type de stockage	Description
MASTER_STORE	Persistant	Utilisé pour stocker les descriptions internes des données du serveur ObjectServer.
TABLE_STORE	Persistant	Utilisé pour stocker les informations du bureau, notamment la table alerts.status.
TEMP_STORE	Transitoire	Utilisé pour stocker les données dont le caractère non persistant n'est pas obligatoire. Les tables système contenant des informations de configuration sont stockées ici et recréées au démarrage.
VIRTUAL_STORE	Virtuel	Utilisé généralement pour cataloguer rapidement les données changeantes (par exemple, données relatives aux clients connectés).

Vous pouvez modifier la taille du magasin de mémoire table_store via l'utilitaire **nco_store_resize**.

Référence associée:

«Modification des limites logicielle et matérielle du magasin de mémoire table_store», à la page 34
Si la base de données d'alertes prend des proportions exceptionnellement importantes, vous pouvez utiliser une commande afin de changer les limites souples et strictes.

Introduction à l'utilisation des points de contrôle

Le serveur ObjectServer prend en charge la persistance des données à l'aide de points de contrôle et de journaux, pour copier les données en mémoire sur disque. Cela permet de récupérer les données après un arrêt planifié ou non planifié.

Les types de fichier suivants sont conservés sur le disque :

- Fichiers de point de contrôle, contenant les données des tables entières
- Journaux de rediffusion, contenant les modifications apportées aux tables depuis le dernier point de contrôle

Toutes les 60 secondes, un point de contrôle est créé et toutes les données persistantes sont copiées dans les fichiers de point de contrôle. Entre les points de contrôle, les nouvelles données et les données modifiées sont écrites dans les fichiers journaux de rediffusion.

Les points de contrôle sont également créés à chaque arrêt planifié du serveur ObjectServer.

Création d'un fichier de point de contrôle

Les fichiers de point de création sont générés pour chaque magasin de mémoire persistant. Seuls les magasins de mémoire persistants font l'objet d'un point de contrôle.

Les fichiers de point de contrôle sont nommés comme suit :

- `$NCHOME/omnibus/db/`*`nom_serveur`*`/`*`nom_magasin`*`_0.chk`
- `$NCHOME/omnibus/db/`*`nom_serveur`*`/`*`nom_magasin`*`_1.chk`

Les journaux de rediffusion sont générés pour chaque magasin de mémoire persistant. Les fichiers de rediffusion sont nommés comme suit :

- `$NCHOME/omnibus/db/`*`nom_serveur`*`/`*`nom_magasin`*`_0.log`
- `$NCHOME/omnibus/db/`*`nom_serveur`*`/`*`nom_magasin`*`_1.log`

Le processus de point de contrôle écrit en alternance sur les fichiers `_0.chk` et `_1.chk`. Si un fichier est endommagé pendant un arrêt imprévu, les données de l'autre fichier de point de contrôle et les journaux de rediffusion sont utilisés pour régénérer les tables de base de données en mémoire. Au démarrage de chaque point de contrôle, le processus de consignation bascule vers l'autre fichier journal. L'ancien fichier journal est supprimé avant le démarrage du point de contrôle suivant.

Lorsqu'un arrêt planifié de serveur survient (via l'exécution de la commande ALTER SYSTEM SHUTDOWN), un fichier `.tab` est créé pour chaque magasin de mémoire persistant. Ce fichier est nommé *`nom_magasin`*`.tab`.

Par exemple, le fichier `.tab` du magasin maître du serveur ObjectServer NCOMS est nommé :

```
$NCHOME/omnibus/db/NCOMS/MASTER_STORE.tab
```

Le format de ces fichiers est spécifique au matériel et au système d'exploitation sur lesquels ils ont été créés.

Récupération des données au démarrage du serveur ObjectServer

Lorsque le serveur ObjectServer redémarre après un arrêt planifié, la base de données est régénérée à l'aide des fichiers `.tab`.

Lorsque le serveur ObjectServer redémarre après un arrêt non planifié, la base de données est régénérée à l'aide des fichiers du point de contrôle (`.chk`) et du journal de rediffusion (`.log`).

Utilitaire de vérification de point de contrôle nco_check_store

L'utilitaire **`nco_check_store`** vérifie la validité des fichiers de point de contrôle existants. Il est conçu pour êtreutilisé par les automatisations et ne peut être utilisé que pour vérifier les magasins du serveur ObjectServer qui ne sont pas en cours d'utilisation.

L'utilitaire **`nco_check_store`** rapporte la validité des fichiers de point de contrôle avec les codes retour suivants :

- `0` : Succès. Les fichiers de point de contrôle sont valides.
- `1` : Echec. Les fichiers de point de contrôle ne sont pas valides et ne doivent pas être utilisés.

Vous pouvez également exécuter l'utilitaire **nco_check_store** dans la ligne de commande. Depuis la ligne de commande, vous devez définir le niveau de consignation du message sur `info` pour afficher la progression et les résultats du test.

Remarque : N'exécutez pas **nco_check_store** lors de l'exécution du serveur ObjectServer.

Les options de ligne de commande de **nco_check_store** sont décrites dans la table suivante.

Tableau 8. Options de ligne de commande d'utilitaire de vérification de point de contrôle

Option de ligne de commande	Description
`-help`	Affiche l'aide sur les options de ligne de commande et quitte.
`-memstoredatadirectory` *chaîne*	Indique le chemin d'accès à la base de données du serveur ObjectServer. Le chemin par défaut est `$NCHOME/omnibus/db`.
`-messagelevel` *chaîne*	Indique le niveau de consignation du message. Les valeurs possibles sont : `debug`, `info`, `warn`, `error` et `fatal`. Le niveau par défaut est `error`. Les messages qui sont consignés à chaque niveau sont les suivants : • `fatal` : messages `fatal` uniquement • `error` : messages `fatal` et `error` • `warn` : messages `fatal`, `error` et `warn` • `info` : messages `fatal`, `error`, `warn` et `info` • `debug` : messages `fatal`, `error`, `warn`, `info` et `debug` **Conseil :** La valeur de la *chaîne* peut être en majuscules et/ou en minuscules.
`-messagelog` *chaîne*	Indique le chemin d'accès à l'emplacement de consignation des messages. La valeur par défaut est `stderr`.
`-server` *chaîne*	Indique le nom de la base de données du serveur ObjectServer à vérifier. La valeur par défaut est `NCOMS`.
`-version`	Affiche les informations de version sur l'utilitaire de vérification du point de contrôle et quitte.

Exemple d'utilisation

Pour utiliser la commande **nco_check_store** depuis une automatisation pour vérifier la validité des fichiers de sauvegarde créés par le serveur ObjectServer, définissez le répertoire contenant la sauvegarde dans l'option de ligne de commande `-memstoredatadirectory`.

Modification des limites logicielle et matérielle du magasin de mémoire table_store

Si la base de données d'alertes prend des proportions exceptionnellement importantes, vous pouvez utiliser une commande afin de changer les limites souples et strictes.

Si les base de données d'alertes devient exceptionnellement importante, le message suivant s'affiche :

```
Limite souple de la région dépassée
```

Cela signifie que le magasin de mémoire table_store, où sont stockées les données de table, a atteint sa taille maximale. Vous pouvez utiliser la commande ALTER MEMSTORE pour modifier la limite logicielle. Par exemple, dans **nco_sql**, entrez ce qui suit :

```
ALTER MEMSTORE table_store SET SOFT LIMIT 500 M;
```

La limite logicielle ne peut pas être supérieure à la limite matérielle, que vous pouvez modifier à l'aide de l'utilitaire **nco_store_resize**. Cet utilitaire permet de modifier la limite matérielle du magasin de mémoire table_store du serveur ObjectServer spécifié.

Remarque : Avant d'exécuter l'utilitaire **nco_store_resize**, vous devez sauvegarder le serveur ObjectServer à l'aide de la commande ALTER SYSTEM BACKUP, puis interrompre le serveur ObjectServer.

Les options de ligne de commande de l'utilitaire **nco_store_resize** sont décrites dans la table suivante.

Tableau 9. Options de ligne de commande d'utilitaire de redimensionnement de magasin de mémoire

Option de ligne de commande	Description
`-help`	Affiche l'aide sur les options de ligne de commande et quitte.
`-messagelevel` *chaîne*	Indique le niveau de consignation du message. Les valeurs possibles sont : `debug`, `info`, `warn`, `error` et `fatal`. Le niveau par défaut est `error`. Les messages qui sont consignées à chaque niveau sont les suivants : • `fatal` : messages `fatal` uniquement • `error` : messages `fatal` et `error` • `warn` : messages `fatal`, `error` et `warn` • `info` : messages `fatal`, `error`, `warn` et `info` • `debug` : messages `fatal`, `error`, `warn`, `info` et `debug` **Conseil :** La valeur de la *chaîne* peut être en majuscules et/ou en minuscules.
`-messagelog` *chaîne*	Indique le chemin d'accès à l'emplacement de consignation des messages. La valeur par défaut est `stderr`.

Tableau 9. Options de ligne de commande d'utilitaire de redimensionnement de magasin de mémoire (suite)

Option de ligne de commande	Description
`-newhardlimit` *entier*	Indique la nouvelle limite matérielle du magasin de mémoire, en Mo. La valeur par défaut est `500` Mo. Vous ne pouvez pas changer la limite stricte pour une limite plus élevée que la taille du magasin de mémoire maximale. Si vous dépassez la taille du magasin, la limite stricte est tronquée de telle sorte qu'elle respecte la taille du magasin de mémoire maximale autorisée. La taille du magasin de mémoire maximale autorisée (en Mo) pour la plateforme associée prend les valeurs suivantes : • AIX : 2047 • HPUX : 1024 • Linux x86 : 2047 • Solaris : 2047 • Windows : 700 • zLinux : 2047
`-server` *chaîne*	Indique le nom du serveur ObjectServer pour laquelle la taille de magasin de mémoire sera augmentée. La valeur par défaut est `NCOMS`.
`-version`	Affiche les informations de version sur l'utilitaire de vérification du point de contrôle et quitte.

Référence associée:

«ALTER SYSTEM», à la page 356
La commande `ALTER SYSTEM` permet de modifier les paramètres par défaut et actuels du serveur ObjectServer en paramétrant les propriétés, d'arrêter le serveur ObjectServer, de supprimer des connexions utilisateur ou de sauvegarder le serveur ObjectServer.

Utilisation de nco_postmsg pour envoyer des alertes aux serveurs ObjectServer

Vous pouvez envoyer une alerte à un serveur ObjectServer à l'aide de l'utilitaire **nco_postmsg**. Vous pouvez utiliser cet utilitaire en remplacement les utilitaires IBM Tivoli **postemsg** et **postzmsg**.

L'utilitaire **nco_postmsg** accepte les paires valeur-nom pour les données d'alerte et crée une instruction SQL INSERT. Cette instruction permet d'insérer une nouvelle ligne de données dans une table de base de données spécifiée dans le serveur ObjectServer.

Vous pouvez exécuter **nco_postmsg** à partir de la ligne de commande ou développer des scripts ou des automatisations qui utilisent la commande **nco_postmsg** pour envoyer des alertes au serveur ObjectServer. La fréquence d'exécution peut varier. Si vous exécutez l'utilitaire à partir de l'interface de ligne de commande, il se peut que l'utilitaire s'exécute plusieurs fois par jour. Si l'utilitaire est exécuté à partir d'un script, il peut être exécuté quelques fois par seconde. Plusieurs instances de l'utilitaire **nco_postmsg** peuvent également être exécutées simultanément. Voici quelques exemples d'utilisation de l'utilitaire **nco_postmsg** :

- L'utilitaire **nco_postmsg** vous permet d'envoyer des alertes simples au serveur ObjectServer à des fins de diagnostic ou de test (par exemple, pendant la configuration du système ou lors de l'identification et de la résolution de problèmes liés à la distribution d'alertes.
- **nco_postmsg** vous permet d'envoyer des alertes qui indiquent le début et la fin d'une période de maintenance.
- Vous pouvez utiliser **nco_postmsg** lorsque des commentaires en retour sur une automatisation externe sont nécessaires dans le serveur ObjectServer. Par exemple, si une alerte est provoquant l'exécution d'une automatisation externe est reçue. L'automatisation externe déclenche un script qui exécute une action. La réussite ou le succès de cette action doit être enregistré(e) dans le serveur ObjectServer. Vous pouvez inclure une commande **nco_postmsg** dans le script pour insérer une nouvelle alerte contenant le statut de l'action, et configurer des automatisations sur le serveur ObjectServer pour traiter la nouvelle alerte.

L'utilitaire **nco_postmsg** est installé avec la fonction **Support de sonde** de Tivoli Netcool/OMNIbus, et peut par conséquent être déployée séparément d'autres fonctions Tivoli Netcool/OMNIbus sur un ou plusieurs hôtes. Un fichier de propriétés appelé `nco_postmsg.props` est disponible pour l'utilitaire.

L'utilitaire **nco_postmsg** permet d'établir une connexion sécurisée au serveur ObjectServer à l'aide du langage SSL en modes FIPS 140-2 et non-FIPS 140-2.

Pour envoyer une alerte à un serveur ObjectServer, entrez la commande suivante :

```
$NCHOME/omnibus/bin/nco_postmsg [ -option [ valeur ] ... ]
"nom_colonne=valeur_colonne" ...
```

où :

- *-option* est l'option de ligne de commande **nco_postmsg** et *valeur* est la valeur définie pour l'option. Il n'est pas nécessaire de spécifier une valeur pour chaque option.
- *nom_colonne* est un nom de colonne valide de la table de base de données dans laquelle vous voulez insérer l'alerte et *valeur_colonne* est la valeur de données correspondante que vous voulez insérer. Chaque paire nom_valeur pour *nom_colonne=valeur_colonne* doit être incluse entre guillemets, comme illustré dans la syntaxe de la commande. En outre, la variable *valeur_colonne* doit être incluse entre guillemets simples s'il s'agit d'une valeur de chaîne.

  Remarque : Si vous envoyez une alerte à la table alerts.status, une paire nom-valeur est obligatoire pour la zone Identifier. Pour obtenir des informations sur la définition d'une valeur pour la colonne Identifier, reportez-vous au document *IBM Tivoli Netcool/OMNIbus Integration Best Practices*, à l'adresse https://www-304.ibm.com/jct01003c/software/brandcatalog/portal/opal/details?catalog.label=1TW10NC10.
  Lorsque vous spécifiez la variable *valeur_colonne*, utilisez un type de données approprié pour la zone ObjectServer. Utilisez Netcool/OMNIbus Administrator pour vérifier les types de données affectés aux zones des tables de base de données ou utiliser la commande SQL du serveur ObjectServer, DESCRIBE.

Exécution de nco_postmsg au codage UTF-8

Vous pouvez exécuter l'utilitaire **nco_postmsg** au codage UTF-8 à l'aide de l'option de ligne de commande `-utf8enabled`. Toutefois, vous ne pouvez pas spécifier de valeurs nom-paire directement à partir de la ligne de commande et devez plutôt

ajouter les paires nom-valeur à un fichier texte. Dans le fichier texte, spécifiez chaque paire nom-valeur sur une ligne distincte et placez la valeur entre des guillemets simples s'il s'agit d'une valeur de chaîne. Le fichier texte doit être au codage UTF-8 et peut contenir uniquement les données d'une alerte. Le fichier de propriétés nco_postmsg.props doit également être au codage UTF-8.

La syntaxe d'exécution de **nco_postmsg** au codage UTF-8 est la suivante :

```
$NCHOME/omnibus/bin/nco_postmsg [ -option [ valeur ] ... ] -utf8enabled
TRUE < nom_fichier.txt
```

où :

- *-option* est l'une des options de ligne de commande **nco_postmsg** (autre que -utf8enabled), et *valeur* est la valeur de l'option.
- *nom_fichier*.txt est le fichier qui contient les paires nom-valeur.

Traitement des erreurs

Lors de l'exécution de l'utilitaire **nco_postmsg**, celui-ci lit son fichier de propriétés et valide les options de ligne de commande spécifiées. Il génère ensuite l'instruction INSERT à l'aide des paires nom-valeur qui étaient spécifiées. Les alertes sont envoyées au serveur ObjectServer à condition que les paires nom-valeur et les options de ligne de commande soient valides. Les erreurs sont catégorisées comme indiqué ci-dessous. Toutes les erreurs sont consignées dans un fichier journal.

Erreurs provoquant l'écriture de l'alerte en cours dans un fichier en mémoire cache.

Si une erreur de communication survient en raison d'une panne du serveur ObjectServer ou d'un temps de réponse trop important, l'utilitaire **nco_postmsg** enregistre l'alerte dans un fichier en mémoire cache nommé nco_postmsg.cache, puis quitte. Les alertes suivantes qui sont générées en cas d'incident de communication sont également inscrites dans le fichier en mémoire cache. Lorsque le fichier en mémoire cache atteint sa taille maximale, le fichier est renommé avec un suffixe _old et un nouveau fichier nco_postmsg.cache est créé. Si un fichier nommé nco_postmsg.cache_old existe, il est supprimé pendant le processus de changement de nom et son contenu est perdu.

Conseil : La limite de temps de réponse du serveur ObjectServer est définie par la propriété **Ipc.Timeout** de **nco_postmsg**, et la taille maximale du fichier en mémoire cache est définie par la propriété **MaxCacheFileSize**.

Lors de la prochaine exécution de l'utilitaire **nco_postmsg** et de la prochaine connexion au serveur ObjectServer, les alertes sont envoyées au serveur ObjectServer dans l'ordre suivant :

1. Les alertes du fichier nco_postmsg.cache_old (s'il en existe un). Les alertes sont envoyées dans l'ordre d'ancienneté, en commençant par les plus anciennes. Lorsque le fichier est vide, il est supprimé.
2. Les alertes du fichier nco_postmsg.cache. Lorsque le fichier est vide, le fichier est tronqué pour atteindre la taille de 0 octet.
3. L'alerte en cours, qui a été générée lors de l'exécution de l'utilitaire **nco_postmsg**.

Les erreurs qui provoquent la suppression de l'alerte en cours.

Les alertes sont supprimées dans les conditions suivantes :

- Erreurs de ligne de commande : un fichier de propriétés non valide a été spécifié ou une syntaxe non valide a été spécifiée pour les paires nom-valeur.
- Erreurs d'authentification : l'authentification au niveau du serveur ObjectServer a échoué en raison de données d'identification non valides, d'un nom du serveur ObjectServer non valide ou d'erreurs SSL.
- Erreurs de base de données : le droit d'accès en écriture dans la table de base de données n'est pas défini, le nom de table de base de données est non valide ou l'instruction INSERT générée contient une erreur de syntaxe.
- Erreurs de fichier : une erreur s'est produite pendant l'écriture de l'entrée dans le fichier cache ou le fichier de propriétés contient des erreurs de syntaxe.

Remarques supplémentaires

- Les utilisateurs de **nco_postmsg** sont sujets au système d'autorisation du serveur ObjectServer et nécessitent par conséquent une autorisation d'insertion dans la table de base de données dans laquelle les données sont écrites.
- Si le mot de passe n'est pas défini dans le fichier de propriétés, il est requis lors de l'exécution de l'utilitaire **nco_postmsg** à partir de la ligne de commande.
- Si l'utilitaire **nco_postmsg** insère des données dans la table alerts.status, il définit la valeur des colonnes FirstOccurrence et LastOccurrence sur l'heure UNIX ou POSIX actuelle, à moins que ces colonnes ne soient spécifiées de façon explicite dans la ligne de commande ou dans un script.
- Si une valeur est spécifiée pour la colonne Class lors de l'exécution de **nco_postmsg**, une paire nom-valeur est ajoutée à l'instruction INSERT. Si aucune valeur n'est spécifiée pour la colonne Class, le serveur ObjectServer affecte la classe par défaut `0` (zéro) à l'alerte.
- Plusieurs instances de l'utilitaire **nco_postmsg** peuvent être configurées pour mettre en mémoire cache des alertes dans le même fichier à l'aide de la propriété **CacheFile**.
- Si une erreur de communication se produit lors de l'envoi d'alertes d'un fichier cache à la table alerts.status, les alertes sont conservées dans le fichier cache. L'utilitaire **nco_postmsg** tente d'envoyer une nouvelle fois les alertes lors de la prochaine exécution de l'utilitaire. Par conséquent, il se peut que les alertes soient envoyées plusieurs fois, auquel cas l'incrémentation de la zone Tally sera peu précise.

Concepts associés:

Chapitre 4, «Utilisation de Netcool/OMNIbus Administrator pour configurer les serveurs ObjectServer», à la page 69
Le serveur ObjectServer stocke, gère et traite les données d'alerte et de statut collectées par des applications externes telles que des sondes et des passerelles. Vous pouvez utiliser Netcool/OMNIbus Administrator pour configurer vos objets du serveur ObjectServer et configurer le contrôle du processus.

Référence associée:

«DESCRIBE», à la page 382
La commande `DESCRIBE` permet d'afficher des informations sur les colonnes de la table ou de la vue spécifiée.

Annexe D, «Tables du serveur ObjectServer», à la page 415
La base de données du serveur ObjectServer contient les tables suivantes : les tables d'alertes, les tables de services, les tables de catalogues système, les tables de statistiques, les tables de prise en charge d'outils client, les tables d'outils de bureau, les tables du serveur ObjectServer de bureau, les tables de sécurité, les tables de canaux IDUC et les tables d'événements affectés par un service.

Propriétés et options de ligne de commande nco_postmsg

L'utilitaire **nco_postmsg** contient un ensemble de propriétés et d'options de ligne de commande pour envoyer des alertes au serveur ObjectServer. Vous pouvez exécuter cette commande à partir du répertoire `$NCHOME/omnibus/bin`.

Le fichier de propriétés **nco_postmsg** est `$NCHOME/omnibus/etc/nco_postmsg.props`. Dans un fichier de propriétés non modifié, toutes les propriétés possèdent les valeurs par défaut et sont indiquées par le symbole dièse(#) en début de ligne. Pour remplacer une valeur par défaut, modifiez son paramètre dans le fichier de propriétés et supprimez le signe dièse (#). Si vous modifiez un paramètre de la ligne de commande, la valeur par défaut et le paramètre du fichier de propriétés sont remplacés.

Les propriétés et options de ligne de commande de **nco_postmsg** sont décrites dans la table suivante.

Tableau 10. Propriétés et options de ligne de commande nco_postmsg

Propriété	Option de ligne de commande	Description
CacheFile *chaîne*	`-cachefile` *chaîne*	Indique le chemin d'accès complet et le nom du fichier de cache où les alertes sont stockées lorsqu'elles ne peuvent pas être envoyées au serveur ObjectServer en raison d'un échec de communication. La valeur par défaut est `$NCHOME/omnibus/var/nco_postmsg.cache`. Lorsque le fichier cache atteint sa taille maximale, tel que cela est spécifié par la propriété **MaxCacheFileSize**, le nom du fichier `nco_postmsg.cache` devient `nco_postmsg.cache_old` et un nouveau fichier `nco_postmsg.cache` est créé.
CacheFileWarnThreshold *entier*	Non applicable	Génère l'inscription d'un message d'avertissement dans le fichier journal quand la taille du fichier `nco_postmsg.cache` dépasse un pourcentage spécifié. Notez que ce paramètre n'a aucun effet sur le fichier `nco_postmsg.cache_old`. La valeur par défaut est `90` %. La plage disponible est comprise entre 10 % et 99 %.

Tableau 10. Propriétés et options de ligne de commande nco_postmsg (suite)

Propriété	Option de ligne de commande	Description
ConfigCryptoAlg *chaîne*	Non applicable	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 : • 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**. La valeur par défaut est AES.
ConfigKeyFile *chaîne*	Non applicable	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**.
Non applicable	-help	Affiche l'aide sur les options de ligne de commande et quitte.
Ipc.Timeout *entier*	-ipctimeout *entier*	Définit le temps d'attente en secondes de l'utilitaire **nco_postmsg** pour une réponse d'un serveur ObjectServer, lorsqu'il tente d'envoyer des alertes. La valeur par défaut est 60 secondes. Si ce temps est dépassé (ou que le serveur ObjectServer est inactif), une erreur de communication survient et l'alerte est envoyée au fichier de mémoire cache.
MaxCacheFileSize *entier*	-maxcachefilesize *entier*	Définit la taille maximale du fichier de mémoire cache en Ko. La valeur par défaut est 10240 ko. Si la valeur est 0 (zéro), aucun fichier de mémoire cache n'est créée, ni utilisée.

Tableau 10. Propriétés et options de ligne de commande nco_postmsg (suite)

Propriété	Option de ligne de commande	Description
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 : messages fatal uniquement • error : messages fatal et error • warn : messages fatal, error, et warn • info : messages fatal, error, warn, et info • debug : messages fatal, error, warn, info et debug **Conseil :** La valeur de *chaîne* peut être en majuscules et/ou en minuscules.
MessageLog *chaîne*	-messagelog *chaîne*	Indique l'emplacement de consignation des messages. Les messages peuvent être consignés dans un fichier journal ou dans stderr. La valeur par défaut est $NCHOME/omnibus/log/nco_postmsg.log.
Name *chaîne*	-name *chaîne*	Indique un nom de client pour l'utilitaire **nco_postmsg**. Ce nom est utilisé en tant que description d'application lorsque **nco_postmsg** se connecte au serveur ObjectServer. La valeur par défaut est nco_postmsg. La valeur de la propriété **Name** détermine également le nom du fichier de propriétés, qui est au format suivant : $NCHOME/omnibus/etc/*nom*.props. **Conseil :** La propriété **Name** vous permet de faire la distinction entre plusieurs clients **nco_postmsg** qui envoient des alertes au serveur ObjectServer, et de conserver des fichiers de propriétés individuels par client.
Password *chaîne*	-password *chaîne*	Indique le mot de passe de l'utilisateur qui se connecte au serveur ObjectServer. La valeur par défaut est ''.
Props.CheckNames TRUE | FALSE	Non applicable	Interrompt l'utilitaire **nco_postmsg** si une propriété indiquée est non valide. La valeur par défaut est TRUE.
PropsFile *chaîne*	-propsfile *chaîne*	Indique le chemin complet et le nom du fichier de propriétés de l'utilitaire **nco_postmsg**. Le nom par défaut est $NCHOME/omnibus/etc/nco_postmsg.props.
Server *chaîne*	-server *chaîne*	Définit le nom du serveur ObjectServer auquel l'utilitaire **nco_postmsg** se connecte et envoie une alerte. La valeur par défaut est NCOMS.
SSLServerCommonName *chaîne*	Non applicable	Indique une liste de noms usuels séparée par une virgule à utiliser si l'utilitaire **nco_postmsg** se connecte au serveur ObjectServer à l'aide de SSL. Par exemple, string1,string2,string3. La valeur par défaut est ''.
Table *chaîne*	-table *chaîne*	Définit le nom de la table de base de données dans laquelle l'instruction INSERT ajoute les données de l'alerte. La valeur par défaut est alerts.status.

Tableau 10. Propriétés et options de ligne de commande nco_postmsg (suite)

Propriété	Option de ligne de commande	Description
UserName *chaîne*	-username *chaîne* ou -user *chaîne*	Indique le nom d'utilisateur employé dans le cadre de l'authentification lors de la connexion au serveur ObjectServer. La valeur par défaut est le nom d'utilisateur employé pour se connecter à l'ordinateur.
N/D	Windows -utf8enabled TRUE | FALSE	Contrôle le codage des données communiquées ou gérées par cette application sous Windows. Définissez la valeur -utf8enabled sur TRUE pour exécuter l'application au codage UTF-8. La valeur par défaut est FALSE ; ce qui implique l'utilisation de la page de codes par défaut du système. **Important :** Bien qu'une propriété **UTF8Enabled** soit disponible, la définition de cette propriété sur TRUE en vue d'activer le codage UTF-8 n'a aucun effet. Pour une exécution sous Windows au codage UTF-8, vous devez toujours utiliser l'option de ligne de commande -utf8enabled.
Non applicable	-version	Affiche les informations de version de l'utilitaire et quitte.

Exemples nco_postmsg et instructions INSERT obtenues

L'utilitaire **nco_postmsg** accepte l'entrée depuis les options de ligne de commande, les scripts ou un fichier texte et génère une instruction SQL INSERT du serveur ObjectServer, qui est envoyée à un serveur ObjectServer. Les exemples suivants présentent des exemples de commandes **nco_postmsg** et leurs instructions INSERT correspondantes.

Exemple 1

Cet exemple écrit une alerte dans le serveur ObjectServer NCOMS par défaut et la table alerts.status par défaut. Le nom d'utilisateur et le mot de passe permettant de s'authentifier auprès de NCOMS n'ont pas été définis dans le fichier de propriétés **nco_postmsg**. Au lieu de cela, ils sont spécifiés à partir de la ligne de commande.

La commande **nco_postmsg** entrée est la suivante :

```
nco_postmsg -user root -password "" "Identifier='example 1'" "Severity=3"
 "Manager='nco_postmsg'" "Summary='An event occurred'"
```

L'instruction INSERT correspondante envoyée à la table alerts.status dans NCOMS est la suivante :

```
insert into alerts.status (Identifier,Severity,Manager,Summary,FirstOccurrence,LastOccurrence)
 values ('example 1',3,'nco_postmsg','an event occurred',1255341764,1255341764);
```

Etant donné que les paires nom-valeur n'ont pas été expressément indiquées pour FirstOccurrence et LastOccurrence dans la commande **nco_postmsg**, FirstOccurrence et LastOccurrence sont automatiquement définies sur l'heure UNIX actuelle dans l'instruction INSERT envoyée à la table alerts.status.

Exemple 2

Cet exemple écrit une alerte dans la table alerts.status d'un serveur ObjectServer nommé PRESLEY. L'utilitaire **nco_postmsg** s'authentifie auprès du serveur

ObjectServer à l'aide du nom d'utilisateur myname et d'un mot de passe codé, qui sont spécifiés dans le fichier de propriétés **nco_postmsg**.

Dans ce cas, le chiffrement des valeurs de propriété est supposé être utilisé pour coder le mot de passe dans le fichier de propriétés afin que le mot de passe ne puisse pas être lu sans une clé. Les paramètres du fichier de propriétés des données d'identification de connexion, notamment le chiffrement du mot de passe, sont initialement vides, puis renseignés comme suit :

```
ConfigCryptoAlg: 'AES_FIPS'
ConfigKeyFile: '/dir/subdir/secret.txt'

...

Password: '@44:Kris2m3QLsy+dZYNt3/jptl8cd7c6Fmboaj+E6XrNw8=@'
UserName: 'myname'
```

La commande **nco_postmsg** entrée est la suivante :

```
nco_postmsg –server PRESLEY "Identifier='example 2'" "Node='London'"
```

L'instruction INSERT correspondante envoyée à la table alerts.status dans PRESLEY est la suivante :

```
insert into alerts.status (Identifier,Node,FirstOccurrence,LastOccurrence)
 values ('example 2','London',1255341764,1255341764);
```

Exemple 3

Cet exemple insère une alerte dans la table mydb.mytable du serveur ObjectServer NCOMS par défaut. Dans ce cas, un nom d'utilisateur et un mot de passe sont supposés être spécifiés dans le fichier de propriétés **nco_postmsg**.

La commande **nco_postmsg** entrée est la suivante :

```
nco_postmsg –table mydb.mytable "Identifier='example 3'" "Occurrence=1234567890"
 "Summary='write into my table'"
```

L'instruction INSERT correspondante envoyée à la table mydb.mytable dans NCOMS est la suivante :

```
insert into mydb.mytable (Identifier,Occurrence,Summary)
 values ('example 3',1234567890,'write into my table');
```

Exemple 4

Cet exemple explique comment exécuter l'utilitaire **nco_postmsg** au codage UTF-8.

L'utilitaire **nco_postmsg** s'authentifie auprès du serveur ObjectServer à l'aide du nom d'utilisateur myname et d'un mot de passe, qui sont indiqués dans le fichier de propriétés nco_postmsg.props de la manière suivante :

```
Password: 'secret'
UserName: 'myname'
```

Ce fichier de propriétés doit être enregistré au codage UTF-8.

Un fichier texte, my_data.txt, est créé avec les paires nom-valeur suivantes pour l'alerte à envoyer au serveur ObjectServer :

```
Identifier='example 2'
Node='London'
```

Ce fichier texte doit être enregistré au codage UTF-8.

Pour exécuter **nco_postmsg** au codage UTF-8 et écrire une alerte dans la table alerts.status d'un serveur ObjectServer nommé PRESLEY (également présumé au codage UTF-8), la commande saisie est la suivante :

```
nco_postmsg –server PRESLEY -utf8enabled TRUE < /chemin_fichier/my_data.txt
```

L'instruction INSERT correspondante envoyée à la table alerts.status dans PRESLEY est la suivante :

```
insert into alerts.status (Identifier,Node,FirstOccurrence,LastOccurrence)
 values ('example 2','London',1255341764,1255341764);
```

Chapitre 2. Configuration d'un serveur proxy

Le serveur ObjectServer reçoit des informations d'alerte des sondes. Dans une configuration standard, les alertes sont réacheminées directement vers le serveur ObjectServer. Vous pouvez configurer un serveur proxy pour réduire le nombre de connexions de sonde à un serveur ObjectServer.

Lorsqu'un grand nombre de sondes réacheminent des informations d'alerte directement vers le serveur ObjectServer et qu'un grand nombre de connexions de bureau sont également établies avec cet ObjectServer, l'impact sur les performances peut être négatif.

Un serveur proxy propose une mémoire tampon pour réduire le nombre de connexions directes au serveur ObjectServer principal. Plusieurs connexions de sonde établies avec le serveur proxy sont multiplexées et réacheminées via une seule connexion vers le serveur ObjectServer.

La figure suivante illustre comment les sondes communiquent avec le serveur proxy.

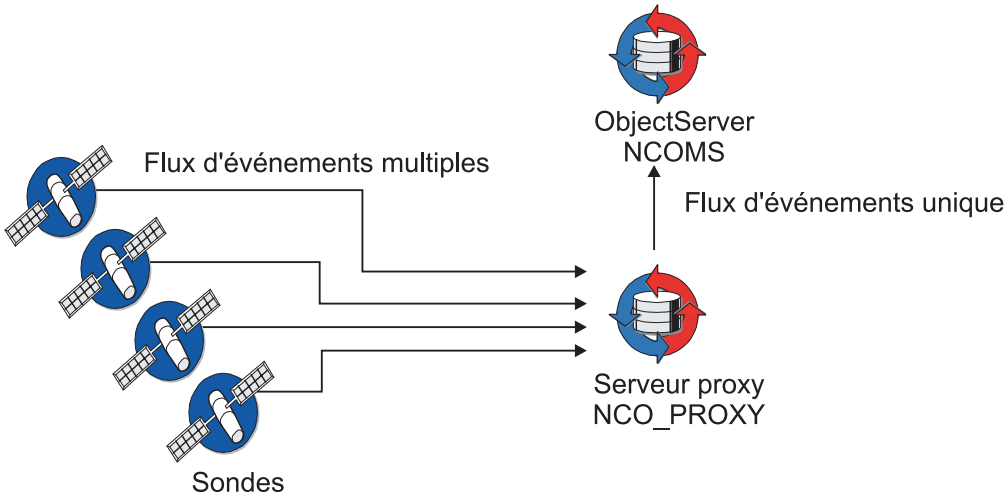

Figure 1. Exemple d'architecture de serveur proxy

Démarrage du serveur proxy

Vous pouvez démarrer le serveur proxy automatiquement en utilisant le contrôle de processus sous UNIX et Windows ainsi que les services sous Windows. Vous pouvez également le démarrer manuellement à partir de la ligne de commande. En règle générale, utilisez le contrôle de processus pour démarrer le serveur proxy.

Pourquoi et quand exécuter cette tâche

© Copyright IBM Corp. 1994, 2014

Démarrage d'un serveur proxy en utilisant le contrôle de processus

Sous UNIX et Windows, vous pouvez démarrer un serveur proxy en tant que processus à l'aide du contrôle des processus. Le serveur proxy doit être défini en tant que processus ou partie d'un service.

Pourquoi et quand exécuter cette tâche

Concepts associés:

Chapitre 7, «Utilisation du contrôle de processus pour gérer des processus et des procédures externes», à la page 247
Le système de contrôle Tivoli Netcool/OMNIbus effectue deux tâches principales. Il gère les processus locaux et distants, puis exécute les processus externes spécifiés dans les automatisations.

Démarrage d'un serveur proxy en utilisant les services (Windows)

Sous Windows, vous pouvez facultativement installer et exécuter le serveur proxy en tant que service Windows. Lorsque le service est défini sur automatique, le serveur proxy est lancé au démarrage de l'ordinateur.

Pourquoi et quand exécuter cette tâche

Vous devez installer et configurer manuellement le serveur proxy pour l'exécuter en tant que service sur un hôte Windows.

Démarrage manuel du serveur proxy

La commande **nco_proxyserv** vous permet de démarrer manuellement le serveur proxy.

Pourquoi et quand exécuter cette tâche

A partir de la ligne de commande, entrez la commande appropriée à votre système d'exploitation :

Tableau 11. Démarrage du serveur proxy à partir de la ligne de commande

Système d'exploitation	Commande
UNIX	`$NCHOME/omnibus/bin/nco_proxyserv [ -name nom_proxy] [-server nom_serveur]`
Windows	`%NCHOME%\omnibus\bin\nco_proxyserv [ -name nom_proxy] [-server nom_serveur]`

Dans les deux commandes, *nom_proxy* est le nom du serveur proxy et *nom_serveur* est le nom du serveur ObjectServer. Si vous ne spécifiez pas l'option de ligne de commande `-name`, le nom de serveur proxy par défaut est `NCO_PROXY`. Si vous ne spécifiez pas l'option de ligne de commande *-server*, le serveur proxy met en mémoire tampon les connexions pour le serveur ObjectServer NCOMS.

Vous pouvez démarrer le serveur proxy avec des options de ligne de commande supplémentaires.

Propriétés et options de ligne de commande du serveur proxy

Le serveur proxy lit son fichier de propriétés lors du démarrage. Si une propriété n'est pas indiquée dans ce fichier, la valeur par défaut est utilisée, sauf si vous utilisez une option de ligne de commande pour la remplacer. L'emplacement par défaut du fichier de propriétés est `$NCHOME/omnibus/etc/`*`serveur proxy`*`.props`.

Dans le fichier de propriétés, une propriété et sa valeur correspondante sont séparées par deux points (:). Les valeurs de chaîne sont entourées par des guillemets, par exemple :

```
ServerName: "NCO_PROXY"
```

Conseil : Vous pouvez chiffrer les valeurs de chaîne dans un fichier de propriétés en utilisant le chiffrement de valeur de propriété.

Les options de ligne de commande du serveur proxy utilisent le format suivant :

```
nco_proxyserv [ -option [ valeur ] ... ]
```

Dans cette commande, *-option* est l'option de ligne de commande et *valeur* est la valeur sur laquelle vous définissez l'option. Vous ne devez pas indiquer de valeur pour toutes les options.

Si vous n'indiquez pas de fichier de propriétés lors du démarrage d'un serveur proxy, le fichier par défaut est utilisé. L'option de ligne de commande `-propsfile` vous permet d'indiquer le chemin d'accès et le nom de fichier complets d'un fichier de propriétés alternatif.

Le tableau suivant répertorie les propriétés et les options de ligne de commande du serveur proxy.

Tableau 12. Propriétés et options de ligne de commande du serveur proxy

Propriété	Option de ligne de commande	Description
AuthPassword *chaîne*	N/D	Mot de passe associé au nom d'utilisateur utilisé pour authentifier le serveur proxy lors de la connexion au serveur ObjectServer s'exécutant 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.

Tableau 12. Propriétés et options de ligne de commande du serveur proxy (suite)

Propriété	Option de ligne de commande	Description
AuthUserName *chaîne*	N/D	Nom d'utilisateur utilisé pour authentifier le serveur proxy lors de la connexion à un serveur ObjectServer s'exécutant en mode sécurisé. La valeur par défaut est `root`.
ConfigCryptoAlg *chaîne*	N/D	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 : • 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**.
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**.
ConnectionRatio *entier*	`-ratio` *entier*	Définit le rapport entre les connexions de sondes entrantes et les connexions sortantes vers un serveur ObjectServer. La valeur par défaut de `10` crée un rapport de 10:1 pour les connexions entrantes et sortantes.
N/D	`-help`	Affiche les options de ligne de commande prises en charge et quitte.
N/D	`-logfile` *chaîne*	Définit le nom du fichier dans lequel le serveur proxy écrit des messages, y compris des erreurs. Par défaut, le fichier est `$NCHOME/omnibus/log/`*`nom de serveur`*`.log` , où la variable *nom de serveur* est définie par l'option `-name`.

Tableau 12. Propriétés et options de ligne de commande du serveur proxy (suite)

Propriété	Option de ligne de commande	Description
Connections *entier*	-connections *entier*	Définit le nombre maximum de connexions disponibles pour les sondes. La valeur par défaut est 256 et la valeur maximale est 1024. **Remarque :** La propriété **Connections** et l'option de ligne de commande -connections sont des alias de la propriété **MaxConnections** et de l'option de ligne de commande -max.
NetworkTimeout *entier*	-networktimeout *entier*	Indique une durée en secondes après laquelle une tentative de connexion ou une connexion au serveur ObjectServer dépasse le délai d'attente si un échec réseau se produit. Après le délai d'attente indiqué, le serveur proxy tente de se reconnecter au serveur ObjectServer. Si la connexion n'aboutit pas après un second délai d'attente, le serveur proxy tente de se connecter à un serveur ObjectServer de sauvegarde, le cas échéant. La valeur par défaut est de 20 secondes.
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.
N/D	-propsfile *chaîne*	Définit le nom du fichier de propriétés du serveur proxy. Le nom par défaut est $NCHOME/omnibus/etc/*nom de serveur*.props, où la variable *nom de serveur* est définie par l'option -name.
RemoteServer *chaîne*	-server *chaîne*	Définit le nom du serveur ObjectServer auquel le serveur proxy se connecte. La valeur par défaut est NCOMS.
SecureMode TRUE | FALSE	-secure	Définit le mode de sécurité du serveur proxy. Si cette option est activée, le serveur proxy authentifie les demandes de connexion de sonde à l'aide d'un nom d'utilisateur et d'un mot de passe. Si elle est désactivée (par défaut), les sondes peuvent se connecter au serveur proxy sans nom d'utilisateur ni de mot de passe.
ServerName *chaîne*	-name *chaîne*	Définit le nom du serveur proxy. Il s'agit du nom qui est configuré dans l'éditeur de serveur. La valeur par défaut est NCO_PROXY.
N/D	-version	Affiche les informations de version sur le serveur proxy et quitte.

Référence associée:

«Exécution du serveur proxy en mode sécurisé»
Vous pouvez exécuter le serveur proxy en mode sécurisé. Lorsque vous indiquez la propriété **SecureMode** ou l'option de ligne de commande `-secure`, le serveur proxy authentifie les connexions de sonde en demandant un nom d'utilisateur et un mot de passe.

«Exécution du serveur ObjectServer en mode sécurisé», à la page 23
Vous pouvez exécuter le serveur ObjectServer en mode sécurisé. Lorsque vous spécifiez l'option de ligne de commande `-secure`, le serveur ObjectServer authentifie les connexions de la sonde, de la passerelle et du serveur proxy via un nom d'utilisateur et un mot de passe.

Connexion au serveur proxy

Pour connecter les sondes au serveur proxy, indiquez le nom du serveur proxy dans la propriété **Server** du fichier de propriétés de la sonde ou utilisez l'option de ligne de commande `-server`.

Toutes les alertes sont ensuite envoyées au serveur proxy.

Exécution du serveur proxy en mode sécurisé

Vous pouvez exécuter le serveur proxy en mode sécurisé. Lorsque vous indiquez la propriété **SecureMode** ou l'option de ligne de commande `-secure`, le serveur proxy authentifie les connexions de sonde en demandant un nom d'utilisateur et un mot de passe.

Lorsqu'une demande de connexion est envoyée, le serveur proxy émet un message d'authentification. La sonde doit répondre par le nom d'utilisateur et le mot de passe corrects.

Si vous n'indiquez pas l'option `-secure`, les demandes de connexion de sonde ne sont pas authentifiées.

Lors de la connexion à un serveur proxy sécurisé, les propriétés **AuthUserName** et **AuthPassword** doivent être définies pour chaque sonde dans son fichier de propriétés. Si la combinaison nom d'utilisateur/mot de passe est incorrecte, le serveur proxy émet un message d'erreur.

Vous pouvez choisir n'importe quel nom d'utilisateur valide pour la propriété **AuthUserName**.

Le tableau suivant contient des détails sur le chiffrement de mots de passes pour l'exécution en mode FIPS 140–2 et non-FIPS 140–2.

Tableau 13. Chiffrement de mots de passe en mode FIPS 140–2 et non-FIPS 140–2

Mode	Action
Mode FIPS 140–2	Lorsque le mode FIPS 140–2 est activé, les mots de passe peuvent être spécifiés en texte normal ou dans un format chiffré. Vous pouvez coder les mots de passe à l'aide du chiffrement des valeurs de propriété de la façon suivante : 1. Si vous ne disposez pas encore d'une clé pour le chiffrement du mot de passe, créez-en une en exécutant l'utilitaire **nco_keygen** qui se trouve dans le répertoire `$NCHOME/omnibus/bin`. 2. Exécutez l'utilitaire **nco_aes_crypt** pour coder le mot de passe avec la clé générée par l'utilitaire **nco_keygen**. L'utilitaire **nco_aes_crypt** se trouve également dans le répertoire `$NCHOME/omnibus/bin`. Notez que vous devez spécifier `AES_FIPS` comme algorithme à utiliser pour le chiffrement du mot de passe. 3. Ouvrez le fichier de propriétés dans lequel vous souhaitez ajouter le mot de passe chiffré et spécifiez cette sortie chiffrée pour le paramètre **AuthPassword**. **Remarque :** Vous devez également définir la propriété **ConfigKeyFile** dans le fichier de clés que vous avez spécifié lors de l'exécution de **nco_aes_crypt**, et définir la propriété **ConfigCryptoAlg** dans l'algorithme de chiffrement utilisé.
Mode non-FIPS 140–2	Lorsque le mode FIPS 140–2 est désactivé, les mots de passe peuvent être spécifiés en texte normal ou dans un format chiffré. Cependant, le client transmet toujours des informations de connexion chiffrées indépendamment du chiffrement du mot de passe utilisé dans le fichier de propriétés. Vous pouvez coder les mots de passe à l'aide de l'utilitaire **nco_g_crypt** ou du chiffrement de la valeur de la propriété de la façon suivante : • Pour coder un mot de passe à l'aide de l'utilitaire **nco_g_crypt**, exécutez la commande suivante : `$NCHOME/omnibus/bin/nco_g_crypt` *`mdp_texte_clair`* Dans cette commande, *mdp_texte_clair* représente la forme non chiffrée du mot de passe. L'utilitaire **nco_g_crypt** prend le mot de passe non chiffré et l'affiche dans une version chiffrée. Ouvrez le fichier de propriétés dans lequel vous souhaitez ajouter le mot de passe chiffré et spécifiez cette sortie chiffrée pour le paramètre **AuthPassword**. • Pour coder un mot de passe à l'aide du chiffrement de la valeur de la propriété, vous avez besoin d'une clé générée à l'aide de l'utilitaire **nco_keygen**. Vous pouvez alors exécuter **nco_aes_crypt** pour coder le mot de passe avec la clé. Notez que vous pouvez spécifier `AES_FIPS` ou `AES` comme algorithme de chiffrement du mot de passe. 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. Ouvrez le fichier dans lequel vous souhaitez ajouter le mot de passe chiffré et spécifiez cette sortie chiffrée pour le paramètre **AuthPassword**. **Remarque :** Vous devez également définir la propriété **ConfigKeyFile** dans le fichier de clés que vous avez spécifié lors de l'exécution de **nco_aes_crypt**, et définir la propriété **ConfigCryptoAlg** dans l'algorithme de chiffrement utilisé.

Si le serveur ObjectServer s'exécute en mode sécurisé, le serveur proxy doit également posséder les propriétés **AuthUserName** et **AuthPassword** dans son fichier de propriétés pour se connecter au serveur ObjectServer. Si la combinaison nom d'utilisateur/mot de passe est incorrecte, le serveur ObjectServer émet un message d'erreur. La valeur **AuthPassword** peut être en texte en clair ou chiffrée, comme décrit dans le tableau précédent.

Avertissement : Les mots de passe chiffrés avec **nco_g_crypt** peuvent être utilisés de la même manière que les mots de passe chiffrés pour accéder au serveur ObjectServer. Par conséquent, vous devez définir les droits adéquats pour les fichiers contenant des mots de passe chiffrés pour éviter tout accès non autorisé. Sinon, les mots de passe qui ont été chiffrés avec **nco_g_crypt** doivent continuer d'être chiffrés avec **nco_aes_crypt** et les droits pour le fichier de clés doivent être définis de manière appropriée.

Pour de plus amples informations sur les propriétés de sonde, voir le manuel *Guide des sondes et des passerelles d'IBM Tivoli Netcool/OMNIbus*.

Chapitre 3. Configuration d'un serveur pare-feu

Dans un environnement sécurisé où le serveur ObjectServer et les sondes sont séparées par un pare-feu, configurez un serveur pare-feu pour que les sondes puissent se connecter au serveur ObjectServer depuis l'extérieur du réseau sécurisé.

Dans une configuration standard, les alertes sont réacheminées directement des sondes vers le serveur ObjectServer. Si les sondes sont situées à l'extérieur du pare-feu, celui-ci refuse les connexions au serveur ObjectServer. En configurant un serveur pare-feu, vous pouvez contourner cette restriction de sécurité.

Le pont pare-feu se compose de deux serveurs : un Server Access Bridge et un Client Access Bridge, qui s'exécutent de chaque côté du pare-feu. Une voie de communication entre les deux serveurs est initiée par le Server Access Bridge.

Le pont pare-feu utilise cette voie de communication pour créer de nouveaux canaux de données entre le Server Access Bridge et le Client Access Bridge. Les connexions et les données client peuvent être envoyées au serveur ObjectServer depuis l'extérieur du pare-feu. Les sondes lancent tout de même une connexion, mais celle-ci est à présent reliée au Client Access Bridge local, situé du même côté du pare-feu. Cela permet au Client Access Bridge, situé à l'extérieur du pare-feu, de fournir des flots de données au serveur ObjectServer, situé à l'intérieur du pare-feu.

La figure ci-après indique comment un flot de données circule à travers un pare-feu entre le serveur ObjectServer et deux sondes situées à l'extérieur du pare-feu.

© Copyright IBM Corp. 1994, 2014

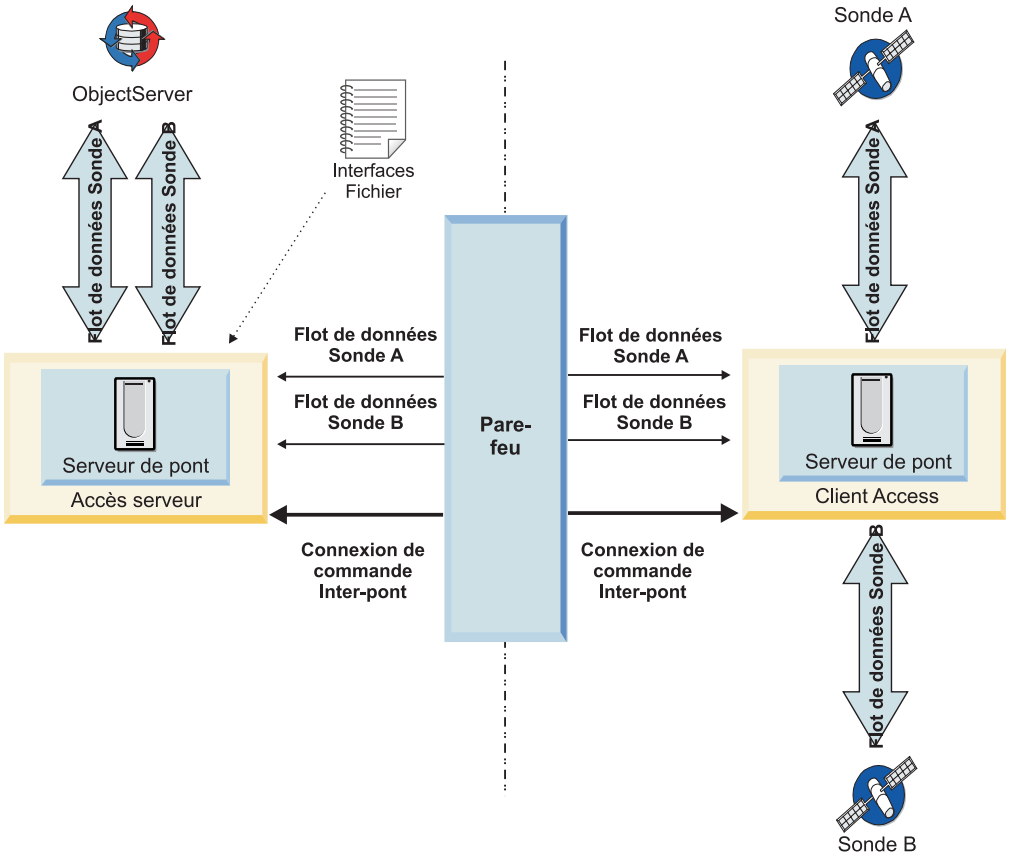

Figure 2. Exemple d'architecture de serveur pare-feu

Configuration d'un serveur pare-feu standard

Un pont pare-feu peut être configuré pour permettre à une sonde de se connecter à un serveur ObjectServer à partir d'un réseau non sécurisé.

La figure ci-après décrit la configuration d'un pont pare-feu standard :

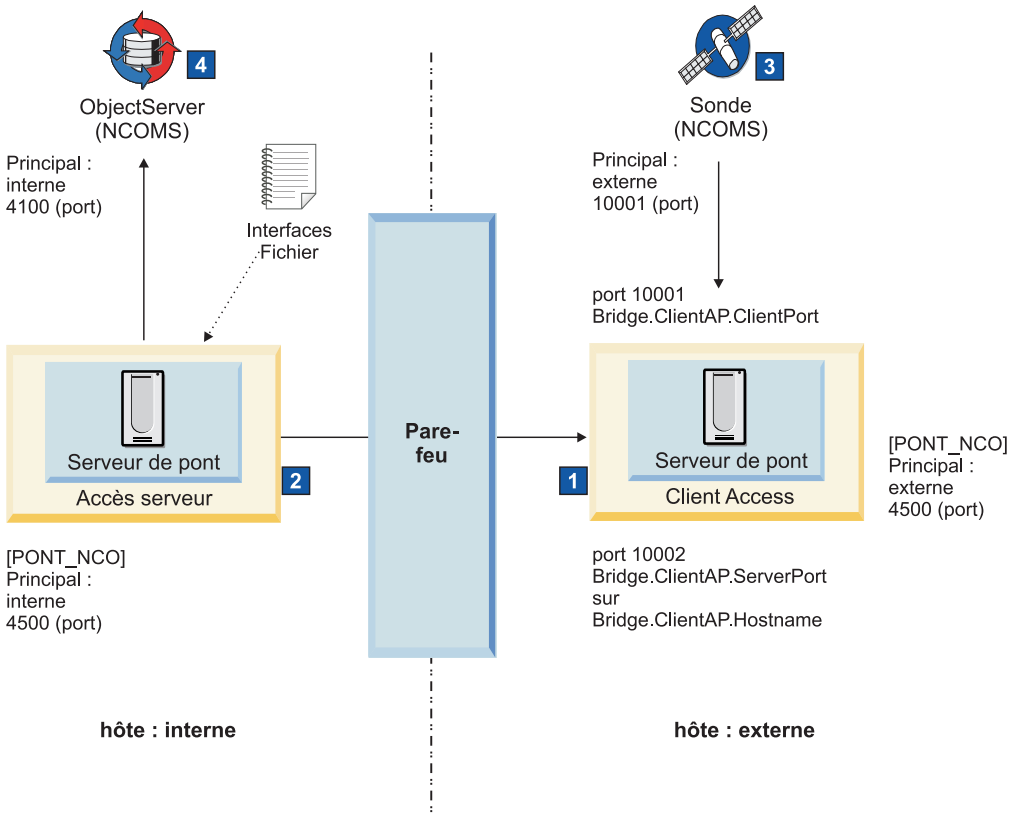

Figure 3. Configuration d'un serveur pare-feu standard

Le flux de configuration est le suivant :

1 Le Client Access Bridge

Le Client Access Bridge est en mode écoute des connexions depuis le Server Access Bridge sur le port et le nom d'hôte définis dans les propriétés **Bridge.ClientAP.ServerPort** et **Bridge.ClientAP.Hostname** du fichier de propriétés du Client Access Bridge qui lui sont associées. Le Client Access Bridge est également en mode écoute des connexions depuis les sondes sur le port indiqué dans la propriété **BridgeClientAP.ClientPort**.

2 Le Server Access Bridge

Le Server Access Bridge établit une connexion au Client Access Bridge (via le pare-feu) sur le port et le nom d'hôte définis dans les propriétés **Bridge.ClientAP.ServerPort** et **Bridge.ClientAP.Hostname** du fichier de propriétés du Server Access Bridge.

3 La sonde

La sonde établit une connexion au port Client Access du Client Access Bridge en se connectant au port et au nom d'hôte associés au nom du serveur défini dans le fichier d'interfaces `omni.dat`.

1 Le Client Access Bridge

Le Client Access Bridge demande alors un nouveau flot de données au Server Access Bridge.

2 Le Server Access Bridge

Le Server Access Bridge établit une nouvelle connexion au serveur ObjectServer (NCOMS) à l'aide du port et du nom d'hôte associés à NCOMS dans le fichier d'interfaces `omni.dat`.

Le Server Access Bridge établit alors une nouvelle connexion de flot de données au Client Access Bridge.

4 Le serveur ObjectServer

Les paquets de données envoyés à partir de la sonde (3) sont reçus par le Client Access Bridge (1), puis acheminés via les connexions de flot de données associées vers le Server Access Bridge (2). Celui-ci réachemine alors les données via la connexion réseau associée vers le serveur ObjectServer (4) et les données sont renvoyées via la même connexion.

Configuration d'un serveur pare-feu multiple

Les serveurs pare-feu multiples peuvent être configurés afin de permettre aux sondes de se connecter à un serveur ObjectServer à partir de plusieurs pare-feu.

La figure suivante décrit la configuration pour serveurs pare-feu multiples :

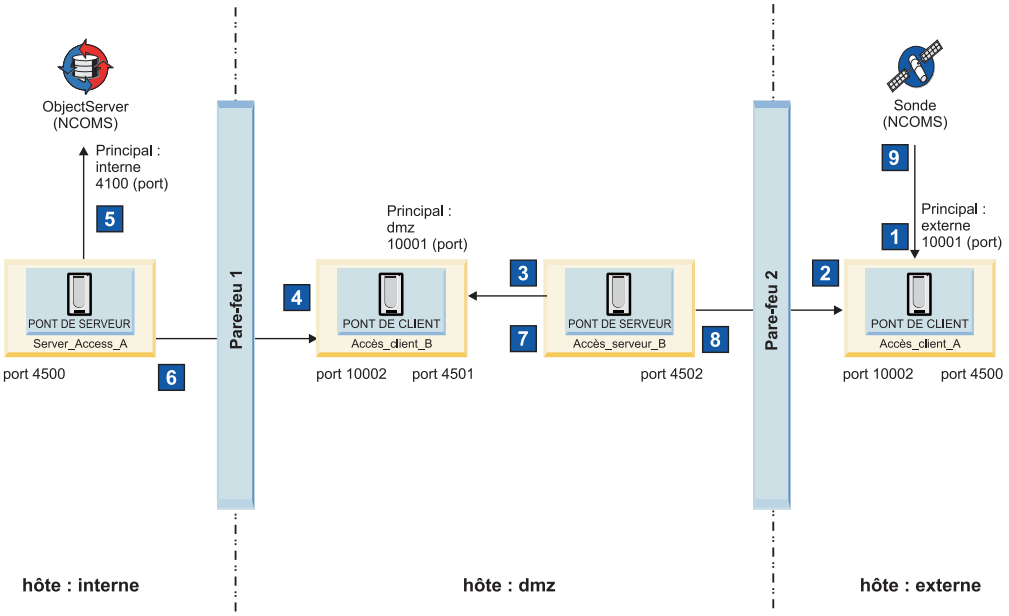

Figure 4. Configuration d'un serveur pare-feu multiple

Le flux de configuration est le suivant :

1

La sonde effectue une connexion initiale au serveur Client Access Bridge (CLIENT_ACCESS_A) sur l'hôte externe. Elle utilise le port (10001) et le nom d'hôte (externe) associés au nom du serveur NCOMS défini dans le fichier d'interfaces `omni.dat`.

2

Le serveur Client Access Bridge (CLIENT_ACCESS_A) demande alors une nouvelle connexion de flot de données (via Firewall 2) à partir du serveur Server Access Bridge (SERVER_ACCESS_B) qui lui est associé en utilisant la voie de communication existante entre les ponts.

3

Le serveur Server Access Bridge (SERVER_ACCESS_B) se connecte à nouveau au serveur Client Access Bridge (CLIENT_ACCESS_B). Elle utilise le port (10001) et le nom d'hôte (dmz) associés au nom du serveur NCOMS défini dans le fichier d'interfaces `omni.dat`.

4

Le serveur Client Access Bridge (CLIENT_ACCESS_B) demande alors une nouvelle connexion de flot de données (via Firewall 1) à partir du serveur Server Access Bridge (SERVER_ACCESS_A) qui lui est associé en utilisant la voie de communication existante entre les ponts.

5

Le serveur Server Access Bridge (SERVER_ACCESS_A) se connecte à nouveau au serveur ObjectServer (NCOMS) sur l'hôte interne. Il utilise le port et le nom d'hôte associés à NCOMS dans le fichier d'interfaces `omni.dat`.

Le serveur ObjectServer (NCOMS) accuse réception de la nouvelle connexion.

6 et 7

Le serveur Server Access Bridge (SERVER_ACCESS_A) lance une nouvelle connexion de flot de données (via Firewall 1) au serveur Client Access Bridge (CLIENT_ACCESS_B) et une nouvelle connexion est établie au serveur Server Access Bridge (SERVER_ACCESS_B).

8 et 9

Le serveur Server Access Bridge (SERVER_ACCESS_B) crée une nouvelle connexion de flot de données (via Firewall 2) au serveur Client Access Bridge (CLIENT_ACCESS_A). Le Client Access Bridge accuse réception de cette connexion et la connexion de la sonde entrante est acceptée à son tour.

Des paquets de données sont à présent acheminés à partir de la sonde, via les connexions ouvertes et les canaux de flots de données initiés par les serveurs pont, et jusqu'au serveur ObjectServer (NCOMS).

Configuration de reprise en ligne du serveur pare-feu

Une architecture de reprise en ligne d'un serveur pare-feu de base inclut tous les composants de l'architecture basique mais également un serveur Server Access Bridge et un serveur Client Access Bridge. En cas d'échec d'une connexion initiale au serveur ObjectServer, la sonde tente de se connecter au serveur ObjectServer à l'aide d'un serveur Server Access Bridge de sauvegarde et d'un serveur Client Access Bridge de sauvegarde.

La figure ci-après illustre la configuration de reprise en ligne d'un serveur pare-feu de base :

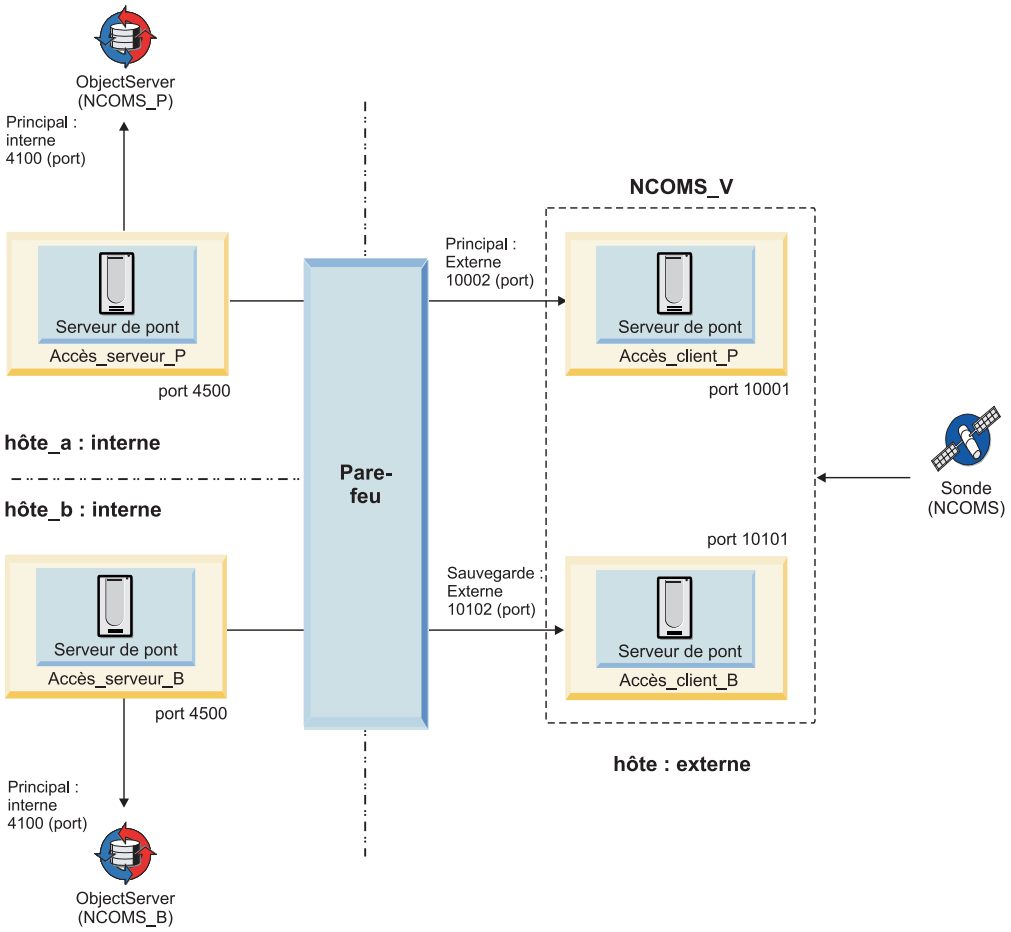

Figure 5. Configuration de base de la reprise en ligne

Dans la configuration de reprise en ligne d'un serveur pare-feu de base, l'hôte_a et l'hôte_b sont déployés à l'intérieur du pare-feu, et un hôte supplémentaire est déployé à l'extérieur du pare-feu.

Le serveur Server Access Bridge (Server_Access_P) s'exécute sur l'hôte_a (interne) et le port 4500. Ce serveur est configuré de sorte qu'il se connecte au serveur ObjectServer (NCOMS_P) à l'aide du nom d'hôte (Primaire) et du port (4100) défini dans le fichier d'interfaces `omni.dat`. Le serveur Server Access Bridge se connecte à un serveur Client Access Bridge correspondant (Client_Access_P), qui s'exécute sur l'hôte externe avec le port du serveur externe : 10002 et le port client : 10001 défini.

Le serveur Server Access Bridge (Server_Access_B) s'exécute sur l'hôte_b (interne) et le port 4500. Ce serveur est configuré de sorte qu'il se connecte au serveur ObjectServer (NCOMS_B) à l'aide du nom d'hôte (Primaire) et du port (4100) défini dans le fichier d'interfaces `omni.dat`. Le serveur Server Access Bridge se connecte à un serveur Client Access Bridge correspondant (Client_Access_B) qui s'exécute sur l'hôte externe avec le port du serveur externe : 10102 et le port client : 10101 défini.

La sonde se connecte initialement au port 10001 du serveur Client Access Bridge (Client_Access_P) avant d'acheminer les données à travers le pare-feu vers le serveur ObjectServer (NCOMS_P) à l'aide du serveur Server Access Bridge (Server_Access_P). Si cette connexion est indisponible, la sonde utilise le port 10101

du serveur Client Access Bridge (Client_Access_B) pour acheminer les données à travers le pare-feu vers le serveur ObjectServer (NCOMS_B) à l'aide du serveur Server Access Bridge (Server_Access_B).

Démarrage du serveur pare-feu

Vous pouvez démarrer le serveur pare-feu automatiquement en utilisant le contrôle de processus sous UNIX et Windows ainsi que les services sous Windows. Vous pouvez également le démarrer manuellement à partir de l'interface de ligne de commande.

Démarrage d'un serveur pare-feu à l'aide du contrôle de processus

Sous UNIX et Windows, vous pouvez démarrer un serveur pare-feu en tant que processus à l'aide du contrôle des processus. Le serveur pare-feu doit être défini en tant que processus ou partie d'un service.

Pourquoi et quand exécuter cette tâche

Concepts associés:

Chapitre 7, «Utilisation du contrôle de processus pour gérer des processus et des procédures externes», à la page 247
Le système de contrôle Tivoli Netcool/OMNIbus effectue deux tâches principales. Il gère les processus locaux et distants, puis exécute les processus externes spécifiés dans les automatisations.

Démarrage d'un serveur pare-feu à l'aide des services Windows

Sous Windows, vous pouvez facultativement installer et exécuter le serveur pare-feu en tant que service Windows. Lorsque le service est défini sur automatique, le serveur pont est lancé au démarrage de l'ordinateur.

Pourquoi et quand exécuter cette tâche

Vous devez installer et configurer manuellement le serveur pare-feu pour l'exécuter en tant que service sur un hôte Windows.

Pour plus d'informations relatives à l'installation et la configuration d'un agent de processus en tant que service Windows, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Démarrage manuel du serveur pare-feu

Utilisez la commande **nco_bridgeserv** pour démarrer le serveur pare-feu manuellement.

Pourquoi et quand exécuter cette tâche

A partir de la ligne de commande, entrez la commande appropriée à votre système d'exploitation :

Tableau 14. Démarrage du serveur pare-feu à partir de l'interface de ligne de commande

Système d'exploitation	Commande
UNIX	`$NCHOME/omnibus/bin/nco_bridgeserv [ -name nom_pont ]`

Tableau 14. Démarrage du serveur pare-feu à partir de l'interface de ligne de commande (suite)

Système d'exploitation	Commande
Windows	`%NCHOME%\omnibus\bin\nco_bridgeserv [ -name nom_pont ]`

Dans les deux commandes, *nom_pont* correspond au nom du serveur pare-feu. Si vous ne spécifiez pas l'option de ligne de commande `-name`, le nom du serveur pare-feu par défaut est `NCO_BRIDGE`.

Vous pouvez démarrer le serveur pare-feu à l'aide d'options de ligne de commande supplémentaires.

Propriétés et options de ligne de commande du serveur pare-feu

Le serveur pare-feu lit son fichier de propriétés lors du démarrage. Si une propriété n'est pas indiquée dans ce fichier, la valeur par défaut est utilisée, sauf si vous utilisez une option de ligne de commande pour la remplacer. L'emplacement par défaut du fichier de propriétés est `$NCHOME/omnibus/etc/serveur_de_pont.props`.

Dans le fichier de propriétés, une propriété et sa valeur correspondante sont séparées par deux points (:). Les valeurs de chaîne sont entourées par des guillemets, par exemple :

```
ServerName: "NCO_BRIDGE"
```

Conseil : Vous pouvez chiffrer les valeurs de chaîne dans un fichier de propriétés en utilisant le chiffrement de valeur de propriété.

Les options de ligne de commande du serveur pare-feu respectent le format suivant :

```
nco_bridgeserv [ -option [ valeur ] ... ]
```

Dans cette commande, *-option* est l'option de ligne de commande et *valeur* est la valeur sur laquelle vous définissez l'option. Toutes les options n'exigent pas que vous indiquiez une valeur.

Si vous n'indiquez pas de fichier de propriétés lors du démarrage d'un serveur pare-feu, le fichier par défaut est utilisé. L'option de ligne de commande `-propsfile` vous permet d'indiquer le chemin d'accès et le nom de fichier complets d'un fichier de propriétés alternatif.

Le tableau suivant répertorie les propriétés et les options de ligne de commande du serveur pare-feu.

Tableau 15. Propriétés et options de ligne de commande du serveur Bridge

Propriété	Option de ligne de commande	Description
AuthPassword *chaîne*	-authpasswd *chaîne*	Mot de passe associé au nom d'utilisateur utilisé pour authentifier le serveur proxy lors de la connexion au serveur ObjectServer s'exécutant 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.
AuthUserName *chaîne*	-authusername *chaîne*	Nom d'utilisateur utilisé pour authentifier le serveur proxy lors de la connexion à un serveur ObjectServer s'exécutant en mode sécurisé. La valeur par défaut est ''.
Bridge.AllowConnections TRUE | FALSE	-disallowconnections	Lorsque cette propriété est définie sur TRUE, un utilisateur peut se connecter au port de commande du serveur pare-feu. La valeur par défaut est TRUE.
Bridge.ClientAP.ClientPort *non signé*	-cpclientport	Indique le port d'écoute client pour les clients Netcool entrants (CLIENT_AP uniquement). La valeur par défaut est 10001.
Bridge.ClientAP.Hostname *chaîne*	-cpclienthname *chaîne*	Indique le nom d'hôte du serveur Client Access Point Bridge(SERVER_AP uniquement). La valeur par défaut est localhost.
Bridge.ClientAP.ServerPort *non signé*	-cpserverport	Indique le port d'écoute du serveur pour les clients Bridge entrants. La valeur par défaut est 10002.
Bridge.PAMEnable TRUE | FALSE	-pamdisable	Lorsque cette propriété est définie sur TRUE, un module PAM est utilisé pour authentifier les connexions client. La valeur par défaut est TRUE.
Bridge.PAMServiceName *chaîne*	-pamsrvname *chaîne*	Indique le nom de service du module PAM utilisé pour l'authentification PAM. La valeur par défaut est nco_bridgeserv.
Bridge.RetryInterval *non signé*	-retryinterval *chaîne*	Indique l'intervalle entre les nouvelles tentatives (en secondes) de communication pont à pont (SERVER_AP uniquement). La valeur par défaut est 30.

Tableau 15. Propriétés et options de ligne de commande du serveur Bridge (suite)

Propriété	Option de ligne de commande	Description
Bridge.Role *chaîne*	-bridgerole *chaîne*	Indique le rôle du serveur pare-feu : CLIENT_AP : Le serveur pare-feu s'exécute comme un pont client. SERVER_AP : Le serveur pare-feu s'exécute comme un pont serveur. La valeur par défaut est SERVER_AP.
Bridge.ServerAP.Server *chaîne*	-spserver *chaîne*	Indique le nom du serveur ObjectServer vers lequel le serveur pare-feu devrait réacheminer les connexions (SERVER_AP uniquement). La valeur par défaut est NCOMS.
Bridge.ThreadPool.MaxJobs *non signé*	-tpmaxjobs	Indique le nombre maximal de flots de données gérés par un travailleur de pool d'unités d'exécution. Le nombre est défini par défaut sur 10.
Bridge.TrustedHostFile *booléen*	-trustedhostfile	Indique le chemin d'accès vers le fichier de sécurité des hôtes sécurisés utilisé par le serveur pare-feu. La valeur par défaut est $OMNIHOME/etc/NCO.thosts. **Remarque :** Cela ne s'applique qu'au serveur pont CLIENT_AP. Le serveur pont SERVER_AP n'accepte aucune connexion entrante.
ConfigCryptoAlg *chaîne*	N/D	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 : • 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**.

Tableau 15. Propriétés et options de ligne de commande du serveur Bridge (suite)

Propriété	Option de ligne de commande	Description
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**.
Connections *entier*	-connections *entier*	Définit le nombre maximal de connexions disponibles pour les clients connectés au port d'administration. La valeur maximale est 1024. La valeur par défaut est 256. Le système peut utiliser jusqu'à deux connexions.
N/D	-help	Affiche les options de ligne de commande prises en charge et quitte.
Ipc.Timeout *entier*	-ipctimeout *entier*	Définit le temps d'attente, en secondes, de l'utilitaire **nco_bridgeserv** pour recevoir une réponse du serveur ObjectServer. La valeur par défaut est 60.
MaxLogFileSize *entier*	-maxlogfilesize *entier*	Indique la taille maximale que peut atteindre le fichier journal (en Ko). La valeur par défaut est 1024 Ko. Une fois la taille spécifiée atteinte, le fichier *nom_serveur*.log est renommé *nom_serveur*.log_OLD et un nouveau fichier journal démarre. Lorsque le nouveau fichier atteint la taille maximale, il est renommé et le processus redémarre.
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 : messages fatal uniquement • error : messages fatal et error • warn : messages fatal, error, et warn • info : messages fatal, error, warn, et info • debug : messages fatal, error, warn, info et debug **Conseil :** La valeur de *chaîne* peut être en majuscules et/ou en minuscules.

Tableau 15. Propriétés et options de ligne de commande du serveur Bridge (suite)

Propriété	Option de ligne de commande	Description
MessageLog *chaîne*	-messagelog *chaîne*	Indique le chemin d'accès à l'emplacement de consignation des messages. Le chemin d'accès par défaut est $NCHOME/omnibus/log/NCOMS.log. Windows Si le système ne peut pas écrire dans le fichier journal spécifié (par exemple, en cas d'erreur fatale), il écrit l'erreur dans un fichier nommé %NCHOME%\omnibus\log\nco_objserv*.err.
Name *chaîne*	-name *chaîne*	Définit le nom du serveur pare-feu, qui doit être unique. Il s'agit du nom configuré dans l'éditeur de serveur. La valeur par défaut est NCO_BRIDGE.
Props.CheckNames TRUE | FALSE	N/D	Lorsque cette propriété est définie sur TRUE, le serveur pare-feu ne s'exécute pas si une des propriétés indiquées est invalide. La valeur par défaut est TRUE.
PropsFile *chaîne*	-propsfile *chaîne*	Définit le nom du fichier de propriétés du serveur pare-feu. Le nom par défaut est *nom_serveur*.props, où *nom_serveur* est défini par l'option -name.
SecureMode TRUE | FALSE	-secure	Définit le mode de sécurité du serveur pare-feu. Si cet paramètre est activé, le serveur pare-feu authentifie les demandes de connexions client à l'aide d'un nom d'utilisateur et d'un mot de passe. Si il est désactivé (valeur par défaut), les clients peuvent se connecter au serveur pare-feu sans nom d'utilisateur ni mot de passe.
UniqueLog TRUE | FALSE	-uniquelog	Si cette propriété est définie sur TRUE, le fichier journal est uniquement nommé en ajoutant l'ID de processus du serveur pont dans le nom du fichier journal par défaut. Par exemple, si le serveur pare-feu NCO_BRIDGE s'exécute en tant que processus 1234, le fichier journal est nommé NCHOME/omnibus/log/NCO_BRIDGE.1234.log. La valeur par défaut est FALSE. Si la propriété **MessageLog** est définie sur stderr oustdout, la propriété **UniqueLog** est ignorée.
N/D	-version	Affiche les informations de version sur le serveur pont et quitte.

Fichier de définition des hôtes sécurisés

Le serveur Client Access Bridge est souvent déployé sur un réseau non sécurisé et fournit un accès distant au serveur ObjectServer situé sur un réseau interne. Pour éviter les accès non autorisés au serveur ObjectServer, le serveur Client Access Bridge utilise le fichier de définition des hôtes sécurisés pour déterminer quels sont les hôtes autorisés à accéder à son port client.

Le fichier de définition des hôtes sécurisés est un fichier texte qui répertorie tous les hôtes autorisés à accéder au port client du serveur Client Access Bridge. La connexion sera supprimée si l'hôte tentant de se connecter au port client n'est pas répertorié dans le fichier de définition des hôtes sécurisés. Si le fichier de définition

des hôtes sécurisés contient au moins un entrée, toutes les connexions entrantes doivent correspondre à cette entrée. Si le fichier de définition des hôtes sécurisés ne contient aucune entrée, toutes les connexions entrantes seront acceptées.

Le fichier de définition des hôtes sécurisés est disponible à l'emplacement par défaut suivant : `$OMNIHOME/etc/NCO_BRIDGE.thosts` (UNIX) ou `%OMNIHOME%\etc\NCO_BRIDGE.thosts` (Windows).

Remarque : Le fichier des hôtes sécurisés n'est consulté que pour les connexions entrantes vers le serveur Client Access Bridge ; il n'est pas consulté pour le serveur Server Access Bridge.

Syntaxe

Le fichier de définition des hôtes sécurisés accepte les entrées aux formats suivants : adresses IP Pv4 ou IPv6 (voir exemple ci-après), caractères génériques et nom d'hôte lisible à l'oeil. De plus, les commentaires sont pris en charge.

```
  #
  # Fichier des hôtes sécurisés
  #
  # IPv4 address - Correspond uniquement à cette adresse IPv4 :
  192.168.1.1

  # IPv4 address/netmask - Correspond à n'importe quelle adresse IPv4 du réseau
192.168.1.0 :
  192.168.1.0/255.255.255.0

  # IPv4 CIDR notation - Correspond à n'importe quelle adresse IPv4 du réseau
192.168.1.0 :
  192.168.1.0/24

  # Hostname - Correspond à ce nom d'hôte uniquement :
  darkstar.example.com

  # Hostname with wildcard - Correspond à tous les hôtes du domaine ibm.com :
  *.ibm.com

  # IPv6 address - Correspond à cette adresse IPv6 uniquement:
  [3ffe:1900:4545:3:200:f8ff:fe21:67cf]

  # IPv6 CIDR notation - Correspond à n'importe quelle adresse IPv6 du réseau
  3ffe:1900:4545:3:200:f8ff:fe21:0000 :
  [3ffe:1900:4545:3:200:f8ff:fe21:67cf]/120
```

Désactivation de l'accès au port de commande interactif sur le serveur Client Access Bridge

Pour plus de sécurité, toute connexion établie au serveur Client Access Bridge à l'aide de l'interface interactive SQL peut être désactivée. Pour ce faire, définissez la propriété **`Bridge.AllowConnections`** sur `FALSE` ou exécutez le serveur Client Access Bridge à l'aide de l'argument de ligne de commande `-disallowconnections`. Cela peut être obligatoire si le serveur Client Access Bridge est en cours d'exécution sur un réseau non-sécurisé à l'extérieur d'un pare-feu.

Langage de commande d'un serveur de pont pare-feu

Le serveur pare-feu fournit une interface de commande SQL à des fins de configuration et d'administration. Vous pouvez utiliser l'interface interactive SQL pour vous connecter à un serveur pare-feu et pour exécuter les commandes de ce serveur.

Avant de commencer à utiliser le langage de commande du pare-feu, assurez-vous d'être familiarisé avec l'interface interactive SQL, comment la démarrer et comment y exécuter des commandes SQL.

Concepts associés:

«interface interactive SQL», à la page 229
Vous pouvez utiliser l'utilitaire de ligne de commande de l'interface interactive SQL fourni avec Tivoli Netcool/OMNIbus pour émettre des commandes SQL en direction d'un serveur ObjectServer.

SHOW PROPS et GET CONFIG

Les commandes SHOW PROPS et GET CONFIG permettent de répertorier toutes les propriétés du serveur pare-feu et leurs valeurs associées.

Syntaxe

```
SHOW PROPS;
GET CONFIG;
```

Exemple

Pour répertorier toutes les propriétés de serveur pare-feu et leurs valeurs associées :

```
1> show props;
2> go
 Property Name                    Type Property Value
 -------------------------------- ---- ------------------------------------------------
 AuthPassword                     7
 AuthUserName                     7
 Bridge.AllowConnections          4    TRUE
 Bridge.ClientAP.ClientPort       3    10001
 Bridge.ClientAP.Hostname         7    omnihost
 Bridge.ClientAP.ServerPort       3    10002
 Bridge.PAMEnable                 4    TRUE
 Bridge.PAMServiceName            7    netcool
 Bridge.Role                      7    SERVER_AP
 Bridge.ServerAP.Server           7    NCOMS
 Bridge.ThreadPool.MaxJobs        3    10
 Bridge.TrustedHostFile           7    /opt/ibm/netcool/omnibus/etc/NCO_BRIDGE.thosts
 ConfigCryptoAlg                  7    AES
 ConfigKeyFile                    7
 Connections                      2    30
 Help                             4    FALSE
 Ipc.QueueSize                    2    1024
 Ipc.ServerCharacterSet           7    iso_1
 Ipc.ServerLanguage               7    us_english
 Ipc.ServerLocale                 7    default
 Ipc.SingleThreaded               4    FALSE
 Ipc.SSLCertificate               7
 Ipc.SSLEnable                    4    FALSE
 Ipc.SSLPrivateKeyPassword        7
 Ipc.StackSize                    2    131072
 Ipc.Timeout                      2    60
 Ipc.TruncateVendorLogFile        4    TRUE
 Ipc.VendorClientLibraryVersion   7    version string
 Ipc.VendorLogFileSize            2    1024
 Ipc.VendorServerLibraryVersion   7    version string
 MaxLogFileSize                   2    1024
 MessageLevel                     7    debug
 MessageLog                       7    stdout
 Name                             7    NCO_BRIDGE
```

```
PAAwareID                       2    0
PAServerName                    7
Props.CheckNames                4    TRUE
PropsFile                       7    /opt/ibm/netcool/omnibus/etc/NCO_BRIDGE.props
SecureMode                      4    FALSE
UniqueLog                       4    FALSE
Version                         4    FALSE
```

GET PROP

Utilisez la commande GET PROP pour renvoyer la valeur d'une propriété du serveur pare-feu caractéristique.

Syntaxe

```
GET PROP[ERTY] <nomprop>;
```

Exemple

Pour afficher une propriété du serveur pare-feu et sa valeur associée :

```
1> get prop 'Name';
2> go
 Property Name                     Type Property Value
 --------------------------------- ---- ---------------
 Name                              7    NCO_BRIDGE
```

SHOW DATAFLOWS

La commande SHOW DATAFLOWS permet de répertorier les flots de données actives via le serveur pare-feu.

Syntaxe

```
SHOW DATAFLOWS;
```

Exemple

Pour répertorier les flots de données actives via un serveur pare-feu :

```
1> show dataflows;
2> go
 TPWorkerName           DataflowID  SPSocketFD  CPSocketFD  SPBytes CPBytes
 ---------------------- ----------- ----------- ----------- ------- -------
 TPWorkerThread_0x74340     4033280          16          17      81     615
```

La sortie représente une seule connexion de flot de données actives via le serveur pare-feu, où chaque colonne représente les éléments suivants :

TPWorkerName	Le nom interne de l'unité d'exécution.
DataflowID	L'ID interne du flot de données.
SPSocketFD	Le descripteur du fichier de système d'exploitation côté serveur de la connexion.
CPSocketFD	Le descripteur du fichier de système d'exploitation côté client de la connexion.
SPBytes	Le nombre d'octets transférés du serveur à la connexion client.
CPBytes	Le nombre d'octets transférés du client à la connexion serveur.

SET LOG LEVEL TO

La commande SET LOG LEVEL TO permet d'indiquer le niveau de journalisation.

Syntaxe

```
SET LOG LEVEL TO niveau;

Où niveau prend une des valeurs suivantes :

debug

information

warning

error

fatal
```

Exemple

Pour répertorier toutes les propriétés du serveur pare-feu et leurs valeurs associées :

```
1> set log level to DEBUG;
2> go
```

SHUTDOWN

La commande SHUTDOWN permet d'ordonner au serveur pare-feu de s'arrêter.

Syntaxe

```
shutdown;
```

Exemple

Pour arrêter le serveur pare-feu :

```
shutdown;
```

Chapitre 4. Utilisation de Netcool/OMNIbus Administrator pour configurer les serveurs ObjectServer

Le serveur ObjectServer stocke, gère et traite les données d'alerte et de statut collectées par des applications externes telles que des sondes et des passerelles. Vous pouvez utiliser Netcool/OMNIbus Administrator pour configurer vos objets du serveur ObjectServer et configurer le contrôle du processus.

Vous pouvez utiliser Netcool/OMNIbus Administrator pour configurer les objets du serveur ObjectServer suivants :

- Utilisateurs, groupes, rôles et filtres de restriction
- Menus de liste des événements
- Outils et invites
- Groupes de déclenchement et déclencheurs
- Procédures
- Signaux définis par l'utilisateur
- Couleurs de gravité d'alerte de liste d'événements (listes d'événements Windows uniquement)
- Conversions
- Classes
- Visuels de colonne
- Bases de données, fichiers et propriétés du serveur ObjectServer
- Canaux pour la notification des événements accélérés

Démarrage de Netcool/OMNIbus Administrator

Netcool/OMNIbus Administrator possède une interface virtuelle à partir de laquelle vous pouvez gérer vos serveurs ObjectServer et configurer le contrôle de processus.

Remarques relatives au support multiculturel

Tivoli Netcool/OMNIbus prend en charge divers codages de caractères à octet unique et multi-octets dans différents paramètres nationaux.

Si des noms d'utilisateur et des mots de passe sont spécifiés en caractères multi-octets et que ces données d'identification doivent être vérifiées par rapport à des sources d'authentification externes, ces sources doivent également prendre en charge les caractères multi-octets. Si les caractères multi-octets ne sont pas pris en charge, vous devez spécifier les noms d'utilisateur et les mots de passe à l'aide de caractères ASCII.

Si vous utilisez Netcool/OMNIbus Administrator, vous devez vous assurer que le codage du jeu de caractères de chaque ObjectServer géré possède une entrée correspondante dans le fichier `$NCHOME/omnibus/java/jars/csemap.dat`. Ce fichier fournit un mappage entre les conventions de dénomination de codage Sybase et du jeu de caractères de l'environnement d'exécution Java. Si le codage du jeu de caractères d'un serveur ObjectServer ne figure pas dans le fichier `csemap.dat`, vous devez ajouter un mappage à ce fichier au format suivant :

codage_Sybase codage_Java

© Copyright IBM Corp. 1994, 2014

Par exemple :

```
ascii_7 ASCII
```

Pour plus d'informations sur le support multiculturel, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Démarrage de Netcool/OMNIbus Administrator

Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Pourquoi et quand exécuter cette tâche

Pour démarrer Netcool/OMNIbus Administrator à partir de la ligne de commande, procédez comme suit :

Procédure

1. Entrez la commande appropriée pour votre système d'exploitation :

Tableau 16. Démarrage de Netcool/OMNIbus Administrator

Option	Description
UNIX	`$NCHOME/omnibus/bin/nco_config [ -option valeur ... ]`
Windows	`%NCHOME%\omnibus\bin\nco_config.vbs [ -option valeur ... ]`

   Dans cette commande, *-option* est une option de ligne de commande valide et *valeur* la valeur que vous attribuez à l'option.
2. S'il s'agit du premier démarrage de Netcool/OMNIbus Administrator ou que votre fichier de paramètres de communication (`$NCHOME/etc/omni.dat` sous UNIX et `%NCHOME%\ini\sql.ini` sous Windows) a été modifié depuis le dernier démarrage de Netcool/OMNIbus Administrator, l'assistant d'importation des connexions est exécuté. Cet assistant permet de choisir les serveurs ObjectServer et agents de processus que vous souhaitez configurer à l'aide de Netcool/OMNIbus Administrator.

   Conseil : Après le démarrage de Netcool/OMNIbus Administrator, vous pouvez à tout moment sélectionner **File** > **Import** (Fichier > Importer) pour importer de nouvelles informations de communication à partir du fichier `omni.dat` (`sql.ini` sous Windows). Pour plus d'informations sur la configuration des communications de composant, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Propriétés et options de ligne de commande Netcool/OMNIbus Administrator

Netcool/OMNIbus Administrator contient un ensemble de propriétés et d'options de ligne de commande pour configurer le composant.

Le fichier de propriétés Netcool/OMNIbus Administrator par défaut est `$NCHOME/omnibus/etc/nco_config.props` (`%NCHOME%\omnibus\etc\nco_config.props` sous Windows). Le fichier de propriétés par défaut est lu chaque fois que vous démarrez Netcool/OMNIbus Administrator ; toutefois, vous pouvez utiliser l'option de ligne de commande `-propsfile` pour spécifier un autre fichier de propriétés.

Vous pouvez utiliser le fichier de propriétés comme modèle et le modifier pour différents objectifs. Par exemple, vous pouvez utiliser différents fichiers de propriétés pour vous connecter à différents serveurs ObjectServer.

Les propriétés et options de ligne de commande de **nco_config** (**nco_config.vbs** sous Windows) sont décrites dans la table suivante.

Tableau 17. Propriétés et options de ligne de commande Netcool/OMNIbus Administrator

Propriété	Option de ligne de commande	Description
audit.active 0 | 1	-auditlogactive 0 | 1	Détermine si la consignation dans le journal d'audit est activée. La valeur par défaut est 1 (consignation dans le journal d'audit activée).
audit.file.max.count *entier*	-auditfilecount *entier*	Nombre maximal de fichiers journaux d'audit Netcool/OMNIbus Administrator. La valeur par défaut est 4.
audit.file.max.size *entier*	-auditfilesize *entier*	Taille de fichier maximale (en octets) des fichiers journaux d'audit Netcool/OMNIbus Administrator. La valeur par défaut est 10000.
audit.file.name *chaîne*	-auditfile *chaîne*	Chemin d'accès complet au fichier journal d'audit Netcool/OMNIbus. La valeur par défaut est $NCHOME/omnibus/log/nco_config_audit.log.
Non applicable	-help	Affiche l'aide sur les options de ligne de commande et quitte.
java.security.policy *chaîne*	-policyfile *chaîne*	Chemin d'accès complet au fichier de règles de sécurité Java.
log.console.active 0 | 1	-logtoconsole 0 | 1	Détermine si les informations de consignation sont envoyées à l'interpréteur de commandes. La valeur par défaut est 1 (les informations de consignation sont envoyées à l'interpréteur de commandes).
log.directory.name *chaîne*	-logdir *chaîne*	Emplacement de sauvegarde du fichier journal du système Netcool/OMNIbus Administrator. L'emplacement par défaut est $NCHOME/omnibus/log.
log.file.max.count *entier*	-logfilecount *entier*	Nombre maximal de fichier journaux du système Netcool/OMNIbus Administrator. La valeur par défaut est4.
log.file.max.size *entier*	-logfilesize *entier*	Taille de fichier maximale (en octets) des fichiers journaux du système Netcool/OMNIbus Administrator. La valeur par défaut est 10000.
log.file.name *chaîne*	-logfile *chaîne*	Nom du fichier journal du système Netcool/OMNIbus. La valeur par défaut est nco_config_system.log. **Conseil :** Pour modifier le répertoire de stockage de ce fichier, utilisez la propriété **log.directory.name** ou l'option -logdir.

Tableau 17. Propriétés et options de ligne de commande Netcool/OMNIbus Administrator (suite)

Propriété	Option de ligne de commande	Description
messagelevel *chaîne*	-messagelevel *chaîne*	Spécifie le niveau de consignation des messages pour la consignation dans le journal d'audit et dans le système. Les valeurs possibles sont les suivantes : FATAL, ERROR, WARN, INFO et DEBUG. Le niveau par défaut est ERROR. Les messages qui sont consignés à chaque niveau sont les suivants : FATAL : messages FATAL uniquement. ERROR : messages FATAL et ERROR. WARN : messages FATAL, ERROR et WARN. INFO : messages FATAL, ERROR, WARN et INFO. DEBUG : messagesFATAL, ERROR, WARN, INFO et DEBUG. **Remarque :** La valeur de la *chaîne* doit être en majuscules.
nco_jdbc.timeout *entier*	-jdbctimeout *entier*	Délai d'attente de connectivité JDBC (Java Database Connectivity) en secondes. La valeur par défaut est 600.
Non applicable	-propsfile *chaîne*	Chemin d'accès complet au fichier de propriétés Netcool/OMNIbus Administrator. La valeur par défaut est $NCHOME/omnibus/etc/nco_config.props (%NCHOME%\omnibus\etc\nco_config.props sous Windows).
server *chaîne*	-server *chaîne*	Nom du serveur ObjectServer auquel vous vous connectez.
system.create.conversion 0 | 1	-createconversion 0 | 1	Configure le système afin qu'il crée automatiquement la conversion pour les utilisateurs que vous créez. La valeur par défaut est 1 (les conversions sont automatiquement créées).
system.conversion.type *chaîne*	-conversiontype *chaîne*	Crée une conversion entre l'ID utilisateur et le nom d'utilisateur ou le nom complet de chaque utilisateur nouvellement créé. Les valeurs possibles sont fullname et username. La valeur par défaut est fullname (crée une conversion entre l'ID utilisateur et le nom complet).
user.name *chaîne*	-user *chaîne*	Nom d'utilisateur de connexion à Netcool/OMNIbus Administrator.
user.password *chaîne*	-password *chaîne*	Mot de passe de connexion à Netcool/OMNIbus Administrator.
Non applicable	-version	Affiche les informations de version et quitte.

Traitement des propriétés et des lignes de commandes

Chaque propriété du fichier de propriétés Netcool/OMNIbus Administrator possède une valeur par défaut. Dans un fichier de propriétés non modifié, les propriétés sont répertoriées avec leurs valeurs par défaut et signalées par le symbole carré (#) en début de ligne. Une propriété et sa valeur correspondante sont séparées par deux points (:).

Vous pouvez modifier les valeurs d'une propriété à l'aide d'un éditeur de texte. Pour remplacer la valeur par défaut, modifiez un paramètre dans le fichier de propriétés et supprimez le symbole carré (#).

Si vous modifiez un paramètre dans la ligne de commande, la valeur par défaut et le paramètre du fichier de propriétés sont remplacés. Pour simplifier la commande que vous entrez pour exécuter **nco_config**, ajoutez autant de propriétés que possible dans le fichier de propriétés au lieu d'utiliser les options de ligne de commande.

Connexion à un serveur ObjectServer

Après le démarrage de Netcool/OMNIbus Administrator, vous devez vous connecter au serveur ObjectServer que vous souhaitez configurer.

Pourquoi et quand exécuter cette tâche

Pour vous connecter à un serveur ObjectServer, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Reports** (Rapports) dans le panneau de gauche.

Figure 6. Boutons de menu Navigator (Navigateur) et Reports (Rapports)

2. Cliquez sur **OS**. La fenêtre ObjectServer Report (Rapport ObjectServer) s'ouvre. Elle affiche tous les serveurs ObjectServer sélectionnés lors de la dernière exécution de l'assistant d'importation des connexions.
3. Cliquez à l'aide du bouton droit de la souris sur le serveur ObjectServer auquel vous souhaitez vous connecter. Dans le menu contextuel, procédez de l'une des façons suivantes :
   - Cliquez sur **Connect As** (Se connecter sous) s'il s'agit de votre première connexion ou si vous souhaitez entrer des informations de connexion mises à jour. Vous êtes invité à entrer un nom d'utilisateur et un mot de passe pour le serveur ObjectServer.

     Si vous cochez la case **Always use for this connection** (Toujours utiliser pour cette connexion), le nom d'utilisateur et le mot de passe sont automatiquement sauvegardés et réutilisés pour les connexions à cet ObjectServer. Si vous cochez la case **Use as default** (Utiliser par défaut), les valeurs spécifiées pour le nom d'utilisateur et le mot de passe sont automatiquement renseignées lors de l'affichage suivant de la fenêtre de connexion. Si vous cochez les deux cases, l'option **Always use for this connection** (Toujours utiliser pour cette connexion) l'emporte. Ces paramètres durent tout le long de la session d'application.

- Cliquez sur **Connect** (Connexion) pour utiliser les informations de connexion précédemment spécifiées.

4. Cliquez sur **OK**.

Résultats

Le panneau central est la fenêtre de configuration du serveur ObjectServer. Dans cette zone de travail, vous pouvez afficher, modifier et gérer les objets ObjectServer.

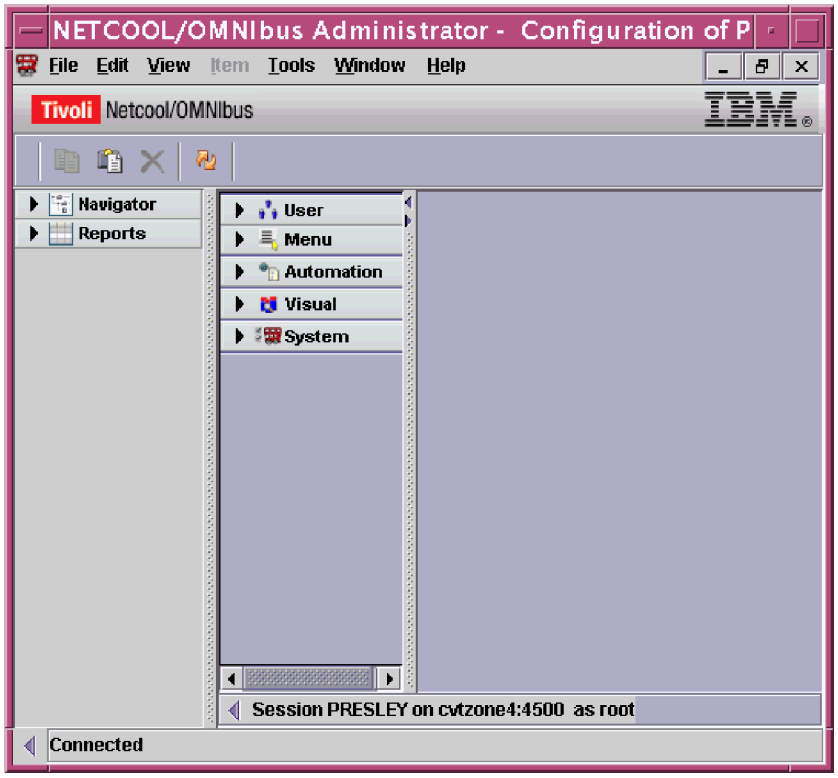

Figure 7. Netcool/OMNIbus Administrator - Fenêtre de configuration du serveur ObjectServer

Tâches associées:

«Sélection d'objets ObjectServer à configurer», à la page 78
Un ensemble de boutons de menu est disponible sur le côté gauche de la fenêtre Netcool/OMNIbus Administrator. Ces boutons permettent de sélectionner l'objet ObjectServer que vous souhaitez configurer.

Connexion à un agent de processus

Avant de vous connecter à un agent de processus, vous devez vous assurer qu'il a été démarré.

Pourquoi et quand exécuter cette tâche

Pour vous connecter à un agent de processus, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Reports** (Rapports) dans le panneau de gauche.

Figure 8. Boutons de menu Navigator (Navigateur) et Reports (Rapports)

2. Cliquez sur **PA**. Le panneau Process Agent Report (Rapport de l'agent de processus) s'ouvre et affiche tous les agents de processus sélectionnés lors de la dernière exécution de l'assistant d'importation des connexions.

   Conseil : Après le démarrage de Netcool/OMNIbus Administrator, vous pouvez à tout moment cliquer sur **File** > **Import** (Fichier > Importer) pour importer de nouvelles informations de communication à partir du fichier `omni.dat` (`sql.ini` sous Windows). Pour plus d'informations sur la configuration des communications de composant, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.
3. Cliquez à l'aide du bouton droit de la souris sur l'agent de processus. Dans le menu contextuel, procédez de l'une des façons suivantes :
   - Cliquez sur **Connect As** (Se connecter sous) s'il s'agit de votre première connexion ou si vous souhaitez entrer des informations de connexion mises à jour. La fenêtre Process Agent Security (Sécurité de l'agent de processus) s'ouvre. Complétez cette fenêtre comme suit :

     Nom d'utilisateur
     Entrez le nom d'utilisateur permettant de vous connecter à l'agent de processus.

     Sous UNIX, tout utilisateur devant accéder à l'agent de processus doit être membre d'un groupe d'utilisateurs UNIX défini comme groupe d'administration à cette fin. Sous Windows, l'utilisateur doit être un utilisateur valide avec un compte local ou de domaine.

     Mot de passe
     Entrez le mot de passe utilisé pour se connecter à l'agent de processus.

     Always use for this connection (Toujours utiliser pour cette connexion)
     Cochez cette case pour indiquer que le nom d'utilisateur et le mot de passe doivent être sauvegardés et utilisés automatiquement lors des tentatives ultérieures de connexion à cet agent de processus. Ces paramètres sont valides pour toute la durée de la session d'application.

     Use as default (Utiliser comme valeur par défaut)
     Cochez cette case si vous voulez que les valeurs spécifiées pour le nom d'utilisateur et le mot de passe soient automatiquement renseignées lors du prochain affichage de cette fenêtre. Ces paramètres sont valides pour toute la durée de la session d'application.

     Remarque : Si vous cochez les deux cases, le paramètre **Always use for this connection** (Toujours utiliser pour cette connexion) prévaut.
   - Cliquez sur **Connect** (Connexion) pour utiliser les informations de connexion précédemment spécifiées.
4. Cliquez sur **OK**.

   La fenêtre Service/Process Details (Détails du service/processus) s'ouvre. Elle contient des informations relatives aux processus et services configurés dans cet agent de processus.

Concepts associés:

Chapitre 7, «Utilisation du contrôle de processus pour gérer des processus et des procédures externes», à la page 247
Le système de contrôle Tivoli Netcool/OMNIbus effectue deux tâches principales. Il gère les processus locaux et distants, puis exécute les processus externes spécifiés dans les automatisations.

Utilisation des composants Tivoli Netcool/OMNIbus

Dans la barre de boutons du panneau de gauche de la fenêtre Netcool/OMNIbus Administrator, vous devez sélectionner les composants à afficher ou configurer.

- Sélectionnez le bouton de menu **Reports**, puis cliquez sur **Host**, **OS** ou **PA**. Sélectionnez un serveur ObjectServer pour effectuer la configuration à partir de la fenêtre ObjectServer Report ou un agent de processus pour effectuer la configuration à partir de la fenêtre PA Report.
- Sélectionnez le bouton de menu **Navigator** pour visualiser les composants par nom d'hôte, puis sélectionnez le composant à configurer.

La figure suivante affiche les boutons de menu utilisés pour sélectionner un composant.

Figure 9. Boutons de menu Navigator (Navigateur) et Reports (Rapports)

Lorsque vous sélectionnez un composant, la fenêtre ou le panneau associé s'affiche dans la zone de droite. Les boutons de la barre d'outils et éléments de menu disponibles dans la fenêtre Netcool/OMNIbus Administrator changent en fonction de votre sélection.

Connexions SSL

Si Netcool/OMNIbus Administrator utilise une connexion SSL cryptée à un serveur ObjectServer ou un agent de processus, une icône de verrouillage s'affiche dans le coin inférieur gauche de la fenêtre.

Pour plus d'informations sur l'utilisation de la connexion SSL avec Tivoli Netcool/OMNIbus, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Validation des certificats serveur

Si Tivoli Netcool/OMNIbus est configuré pour la communication SSL, le serveur ObjectServer et l'agent de processus présentent leurs certificats serveur au client Netcool/OMNIbus Administrator sur demande afin d'établir une connexion.

Pourquoi et quand exécuter cette tâche

Si une non concordance est détectée entre le nom usuel défini dans le certificat serveur et le nom de serveur qu'utilise le client Netcool/OMNIbus Administrator pour identifier et se connecter au serveur, une fenêtre Certificate Validation (Validation du certificat) s'ouvre et vous propose d'accepter ou de refuser le certificat serveur. Les connexions ne sont pas établies si le certificat n'est pas valide.

La fenêtre Certificate Validation (Validation du certificat) explique la demande de validation et présente un certain nombre d'options. Renseignez la fenêtre comme suit :

Procédure

1. Sélectionnez l'une des options suivantes afin d'accepter ou de refuser le certificat :
   - **Accept this certificate permanently** (Accepter définitivement ce certificat) : sélectionnez cette option pour accepter définitivement ce certificat valide. Vous ne serez plus invité à l'accepter lors des sessions Netcool/OMNIbus Administrator en cours ou ultérieures.

     Important : Avant d'accepter le certificat, cliquez sur **Examine Certificate** (Examiner le certificat) pour consulter son contenu dans la fenêtre Certificate Details (Détails du certificat). Après l'avoir lu attentivement, cliquez sur **OK** pour retourner à la fenêtre Certificate Validation (Validation du certificat).
   - **Accept this certificate temporarily for this session** (Accepter provisoirement ce certificat pour cette session) : sélectionnez cette option pour accepter le certificat uniquement pour la session en cours, après l'avoir consulté en cliquant sur le bouton **Examine Certificate** (Examiner le certificat). Plus aucune invite ne s'affiche pendant la durée de la session.
   - **Do not accept this certificate** (Ne pas accepter ce certificat) : sélectionnez cette option pour refuser le certificat et annuler la connexion entre le serveur et le client.
2. Cliquez sur **OK** pour poursuivre le processus de connexion. Cliquez sur **Cancel** (Annuler) (ou sur le bouton **Close** (Fermer) de la barre de titre) pour refuser le certificat quelle que soit l'option sélectionnée à l'étape 1.

Résultats

Si vous décidez d'accepter définitivement le certificat, le nom usuel et la clé publique provenant du certificat sont enregistrés dans le fichier suivant :

```
répertoire_utilisateur/.netcool/nco_config_settings/
user_allowed_certs.properties
```

Dans ce chemin, *répertoire_utilisateur* représente votre répertoire de base.

Le fichier `user_allowed_certs.properties` est un fichier système et ne doit pas être modifié par les utilisateurs. Lors des tentatives de connexion suivantes, ce fichier est lu et utilisé pour identifier tous les noms usuels qui ont été préalablement acceptés.

Vous pouvez effacer le contenu du fichier de propriétés en spécifiant l'argument de ligne de commande suivant :

```
mode.clear.certs "true"
```

Pour plus d'informations relatives aux certificats SSL, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Sélection d'objets ObjectServer à configurer

Un ensemble de boutons de menu est disponible sur le côté gauche de la fenêtre Netcool/OMNIbus Administrator. Ces boutons permettent de sélectionner l'objet ObjectServer que vous souhaitez configurer.

Pourquoi et quand exécuter cette tâche

Ces boutons de menu sont libellés comme suit :

Procédure

- User (Utilisateur)
- Menu
- Automation (Automatisation)
- Visual (Visuel)
- Système

Résultats

La figure ci-dessous illustre ces boutons.

Figure 10. Boutons de menu User (Utilisateur), Menu, Automation (Automatisation), Visual (Visuel) et System (Système)

Cliquez sur chaque bouton de menu afin d'obtenir un ensemble d'objets connexes que vous pouvez configurez. Les boutons de la barre d'outils et éléments de menu disponibles dans la fenêtre Netcool/OMNIbus Administrator changent en fonction de votre sélection.

Gestion des objets

Lorsque vous gérez un objet, il existe plusieurs façons de sélectionner une option : la barre de menus, la barre d'outils et un menu contextuel. Par exemple, si vous souhaitez créer un utilisateur, vous devez en premier lieu sélectionnez le bouton de menu **User** (Utilisateur), puis cliquer sur **Users** (Utilisateurs) pour ouvrir le panneau Users (Utilisateurs). Ensuite, vous pouvez réaliser l'une des actions suivantes pour créer l'utilisateur :

- Cliquez sur **Item** > **Add User** (Elément > Ajouter un utilisateur).
- Dans la barre d'outils, cliquez sur **Add User** (Ajouter un utilisateur).
- Dans le panneau **Users** (Utilisateurs), cliquez à l'aide du bouton droit de la souris et sélectionnez **Add User** (Ajouter un utilisateur) dans le menu contextuel.

  Si vous éditez ou supprimez des objets, vous pouvez réaliser un ensemble d'actions équivalent. Si vous éditez des objets, vous pouvez également cliquer deux fois sur l'objet pour ouvrir la fenêtre d'édition équivalente.

Conseil : Vous pouvez également utiliser des commandes SQL pour interagir avec le serveur ObjectServer et le configurer.

Définition des préférences dans Netcool/OMNIbus Administrator

Vous pouvez définir les préférences de Netcool/OMNIbus Administrator en triant les tables de résultats, en définissant la présentation de l'affichage des colonnes à l'aide des vues et en sélectionnant les lignes à afficher à l'aide de filtres. Vous pouvez également copier et coller, configurer les couleurs de la syntaxe de l'éditeur et sélectionner un navigateur Web pour afficher l'aide en ligne.

Tri des tables de résultats

La plupart des fenêtres Netcool/OMNIbus Administrator (y compris les fenêtres Host Report (Rapport de l'hôte), PA Report (Rapport PA) et ObjectServer Report (Rapport du serveur ObjectServer), les résultats SQL et les tables de base de données) affichent des informations sous la forme de tables de résultats.

Pourquoi et quand exécuter cette tâche

Vous pouvez cliquer sur un titre de colonne pour trier les lignes de la table en fonction des valeurs de cette colonne. Cliquez plusieurs fois pour sélectionner le tri dans l'ordre croissant ou décroissant, indiqué par la flèche orientée vers le haut ou le bas en regard du titre de colonne.

Définition de l'affichage des colonnes à l'aide des vues

Vous pouvez modifier l'affichage des colonnes dans les tables de résultat, qui contiennent la fenêtre Host Report (Rapport de l'hôte), PA Report (Rapport PA), ObjectServer Report (Rapport du serveur ObjectServer), les résultats SQL et les tables de base de données.

Pourquoi et quand exécuter cette tâche

Pour masquer une colonne, cliquez à l'aide du bouton droit de la souris sur l'en-tête de colonne, puis cliquez sur **Hide Column** (Masquer la colonne).

Pour aligner une colonne, cliquez à l'aide du bouton droit de la souris sur l'en-tête de colonne, puis cliquez sur **Align Column** (Aligner la colonne). Ensuite, cliquez sur **Left** (Gauche), **Right** (Droite) ou **Center** (Centre).

Pour sélectionner les colonnes à afficher dans une liste de toutes les colonnes, procédez comme suit :

Procédure

1. Cliquez à l'aide du bouton droit de la souris sur l'en-tête de colonne, puis cliquez sur **Table Columns** (Colonnes de table). La fenêtre Column Settings (Paramètres de la colonne) s'ouvre.
2. Sélectionnez les colonnes à afficher dans la fenêtre active. Vous pouvez cliquer sur un nom de colonne pour la sélectionner ou la désélectionner. Une coche se trouve en regard des colonnes sélectionnées pour être affichées. Cliquez sur **Fermer** pour sauvegarder vos paramètres.

Sélection de lignes à afficher à l'aide de filtres

Vous pouvez modifier l'affichage des lignes des tables de résultats, qui contiennent les fenêtres Host Report (Rapport de l'hôte), PA Report (Rapport PA) et ObjectServer Report (Rapport du serveur ObjectServer), les résultats SQL et les tables de base de données, en créant des conditions de filtre.

Pourquoi et quand exécuter cette tâche

Pour créer un filtre, procédez comme suit :

1. Cliquez à l'aide du bouton droit de la souris sur l'en-tête de colonne auquel vous souhaitez appliquer un filtre, puis cliquez sur **Filter On** (Appliquer un filtre). La fenêtre Filter (Filtre) s'ouvre.
2. Sélectionnez la colonne, l'opération de filtrage et la valeur de la condition.

Dans la fenêtre Filter (Filtre), vous pouvez :

- Cliquez sur **OK** pour créer le filtre et fermer la fenêtre
- Cliquez sur **Apply** (Appliquer) pour créer et conserver la fenêtre ouverte
- Cliquez sur **Cancel** (Annuler) pour fermer la fenêtre sans créer de filtre

Pour créer un filtre en fonction des valeurs d'une cellule de table particulière, procédez comme suit :

1. Cliquez à l'aide du bouton droit de la souris dans la cellule, puis sélectionnez l'icône de filtre. Un sous-menu apparaît avec le nom de colonne et la valeur correspondants et une sélection d'opérateur `equals` ou `not equals`.
2. Sélectionnez la condition appropriée pour le filtre.

Lorsqu'un filtre est appliqué à une table de résultats, l'icône de filtre s'affiche dans la barre d'état. Pour chaque colonne à laquelle est appliqué un filtre, l'icône apparaît également dans l'en-tête de colonne.

Pour afficher toutes les conditions de filtre appliquées à la table, cliquez à l'aide du bouton droit de la souris sur l'en-tête de colonne, puis cliquez sur **Filter Details** (Détails du filtre). La fenêtre Filter Settings (Paramètres du filtre) affiche toutes les conditions de filtre appliquées à la table.

Pour supprimer tous les filtres, cliquez à l'aide du bouton droit de la souris sur l'en-tête de colonne, puis cliquez sur **See All Values** (Afficher toutes les valeurs).

Copie et collage

Vous pouvez utiliser les fonctions **Copier** et **Coller** dans le menu **Editer** sur certains types d'information. Par exemple, vous pouvez copier un outil ObjectServer existant, le coller, puis le modifier pour créer un outil.

Pourquoi et quand exécuter cette tâche

Vous pouvez copier et coller les éléments suivants :

Procédure

- Utilisateurs (dans le même ObjectServer uniquement)
- Filtres de restriction
- Menus et éléments de menu
- Outils

- Invites
- Déclencheurs
- Procédures
- Signaux définis par l'utilisateur
- Couleurs
- Visuels de colonne
- Fichiers ObjectServer
- Bases de données
- Tables
- Processus (dans le cadre d'un contrôle de processus)
- Services (dans le cadre d'un contrôle de processus)

Configuration des couleurs des éléments de syntaxe dans les éditeurs SQL par défaut

Vous pouvez entrer la syntaxe SQL dans les fenêtres de Netcool/OMNIbus Administrator (lors de la création ou de l'édition des déclencheurs, par exemple). Par défaut, Netcool/OMNIbus Administrator propose des couleurs pour mettre en évidence les différents éléments syntaxiques. Vous pouvez modifier le schéma de couleurs de syntaxe par défaut.

Pourquoi et quand exécuter cette tâche

Pour modifier le schéma de couleurs de syntaxe par défaut, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, cliquez sur **Tools** > **Editor Options** (Outils > Options de l'éditeur). La fenêtre Highlight Details (Détails de la mise en évidence) s'ouvre.
2. Complétez cette fenêtre comme suit :

   Elément
   Sélectionnez l'élément de syntaxe dont vous souhaitez afficher ou modifier la couleur.

   Couleur
   Cette zone affiche la couleur actuellement sélectionnée pour l'élément. Cliquez sur le bouton situé en regard de cette zone pour sélectionner une nouvelle couleur. Vous pouvez sélectionner une valeur à l'aide de ses valeurs swatch, HSB, ou RGB.

   Use Default (Utiliser la valeur par défaut)
   Cochez cette case pour utiliser la couleur par défaut pour l'élément sélectionné.
3. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder les détails des couleurs et fermer la fenêtre.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Sélection d'un navigateur Web pour l'affichage de l'aide en ligne

Dans Netcool/OMNIbus Administrator, vous devez sélectionner un navigateur Web dans lequel vous allez afficher l'aide en ligne. L'aide en ligne est déployée à l'aide du système d'aide IBM® Eclipse, sur lequel un certain nombre de navigateurs sont pris en charge.

Pourquoi et quand exécuter cette tâche

Pour obtenir une liste des navigateurs Web pris en charge pour l'aide en ligne, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Pour sélectionnez le navigateur Web et afficher l'aide en ligne, procédez comme suit :

Procédure

1. Dans Netcool/OMNIbus Administrator, sélectionnez **Tools** > **Configure Tools** (Outils > Configurer les outils). La fenêtre Choose Tool (Choisir l'outil) s'ouvre.
2. Sélectionnez **Browser** (Navigateur). La fenêtre External Program (Programme externe) s'ouvre.
3. Renseignez cette fenêtre comme suit :

   Nom d'outil
   Cette zone affiche le type d'outil que vous configurez.

   Executable (Exécutable)
   Entrez le nom et le chemin complet du programme exécutable correspondant au type d'outil. Vous pouvez cliquez sur le bouton situé à droite de la zone pour rechercher et sélectionner le programme exécutable.

   Arguments
   Entrez les arguments de ligne de commande à exécuter avec ce programme exécutable.

   Run time environment (Environnement d'exécution)
   Sélectionnez l'environnement d'exécution.

   Remarque : La zone **Run time environment** (Environnement d'exécution) n'est pas disponible si vous configurez un navigateur Web.
4. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder les détails du programme et fermer la fenêtre.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Sortie de Netcool/OMNIbus Administrator

Pour quitter Netcool/OMNIbus Administrator, cliquez sur **File** > **Exit** (Fichier > Quitter).

Gestion des autorisations avec des utilisateurs, des groupes, des rôles et des filtres de restriction

Une *autorisation* est la vérification des droits d'affichage et de modification des informations.

Pourquoi et quand exécuter cette tâche

L'accès aux objets ObjectServer est contrôlé par des *groupes* (ensembles d'utilisateurs) et des *rôles* (ensembles de droits du système et d'objet) accordés aux groupes. Les *droits* contrôlent l'accès aux objets et aux données du serveur ObjectServer. En combinant un ou plusieurs droits dans des rôles, vous pouvez rapidement et efficacement gérer les accès.

Les administrateurs peuvent autoriser et refuser des actions sur le système et, pour les objets individuels, attribuer des droits aux rôles, et accorder ou révoquer des rôles pour les groupes appropriés d'utilisateurs. Vous pouvez utiliser Netcool/OMNIbus Administrator pour accorder et révoquer des droits pour des utilisateurs d'un système Tivoli Netcool/OMNIbus.

Par exemple, la création d'automatisations implique de connaître les opérations Tivoli Netcool/OMNIbus et la manière dont est configuré un serveur ObjectServer particulier. En général, tous les utilisateurs ne sont pas autorisés à créer ou modifier des automatisations. Une solution consiste à créer un rôle nommé AutoAdmin, détenant des droits de création et de modification des déclencheurs, des groupes de déclencheurs, des fichiers, des procédures SQL, des procédures externes et des signaux. Vous pouvez accorder ce rôle à un groupe d'administrateurs qui va créer ou mettre à jour les déclencheurs.

Pour configurer l'autorisation Tivoli Netcool/OMNIbus, configurez les objets de sécurité dans l'ordre suivant :

Procédure

1. Rôles : attribuez des droits aux rôles.
2. Groupes : attribuez un ou plusieurs rôles à chaque groupe. Les rôles attribués déterminent les actions que les membres du groupe peuvent réaliser sur des objets de base de données.
3. Utilisateurs : ajoutez des utilisateurs aux groupes. Vous devez attribuer chaque utilisateur à un ou plusieurs groupes.

Résultats

Vous pouvez créer des groupements logiques (des utilisateurs root ou des administrateurs système, par exemple), des groupements physiques (les centres d'exploitation des réseau de Paris ou de Bordeaux, par exemple) ou tout autre groupement permettant de simplifier la configuration de la sécurité.

Configuration des rôles

Les rôles sont des collectes de droits d'accès que vous pouvez affecter aux utilisateurs et aux groupes.

Tivoli Netcool/OMNIbus fournit un certain nombre de rôles par défaut. Vous pouvez également créer des rôles personnalisés pour une association avec des utilisateurs et des groupes. Les rôles par défaut sont décrits dans le tableau suivant.

Tableau 18. Rôles par défaut

Nom du rôle	Description
CatalogUser	Ce rôle comprend des droits d'accès permettant de visualiser des informations sur le système, les outils, la sécurité et les tableaux de la base de données du bureau. Ce rôle offre une base pour les droits d'accès de Tivoli Netcool/OMNIbus. Ce rôle ne fournit pas de droits d'accès suffisants pour utiliser des applications Tivoli Netcool/OMNIbus. Attribuez ce rôle à tous les groupes.
AlertsUser	Ce rôle comprend les droits d'accès suivants : • Afficher, mettre à jour et supprimer des entrées dans la table alerts.status • Afficher, insérer et supprimer des entrées dans la table alerts.journal • Afficher et supprimer des entrées dans la table alerts.details Utilisez ce rôle avec le rôle CatalogUser afin d'afficher et de manipuler des alertes, de créer des filtres et des vues et d'exécuter des outils standard dans la liste d'événements.
AlertsProbe	Ce rôle comprend les droits d'accès permettant d'insérer et de mettre à jour des entrées dans la table alerts.status ainsi que d'insérer des entrées dans la table alerts.details. En association avec le rôle CatalogUser, il fournit les droits d'accès nécessaires à une sonde pour générer des alertes sur le serveur ObjectServer. Accordez ces droits d'accès à tout utilisateur exécutant une application de sonde.

Tableau 18. Rôles par défaut (suite)

Nom du rôle	Description
AlertsGateway	Ce rôle comprend des droits d'accès permettant d'insérer, de mettre à jour et de supprimer des entrées dans les tables alerts.status, alerts.details, alerts.journal, alerts.conversions, alerts.col_visuals, alerts.colors ainsi que dans la table des `outils` du bureau et les tables de la base de données de `transfert`. La base de données de `transfert` est utilisée en interne par la passerelle bidirectionnelle du serveur ObjectServer pour synchroniser les informations de sécurité entre les serveurs ObjectServer. Ce rôle inclut également des droits pour sélectionner, insérer, mettre à jour et supprimer des entrées de la table master.servergroups et des droits pour déclencher les signaux : gw_counterpart_down, gw_counterpart_up, gw_resync_start et gw_resync_finish. Ce rôle, en association avec le rôle CatalogUser, fournit les droits d'accès nécessaires à une passerelle pour générer des alertes sur le serveur ObjectServer. Accordez ces droits d'accès à tout utilisateur exécutant une application passerelle.
DatabaseAdmin	Ce rôle comprend des droits d'accès permettant de créer des bases de données et des fichiers ainsi que des tables dans les bases de données `d'alertes`, `d'outils` et `de service`. Il comprend également des droits d'accès permettant de modifier ou de supprimer les tables alerts.status, alerts.details et alerts.journal ainsi que des droits d'accès permettant de créer et de supprimer des index dans les tables alerts.status, alerts.details et alerts.journal. En association avec le rôle CatalogUser, il fournit les droits d'accès permettant de créer des structures de données relationnelles sur le serveur ObjectServer.
AutoAdmin	Ce rôle comprend les droits d'accès permettant de créer des groupes de déclencheurs, des fichiers, des procédures SQL, des procédures externes et des signaux utilisateur. Il comprend également des droits d'accès permettant de créer, de modifier et de supprimer des déclencheurs dans les groupes de déclencheurs par défaut ainsi que de modifier ou de supprimer les groupes de déclencheurs par défaut. En association avec le rôle CatalogUser, il fournit des droits d'accès permettant de créer des automatisations sur le serveur ObjectServer.
ToolsAdmin	Ce rôle comprend des droits d'accès permettant de supprimer, d'insérer et de mettre à jour toutes les tables d'outils. En association avec le rôle CatalogUser, il fournit des droits d'accès permettant de créer et de modifier les outils pouvant être exécutés depuis le bureau et Netcool/OMNIbus Administrator.

Tableau 18. Rôles par défaut (suite)

Nom du rôle	Description
DesktopAdmin	Ce rôle comprend des droits d'accès permettant de mettre à jour tous les catalogues du bureau afin d'insérer, de mettre à jour et de supprimer des couleurs, des vues, des menus, des classes, des résolutions et des conversions. En association avec le rôle CatalogUser, il fournit des droits d'accès permettant de personnaliser le bureau.
SecurityAdmin	En association avec le rôle CatalogUser, ce rôle comprend des droits d'accès permettant de manipuler des utilisateurs, des groupes et des rôles à l'aide de Netcool/OMNIbus Administrator ou de l'interface interactive SQL. Ce rôle comprend également des droits d'accès permettant de définir et de supprimer des connexions utilisateur.
ISQL	En association avec le rôle CatalogUser, ce rôle comprend des droits d'accès permettant de visualiser les données du serveur ObjectServer à l'aide de l'interface interactive SQL.
ISQLWrite	En association avec le rôle CatalogUser, ce rôle comprend des droits d'accès permettant de visualiser et de modifier les données du serveur ObjectServer à l'aide de l'interface interactive SQL.
SuperUser	Ce rôle dispose de tous les droits d'accès disponibles. Vous ne pouvez pas modifier le rôle SuperUser.
Public	Ce rôle est affecté à tous les utilisateurs. Par défaut, aucun droit d'accès n'est affecté au rôle Public. Vous pouvez le modifier mais pas le supprimer.
ChannelAdmin	Ce rôle comprend des droits d'accès permettant de définir des canaux pour une notification d'événement accéléré.
ChannelUser	Ce rôle comprend des droits d'accès permettant de recevoir des notifications d'événements accélérés et d'agir sur celles qui sont diffusées via les canaux.
RegisterProbe	Ce rôle inclut des droits d'accès permettant d'ajouter et de mettre à jour des entrées dans la table registry.probes. Il doit être attribué à tous les comptes utilisateur de sonde.
RegistryReader	Ce rôle inclut des droits d'accès permettant de visualiser des données dans la table registry.probes. Ce rôle n'inclut pas de droit pour modifier des données dans la table registry.probes.
RegistryAdmin	Ce rôle inclut des droits d'accès pour visualiser, modifier, ajouter et supprimer des données dans la table registry.probes. Ce rôle s'adresse uniquement aux administrateurs système et leur permet de résoudre des problèmes inattendus liés à l'enregistrement des sondes.

Les rôles déterminent les types de tâche que les utilisateurs du groupe peuvent effectuer. Par exemple, si vous affectez le rôle AlertsUser à un groupe, les utilisateurs de ce groupe se voient accorder les autorisations suivantes :

- Afficher, mettre à jour et supprimer des entrées dans la table alerts.status
- Afficher, insérer et supprimer des entrées dans la table alerts.journal
- Afficher et supprimer des entrées dans la table alerts.details

Concepts associés:

«Configuration des groupes», à la page 89
Utilisez les groupes pour classer les utilisateurs Tivoli Netcool/OMNIbus en unités disposant d'objectifs fonctionnels communs. Tous les membres d'un groupe disposent des droits d'accès affectés aux rôles du groupe.

«Configuration d'utilisateurs», à la page 94
Vous pouvez créer et modifier des utilisateurs Tivoli Netcool/OMNIbus et organiser ces utilisateurs en groupes. Vous pouvez affecter des rôles aux groupes d'utilisateurs pour contrôler l'accès aux objets du serveur ObjectServer.

Création et édition des rôles

Utilisez les rôles pour attribuer des droits aux utilisateurs appartenant à un groupe particulier.

Avant de commencer

Vous devez avoir démarré Netcool/OMNIbus Administrator en exécutant l'utilitaire **nco_config**. Pour plus d'informations, voir «Démarrage de Netcool/OMNIbus Administrator», à la page 70.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un rôle, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **User** (Utilisateur).
2. Cliquez sur **Roles** (Rôles). Le panneau Roles (Rôles) s'ouvre.
3. Pour ajouter un rôle, cliquez sur **Add Role** (Ajouter un rôle) dans la barre d'outils. La fenêtre Role Details (Détails du rôle) s'ouvre.
4. Pour éditer un rôle, sélectionnez le rôle, puis cliquez sur **Edit Role** (Editer le rôle) dans la barre d'outils. La fenêtre Role Details (Détails du rôle) s'ouvre.
5. Définissez un nouveau rôle comme suit :

   Role Name (Nom du rôle)
   Entrez un nom de rôle unique. Si vous modifiez un rôle, vous ne pouvez pas le renommer.

   Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

   Role ID (ID de rôle)
   Indiquez un numéro d'identification unique pour le rôle. Si vous modifiez un rôle, vous ne pouvez pas changer son ID.
6. Dans l'onglet **Details** (Détails), entrez une description significative du rôle ou mettez à jour la description.
7. Dans l'onglet **Permissions** (Droits), accordez les droits au rôle ou révoquez les droits, comme suit :

Add permission (Ajouter un droit)

Cliquez sur ce bouton pour accorder des droits au rôle. La fenêtre Permission Objects (Droits sur les objets) s'ouvre. Complétez cette fenêtre comme suit :

Type d'objet

Sélectionnez le type d'objet pour lequel vous souhaitez accorder ou révoquer des droits.

Available Objects (Objets disponibles)

Le contenu de cette liste dépend du type d'objet sélectionné. La liste **Available Objects** (Objets disponibles) affiche les objets pour lesquels aucun droit n'a été accordé. Pour accorder des droits sur un ou plusieurs objets, utilisez les flèches de déplacement pour déplacer les objets dans la liste **Selected Objects** (Objets sélectionnés).

Pour déplacer tous les objets dans la liste **Selected Objects** (Objets sélectionnés), cliquez sur >>. Pour déplacer un ou plusieurs objets dans la liste **Selected Objects** (Objets sélectionnés), sélectionnez chaque objet et cliquez sur >. Vous pouvez utiliser la touche majuscule pour effectuer des sélections d'objets consécutives, ou la touche CTRL pour effectuer des sélections non consécutives.

Selected Objects (Objets sélectionnés)

Cette liste affiche les objets pour lesquels les droits ont été accordés. Pour révoquer les droits sur un ou plusieurs objets, utilisez les touches de déplacement et déplacez les objets dans la liste **Available Objects** (Objets disponibles).

Pour déplacer tous les objets dans la liste **Available Objects** (Objets disponibles), cliquez sur <<. Pour déplacer un ou plusieurs objets dans la liste **Available Objects** (Objets disponibles), sélectionnez chaque objet et cliquez sur <. Vous pouvez utiliser la touche majuscule pour sélectionner des objets qui se suivent, ou la touche CTRL pour sélectionner des objets qui ne se suivent pas.

Conseil : Vous pouvez ajouter plusieurs droits en sélectionnant chaque type d'objet requis dans la liste **Type d'objet**, et en ajoutant l'objet à la liste **Selected Objects** (Objets sélectionnés).

OK Cliquez sur ce bouton pour sauvegarder les droits configurés pour le rôle et fermer la fenêtre Permission Objects (Droits sur les objets).

Annuler

Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Lorsque vous revenez sur l'onglet **Permissions** (Droits) de la fenêtre Role Details (Détails de rôle), l'arborescence de droits est mise à jour avec vos modifications. Le noeud parent affiche l'objet principal, les noeuds enfant imbriqués affichent les sous-objets associés. Vous pouvez développer ces sous-objets pour afficher les droits SQL associés ; par exemple, ALTER, DELETE, et DROP. Cochez la case associée au droit SQL pour attribuer ce droit au rôle. Décochez la case pour révoquer le droit.

Delete permission (Supprimer le droit)
Dans l'arborescence des droits, sélectionnez un noeud parent ou enfant dont vous souhaitez révoquer les droits, puis cliquez sur ce bouton. Le noeud est supprimé de l'arborescence.

8. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails de rôle et fermer la fenêtre. Les nouveaux rôles sont ajoutés au panneau Rôles.

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Suppression de rôles

Avant de commencer

Vous devez avoir démarré Netcool/OMNIbus Administrator en exécutant l'utilitaire **nco_config**. Pour plus d'informations, voir «Démarrage de Netcool/OMNIbus Administrator», à la page 70.

Pourquoi et quand exécuter cette tâche

Pour supprimer un rôle, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **User** (Utilisateur).
2. Cliquez sur **Roles** (Rôles). Le panneau Roles (Rôles) s'ouvre.
3. Sélectionnez le rôle à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. Le rôle est supprimé. Si ce rôle était attribué à un groupe, il est supprimé de ce groupe.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Configuration des groupes

Utilisez les groupes pour classer les utilisateurs Tivoli Netcool/OMNIbus en unités disposant d'objectifs fonctionnels communs. Tous les membres d'un groupe disposent des droits d'accès affectés aux rôles du groupe.

Tivoli Netcool/OMNIbus fournit un certain nombre de groupes par défaut. Vous pouvez également créer des groupes personnalisés. Les groupes par défaut sont décrits dans le tableau suivant.

Tableau 19. Groupes par défaut

Nom du groupe	Description
Sonde	Les rôles CatalogUser, AlertsUser, AlertsProbe et RegisterProbe sont affectés à ce groupe.

Tableau 19. Groupes par défaut (suite)

Nom du groupe	Description
Passerelle	Les rôles CatalogUser, AlertsUser, AlertsGateway et RegistryAdmin sont affectés à ce groupe.
ISQL	Le rôle ISQL est affecté à ce groupe.
ISQLWrite	Le rôle ISQLWrite est affecté à ce groupe.
Public	Le rôle Public est affecté à ce groupe. Tous les utilisateurs sont membres de ce groupe.
Normal	Les rôles CatalogUser, AlertsUser, ChannelUser et RegistryReader sont affectés à ce groupe. Ce groupe ne peut pas être supprimé ou renommé.
Administrateur	Les rôles CatalogUser, AlertsUser, ToolsAdmin, DesktopAdmin, ChannelUser, ChannelAdmin, RegistryAdmin et OSLCAdmin sont affectés à ce groupe. Ce groupe ne peut pas être supprimé ou renommé.
Système	Les rôles CatalogUser, AlertsUser, ToolsAdmin, DesktopAdmin, AlertsProbe, AlertsGateway, DatabaseAdmin, AutoAdmin, SecurityAdmin, ISQL, ISQLWrite, SuperUser, ChannelUser, ChannelAdmin, OSLCAdmin et RegistryAdmin sont affectés à ce groupe. Ce groupe ne peut pas être supprimé ou renommé.

Concepts associés:

«Configuration des rôles», à la page 84
Les rôles sont des collectes de droits d'accès que vous pouvez affecter aux utilisateurs et aux groupes.

«Configuration d'utilisateurs», à la page 94
Vous pouvez créer et modifier des utilisateurs Tivoli Netcool/OMNIbus et organiser ces utilisateurs en groupes. Vous pouvez affecter des rôles aux groupes d'utilisateurs pour contrôler l'accès aux objets du serveur ObjectServer.

Création et édition de groupes

Avant de commencer

Vous devez avoir démarré Netcool/OMNIbus Administrator en exécutant l'utilitaire **nco_config**. Pour plus d'informations, voir «Démarrage de Netcool/OMNIbus Administrator», à la page 70.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un groupe, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **User** (Utilisateur).
2. Cliquez sur **Groups** (Groupes). Le panneau Groups (Groupes) s'ouvre.
3. Pour ajouter un groupe, cliquez sur **Add Group** (Ajouter un groupe) dans la barre d'outils. La fenêtre Group Details (Détails du groupe) s'ouvre.
4. Pour éditer un groupe, sélectionnez-le, puis cliquez sur **Edit Group** (Editer le groupe) dans la barre d'outils. La fenêtre Group Details (Détails du groupe) s'ouvre.
5. Définissez le groupe comme suit :

Nom de groupe
Entrez un nom de groupe unique. Si vous modifiez un groupe, vous ne pouvez pas le renommer.

Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

ID Groupe
Entrez un identificateur numérique unique pour le groupe. Si vous modifiez un groupe, vous ne pouvez pas modifiez son ID.

Description
Entrez une description claire du groupe, ou mettez à jour la description.

6. Dans l'onglet **Roles** (Rôles), indiquez les rôles que vous souhaitez attribuer au groupe. Lorsque vous attribuez un rôle à un groupe d'utilisateurs, tout utilisateur ajouté à ce groupe dispose des autorisations de sécurité accordées au rôle. Renseignez l'onglet comme suit :

   Rôles disponibles
   Cette liste affiche les rôles disponibles que vous pouvez attribuer au groupe. Pour attribuer un ou plusieurs de ces rôles, utilisez les touches de déplacement pour faire passer les rôles dans la liste **Applied Roles** (Rôles attribués).

   Pour déplacer tous les rôles dans la liste **Applied Roles** (Rôles attribués), cliquez sur **>>**. Pour déplacer un ou plusieurs rôles dans la liste **Applied Roles** (Rôles attribués), sélectionnez chaque rôle et cliquez sur **>**. Utilisez la touche majuscule pour effectuer des sélections de rôles consécutives, ou la touche CTRL effectuer des sélections non consécutives.

   Applied Roles (Rôles attribués)
   Cette liste affiche les rôles déjà attribués au groupe. Pour désattribuer les rôles, utilisez les touches de déplacement pour déplacer les rôles dans la liste **Rôles disponibles**.

   Pour déplacer tous les rôles dans la liste **Rôles disponibles**, cliquez sur **<<**. Pour déplacer un ou plusieurs rôles dans la liste **Rôles disponibles**, sélectionnez chaque rôle et cliquez sur **<**. Vous pouvez utiliser la touche majuscule pour effectuer des sélection de rôles consécutives, ou la touche CTRL pour effectuer des sélections non consécutives.

   Add new role (Ajouter un nouveau rôle)
   Cliquez sur ce bouton pour créer un rôle pouvant être attribué au groupe. La fenêtre Role Details (Détails du rôle) s'ouvre. Complétez cette fenêtre et enregistrez vos modifications.

   Lorsque vous revenez sur l'onglet **Rôles**, le nouveau rôle est ajouté à la liste **Rôles disponibles**. Vous pouvez alors ajouter ce rôle au groupe.

7. Dans l'onglet **Restriction Filters** (Filtres de restriction), indiquez tous les filtres de restriction que vous souhaitez attribuer au groupe. Vous pouvez utiliser les

filtres de restriction pour empêcher le groupe d'utilisateurs d'afficher ou de modifier certaines lignes des tables du serveur ObjectServer. Renseignez l'onglet comme suit :

All Restriction Filters (Tous les filtres de restriction)
Cette liste affiche les filtres de restriction disponibles qui ont été définis. Pour appliquer un filtre de restriction à un groupe, sélectionnez la ligne du filtre et cliquez sur **Add Restriction Filter** (Ajouter le filtre de restriction). Le filtre sélectionné est transféré dans la liste **Assigned Restriction Filters** (Filtres de restriction attribués).

Si aucun filtre de restriction n'a été défini précédemment, ou si vous souhaitez créer un filtre de restriction et l'appliquer au groupe, cliquez sur **New Restriction Filter** (Nouveau filtre de restriction). La fenêtre Restriction Filter Details (Détails des filtres de restriction) s'ouvre. Complétez cette fenêtre et enregistrez vos modifications.

Lorsque vous revenez sur l'onglet **Restriction Filters** (Filtres de restriction), le nouveau filtre est ajouté à la liste **All Restriction Filters** (Tous les filtres de restriction). Vous pouvez alors appliquer ce filtre de restriction au groupe.

Assigned Restriction Filters (Filtres de restriction attribués)
Cette liste affiche les filtres de restriction attribués au groupe. Pour libérer un filtre de restriction, sélectionnez la ligne du filtre et cliquez sur **Remove Restriction Filter** (Supprimer le filtre de restriction). Le filtre sélectionné est transféré dans la liste **All Restriction Filters** (Tous les filtres de restriction).

8. Dans l'onglet **Users** (Utilisateurs), indiquez les utilisateurs que vous souhaitez ajouter comme membres du groupe. Renseignez l'onglet comme suit :

Non members (Non membres)
Cette liste affiche les utilisateurs que vous pouvez affecter à ce groupe. Pour affecter un ou plusieurs utilisateurs, utilisez les touches de déplacement pour déplacer les utilisateurs dans la liste **Members** (Membres).

Pour déplacer tous les utilisateurs dans la liste **Members** (Membres), cliquez sur **>>**. Pour déplacer un ou plusieurs utilisateurs dans la liste **Members** (Membres), sélectionnez chaque utilisateur et cliquez sur **>**. Utilisez la touche majuscule pour effectuer des sélections d'utilisateurs consécutives, ou la touche CTRL pour effectuer des sélections non consécutives.

Membres
Cette liste affiche les utilisateurs déjà affectés au groupe. Pour libérer les utilisateurs, utilisez les touches de déplacement pour faire passer les utilisateurs dans la liste **Non members** (Non membres).

Pour déplacer tous les utilisateurs dans la liste **Non members** (Non membres), cliquez sur **<<**. Pour déplacer un ou plusieurs utilisateurs dans la liste **Non members** (Non membres), sélectionnez chaque utilisateur et cliquez sur **<**. Utilisez la touche majuscule pour sélectionner des utilisateurs qui se suivent, ou la touche CTRL pour sélectionner des utilisateurs qui ne se suivent pas.

Add new user (Ajouter un nouvel utilisateur)
Cliquez sur ce bouton pour créer un utilisateur pouvant être affecté à ce groupe. La fenêtre User Details (Détails de l'utilisateur) s'ouvre. Complétez cette fenêtre et sauvegardez vos modifications.

Lorsque vous revenez sur l'onglet **Utilisateurs**, le nouvel utilisateur est ajouté à la liste **Non members** (Non membres). Vous pouvez désormais ajouter cet utilisateur au groupe.

9. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder les détails du groupe et fermer la fenêtre. Les nouveaux groupes sont ajoutés au panneau Groupes.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Création et édition des rôles», à la page 87
Utilisez les rôles pour attribuer des droits aux utilisateurs appartenant à un groupe particulier.

«Création et édition de filtres de restriction», à la page 100

«Création et édition d'utilisateurs», à la page 94
Lors de la configuration des utilisateurs, vous devez les attribuer à des groupes avec des rôles alloués. Cela permet de déterminer les droits utilisateur.

Suppression de groupes

Vous ne pouvez pas supprimer les groupes Normal, Administrateur et Système.

Avant de commencer

Vous devez avoir démarré Netcool/OMNIbus Administrator en exécutant l'utilitaire **nco_config**. Pour plus d'informations, voir «Démarrage de Netcool/OMNIbus Administrator», à la page 70.

Pourquoi et quand exécuter cette tâche

Pour supprimer un groupe, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **User** (Utilisateur).
2. Cliquez sur **Groups** (Groupes). Le panneau Groups (Groupes) s'ouvre.
3. Sélectionnez le groupe à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. Le groupe est supprimé.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Configuration d'utilisateurs

Vous pouvez créer et modifier des utilisateurs Tivoli Netcool/OMNIbus et organiser ces utilisateurs en groupes. Vous pouvez affecter des rôles aux groupes d'utilisateurs pour contrôler l'accès aux objets du serveur ObjectServer.

Tivoli Netcool/OMNIbus fournit un ensemble de comptes utilisateur par défaut. Les utilisateurs par défaut sont décrits dans le tableau suivant.

Tableau 20. Utilisateurs par défaut

Nom d'utilisateur	Description
root	Cet utilisateur est créé avec une chaîne vide comme mot de passe par défaut. Vous pouvez redéfinir le mot de passe à l'aide de Netcool/OMNIbus Administrator ou de la commande SQL du serveur ObjectServer ALTER USER.
nobody	Cet utilisateur est désactivé et ne peut pas être utilisé pour accéder au serveur ObjectServer. La propriété de chaque alerte dans la table alerts.status est affectée à un utilisateur lorsque la ligne est insérée. Par défaut, les sondes affectent des lignes à l'utilisateur `nobody`.

Concepts associés:

«Configuration des groupes», à la page 89
Utilisez les groupes pour classer les utilisateurs Tivoli Netcool/OMNIbus en unités disposant d'objectifs fonctionnels communs. Tous les membres d'un groupe disposent des droits d'accès affectés aux rôles du groupe.

«Configuration des rôles», à la page 84
Les rôles sont des collectes de droits d'accès que vous pouvez affecter aux utilisateurs et aux groupes.

Création et édition d'utilisateurs

Lors de la configuration des utilisateurs, vous devez les attribuer à des groupes avec des rôles alloués. Cela permet de déterminer les droits utilisateur.

Avant de commencer

Vous devez avoir démarré Netcool/OMNIbus Administrator en exécutant l'utilitaire **`nco_config`**. Pour plus d'informations, voir «Démarrage de Netcool/OMNIbus Administrator», à la page 70.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un utilisateur, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **User** (Utilisateur).
2. Cliquez sur **Users** (Utilisateurs). Le panneau Users (Utilisateurs) s'ouvre.
3. Pour ajouter un utilisateur, cliquez sur **Add User** (Ajouter un utilisateur) dans la barre d'outils. La fenêtre User Details (Détails de l'utilisateur) s'ouvre.
4. Pour éditer un utilisateur, sélectionnez-le, puis cliquez sur **Edit User** (Editer l'utilisateur) dans la barre d'outils. La fenêtre User Details (Détails de l'utilisateur) s'ouvre.
5. Définissez l'utilisateur comme suit :

Nom d'utilisateur
Entrez un nom d'utilisateur unique. Si l'utilisateur doit être authentifié en externe, par exemple, dans un référentiel LDAP (Lightweight Directory Access Protocol) ou via l'utilisation de modules PAM (Pluggable Authentication Modules), le nom entré dans cette zone doit être identique au nom stocké dans le référentiel d'authentification externe. Si vous modifiez un utilisateur, vous ne pouvez pas le renommer.

Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

ID utilisateur
Indiquez un identificateur numérique unique pour l'utilisateur. Lorsque cela est possible, il doit correspondre au numéro d'utilisateur UNIX. Si vous modifiez un utilisateur, vous ne pouvez pas changer son ID.

L'identificateur de l'utilisateur `root` est `0`. L'identificateur de l'utilisateur `nobody` est `65534`. Les identificateurs des autres utilisateurs doivent être définis sur une valeur située entre `1` et `2147483647`.

Nom complet
Entrez le nom complet de l'utilisateur.

Create Conversion (Créer une conversion)
Cochez cette case si vous souhaitez créer une conversion pour l'utilisateur. Le nom d'utilisateur ou son nom complet s'affiche alors dans les listes d'événements Tivoli Netcool/OMNIbus. Le nom affiché est déterminé par la valeur de la propriété **`system.conversion.type`** dans le fichier de propriétés Netcool/OMNIbus Administrator `$NCHOME/omnibus/etc/nco_config.props`. Cette propriété crée une conversion entre l'ID utilisateur et le nom ou nom complet de chaque utilisateur créé.

6. Dans l'onglet **Groups** (Groupes), indiquez les groupes auxquels va appartenir l'utilisateur. Renseignez l'onglet comme suit :

Unassigned Groups (Groupes non affectés)
Cette liste affiche les groupes disponibles auxquels vous pouvez affecter l'utilisateur. Pour affecter l'utilisateur à un ou plusieurs de ces groupes, utilisez les touches de déplacement pour faire passer les groupes dans la liste **Assigned Groups** (Groupes affectés).

Pour déplacer tous les groupes dans la liste **Assigned Groups** (Groupes affectés), cliquez sur **>>**. Pour déplacer un ou plusieurs groupes dans la liste **Assigned Groups** (Groupes affectés), sélectionnez chaque groupe et cliquez sur **>**. Vous pouvez utiliser la touche majuscule pour effectuer une sélection de groupes consécutive, ou la touche CTRL pour effectuer une sélection non consécutive.

Remarque : Si vous n'affectez pas l'utilisateur à un groupe, il ne disposera d'aucun droit.

Assigned Groups (Groupes affectés)
Cette liste affiche les groupes auxquels l'utilisateur est affecté. Pour

annuler l'affectation à un groupe, utilisez les touches de déplacement pour déplacer le groupe dans la liste **Unassigned Groups** (Groupes non affectés).

Pour déplacer tous les groupes dans la liste **Unassigned Groups** (Groupes non affectés) cliquez sur **<<**. Pour déplacer un ou plusieurs groupes dans la liste **Unassigned Groups** (Groupes non affectés), sélectionnez chaque groupe et cliquez sur **<**. Vous pouvez utiliser la touche majuscule pour effectuer des sélections de groupes consécutives, ou la touche CTRL pour effectuer des sélections non consécutives.

Ajouter un nouveau groupe
Cliquez sur ce bouton si vous souhaitez créer un nouveau groupe auquel l'utilisateur peut être affecté. La fenêtre Group Details (Détails du groupe) s'ouvre. Complétez cette fenêtre et sauvegardez vos modifications.

Lorsque vous revenez sur l'onglet **Groupes** dans la fenêtre User Details (Détails de l'utilisateur), le nouveau groupe sera ajouté à la liste **Unassigned Groups** (Groupes non affectés). Vous pouvez alors affecter l'utilisateur à ce groupe.

7. Dans l'onglet **Restriction Filters** (Filtres de restriction), indiquez tous les filtres de restriction que vous souhaitez attribuer à l'utilisateur. Vous pouvez utiliser les filtres de restriction pour empêcher l'utilisateur d'afficher ou de modifier certaines lignes des tables du serveur ObjectServer. Renseignez l'onglet comme suit :

   All Restriction Filters (Tous les filtres de restrictions)
   Cette liste affiche les filtres de restriction qui ont été configurés. Pour appliquer un filtre de restriction à un utilisateur, sélectionnez la ligne du filtre de restriction et cliquez sur **Add Restriction Filter** (Ajouter un filtre de restriction). Le filtre de restriction sélectionné est ajouté à la liste **Assigned Restriction Filters** (Filtres de restriction affectés).

   Si aucun filtre de restriction n'a été précédemment défini, ou si vous voulez créer un nouveau filtre de restriction et l'appliquer à l'utilisateur, cliquez sur **New Restriction Filter** (Nouveau filtre de restriction). La fenêtre Restriction Filter Details (Détails de filtre de restriction) s'ouvre. Complétez cette fenêtre et sauvegarder les modifications.

   Lorsque vous revenez sur l'onglet **Restriction Filters** (Filtres de restriction), le nouveau filtre de restriction est ajouté à la liste **All Restriction Filters** (Tous les filtres de restriction). Vous pouvez alors appliquer le filtre de restriction à l'utilisateur.

   Assigned Restriction Filters (Filtres de restriction affectés)
   Cette liste affiche les filtres de restriction affectés à l'utilisateur. Pour annuler l'affectation d'un filtre, sélectionnez la ligne du filtre et cliquez sur **Remove Restriction Filter** (Supprimer le filtre de restriction). Le filtre de restriction sélectionné est déplacé dans la liste **All Restriction Filters** (Tous les filtres de restriction).

8. Dans l'onglet **Settings** (Paramètres), indiquez les détails d'authentification de l'utilisateur, puis activez l'utilisateur sur le système.

   Mot de passe
   Entrez un mot de passe facultatif pour l'utilisateur. Des astérisques (*) s'affichent à la place des caractères que vous saisissez.

   Si l'utilisateur doit être authentifié en externe, le mot de passe est stocké dans le référentiel externe ; laissez cette zone vide.

Verify (Confirmer)
Si vous avez saisi le mot de passe dans la zone **Mot de passe**, entrez à nouveau le mot de passe ici.

Modification
Cliquez sur ce bouton pour redéfinir le mot de passe utilisateur. Vous devez ensuite le saisir à nouveau dans les zones **Mot de passe** et **Verify** (Confirmer).

Ce bouton est visible uniquement lorsque vous modifiez les détails d'un utilisateur.

External Authentication (Authentification externe)
Cochez cette case pour authentifier l'utilisateur en externe. Décochez cette case pour stocker le nom d'utilisateur et son mot de passe dans le serveur ObjectServer, et pour effectuer une authentification ObjectServer.

User Type (Type d'utilisateur)
Cette zone en lecture seule indique le type de l'utilisateur. Elle est automatiquement renseignée en fonction des groupes d'appartenance de l'utilisateur.

User Enabled (Utilisateur activé)
Cochez cette case pour activer le compte utilisateur et lui permettre de se connecter à Tivoli Netcool/OMNIbus. Les nouveaux comptes d'utilisateurs sont activés par défaut.

Décochez cette case pour créer un compte d'utilisateur sans l'activer. Cela peut s'avérer nécessaire lorsque vous souhaitez refuser l'accès à un utilisateur tant que tous les droits appropriés ne lui sont pas attribués.

9. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails de l'utilisateur et fermer la fenêtre. Les nouveaux utilisateurs sont ajoutés au panneau Utilisateurs.

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Création et édition de groupes», à la page 90

«Création et édition de filtres de restriction», à la page 100

Suppression des utilisateurs

Avant de commencer

Vous devez avoir démarré Netcool/OMNIbus Administrator en exécutant l'utilitaire **nco_config**. Pour plus d'informations, voir «Démarrage de Netcool/OMNIbus Administrator», à la page 70.

Pourquoi et quand exécuter cette tâche

Pour supprimer un utilisateur, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **User** (Utilisateur).
2. Cliquez sur **Users** (Utilisateurs). Le panneau Users (Utilisateurs) s'ouvre.
3. Sélectionnez l'utilisateur à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. L'utilisateur est supprimé.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Affichage des connexions utilisateur au serveur ObjectServer

Vous pouvez afficher les détails de la connexion d'un utilisateur connecté au serveur ObjectServer.

Avant de commencer

Vous devez avoir démarré Netcool/OMNIbus Administrator en exécutant l'utilitaire **nco_config**. Pour plus d'informations, voir «Démarrage de Netcool/OMNIbus Administrator», à la page 70.

Pourquoi et quand exécuter cette tâche

Pour afficher les connexions utilisateur, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **User** (Utilisateur).
2. Cliquez sur **Users** (Utilisateurs). Le panneau Users (Utilisateurs) s'ouvre et affiche une ligne pour chaque utilisateur configuré sur le système.

   Une icône d'indicateur de statut est affichée à gauche du nom d'utilisateur. La couleur de l'icône indique le statut de connexion de l'utilisateur, comme suit :

   - La couleur verte indique que l'utilisateur est connecté.
   - La couleur bleue indique que l'utilisateur est activé, mais pas connecté.
   - La couleur grise indique que l'utilisateur n'est pas activé.

   Dans le panneau Utilisateurs, vous pouvez afficher les détails de connexion de tout utilisateur sélectionné connecté au serveur ObjectServer. Par exemple, si vous souhaitez afficher le nombre d'utilisateurs connectés en tant qu'utilisateur `root`, avec leurs détails de connexion, sélectionnez la ligne de l'utilisateur `root`. Cliquez avec le bouton droit de la souris sur la ligne sélectionnée, puis cliquez

sur **See connections** (Afficher les connexions) dans le menu contextuel. Le panneau ObjectServer Connections (Connexions ObjectServer) s'ouvre. Un filtre rapide dynamique est automatiquement appliqué à la colonne Login Name pour que seules les applications auxquelles l'utilisateur sélectionné est connecté soient affichées. L'icône **Filtre rapide** s'affiche dans l'en-tête de colonne Login Name pour indiquer que le filtre a été appliqué à ce nom. Vous pouvez supprimer ce filtre pour afficher toutes les connexions de tous les utilisateurs en cliquant sur **Connexions** dans le bouton de menu **Système**.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Surveillance des connexions au serveur ObjectServer», à la page 162
Vous pouvez afficher toutes les connexions en cours au serveur ObjectServer et en déconnecter une ou plusieurs. Vous devez détenir le droit ALTER SYSTEM DROP CONNECTION pour pouvoir arrêter les connexions au serveur ObjectServer.

Configuration des filtres de restriction

Vous pouvez utiliser les filtres de restriction pour contrôler les lignes des tables ObjectServer que les utilisateurs sont autorisés à consulter et à modifier. Les filtres de restriction s'appliquent aux applications client Tivoli Netcool/OMNIbus et aux instructions INSERT, UPDATE et DELETE. Vous pouvez affecter des filtres de restriction aux utilisateurs individuels ou à une collection d'utilisateurs dans un groupe.

Vous ne pouvez affecter à un utilisateur ou à un groupe qu'un filtre de restriction par table. Cependant, plusieurs filtres de restriction peuvent s'appliquer à un utilisateur. Par exemple, un utilisateur peut être associé à un filtre de restriction unique et peut également être membre d'un ou de plusieurs groupes, auxquels sont appliqués des filtres de restriction. Ainsi, si un utilisateur est membre de trois groupes auxquels des filtres de restriction sont affectés et qu'il se voit également attribuer un filtre de restriction, quatre filtres de restriction s'appliquent à cet utilisateur. Si plusieurs filtres de restriction s'appliquent à un utilisateur, les données obtenues sont une combinaison de tous les filtres de restriction applicables pour l'utilisateur ou le groupe.

Si vous utilisez plusieurs filtres de restriction, vérifiez que vous avez paramétré correctement la propriété de l'ObjectServer **RestrictionFiltersAND**.

Par exemple, vous pouvez créer un filtre de restriction sur la table alerts.status qui possède la condition suivante :

```
Tally > 100 AND Severity > 4
```

Une fois ce filtre de restriction appliqué à un utilisateur, ce dernier ne peut visualiser que les événements qui se sont produits plus de 100 fois et qui possèdent un niveau de gravité supérieur à 4.

On peut aussi, par exemple, utiliser des filtres de restriction pour limiter la couverture géographique des événements que les utilisateurs peuvent visualiser. Vous pouvez créer un filtre de restriction sur la table alerts.status qui autorise les utilisateurs basés à Londres à ne voir que les événements en provenance des centres d'opérations de Londres.

Création et édition de filtres de restriction

Avant de commencer

Vous devez avoir démarré Netcool/OMNIbus Administrator en exécutant l'utilitaire **nco_config**. Pour plus d'informations, voir «Démarrage de Netcool/OMNIbus Administrator», à la page 70.

Procédure

Pour créer ou éditer un filtre de restriction, procédez comme suit :

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **User** (Utilisateur).
2. Cliquez sur **Restriction Filters** (Filtres de restriction). Le panneau Restriction Filters (Filtres de restriction) s'ouvre.
3. Pour ajouter ou éditer un filtre de restriction, procédez comme suit :
   - Pour ajouter un filtre de restriction, cliquez sur **Add Restriction Filter** (Ajouter un filtre de restriction) dans la barre d'outils.
   - Pour éditer un filtre de restriction, sélectionnez-le, puis cliquez sur **Edit Restriction Filter** (Editer le filtre de restriction) dans la barre d'outils.
4. Renseignez la fenêtre Restriction Filter Details (Détails du filtre de restriction) comme suit :

   Name Entrez un nom de filtre de restriction unique.

   Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

   Database
   Sélectionnez la base de données pour laquelle vous créez le filtre de restriction.

   Table Sélectionnez la table pour laquelle vous créez le filtre de restriction.

   Condition
   Entrez la condition SQL (instruction WHERE) pour le filtre de restriction. Par exemple :

   ```
   Tally > 100 AND Severity >=4
   ```

   Si cette condition est appliquée à un utilisateur ou un groupe, l'utilisateur ou les membres du groupe ne sont autorisés à voir les alertes dans la liste des événements uniquement que si elles sont générées plus de 100 fois et qu'elles ont un niveau de gravité supérieur ou égal à 4. Vous pouvez utiliser les boutons de l'assistant SQL pour créer la condition de filtre, comme décrit dans le tableau ci-dessous.

Tableau 21. Boutons de création des conditions de filtre

Bouton	Description
Columns	Cliquez sur ce bouton pour sélectionner un nom de colonne de table à ajouter à la commande. Le nom de colonne est remplacé par la valeur de la ligne correspondante de la liste d'événements lors de l'exécution de l'outil.
Conversions	Cliquez sur ce bouton pour procéder à une sélection dans une liste de conversions disponibles.
Complete	Cliquez sur ce bouton pour concevoir une liste de mots clés qui complète l'instruction SQL entrée. **Conseil :** Vous pouvez également saisir un ou plusieurs caractères et appuyer sur Ctrl+F1 pour générer une boîte de dialogue avec une liste de mots clés susceptibles de correspondre à votre saisie. Sélectionnez le mot clé requis et cliquez sur **OK** pour saisir l'entrée. Si un seul mot clé correspond aux caractères saisis, le mot clé est automatiquement sélectionné. Si vous appuyez sur Ctrl+F1 après avoir saisi un mot clé associé à une base de données, la boîte de dialogue fournit la liste des bases de données ObjectServer que vous pouvez sélectionner. Si vous appuyez sur Ctrl+F1 après avoir saisi le nom de la base de données suivi d'un point (par exemple : `alerts.`), vous pouvez appuyer à nouveau sur Ctrl+F1 pour afficher et sélectionner dans une liste de tables de la base de données.
Check SQL	Cliquez sur ce bouton pour vérifier la validité de la syntaxe SQL entrée.

OK Cliquez sur ce bouton pour sauvegarder les détails de filtre de restriction et fermer la fenêtre. Les nouveaux filtres de restriction sont ajoutés au panneau Restriction Filters (Filtres de restriction).

Cancel
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **`nco_config`** pour démarrer Netcool/OMNIbus Administrator.

«Création et édition des conversions», à la page 143
Les conversions sont associées aux colonnes de la table alerts.status du serveur ObjectServer et mettent en correspondance les valeurs entières des colonnes avec les chaînes. Les conversions configurées dans Netcool/OMNIbus Administrator sont utilisées dans la liste d'événements afin de convertir les valeurs entières en chaîne, pour des raisons de lisibilité.

Suppression de filtres de restriction

Vous ne pouvez pas supprimer les filtres de restriction attribués à un utilisateur ou un groupe.

Avant de commencer

Vous devez avoir démarré Netcool/OMNIbus Administrator en exécutant l'utilitaire **nco_config**. Pour plus d'informations, voir «Démarrage de Netcool/OMNIbus Administrator», à la page 70.

Pourquoi et quand exécuter cette tâche

Pour supprimer un filtre de restriction, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **User** (Utilisateur).
2. Cliquez sur **Restriction Filters** (Filtres de restriction). Le panneau Restriction Filters (Filtres de restriction) s'ouvre.
3. Sélectionnez le filtre de restriction à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. Le filtre de restriction est supprimé.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Configuration des menus, outils et invites

Vous pouvez configurer les menus, outils et invites du bureau Tivoli Netcool/OMNIbus (liste d'événements et conducteur).

Pourquoi et quand exécuter cette tâche

Chaque menu est composé d'un nom et d'une liste d'éléments de menu. Les éléments de menu sont des outils, des séparateurs et des sous-menus. Les sous-menus sont utilisés dans les menus pour grouper au moins deux éléments.

Les outils permettent de contrôler les fonctions de gestion des alertes dans Tivoli Netcool/OMNIbus. Chaque outil est associé à une instruction SQL (appelée *effet interne*) et/ou un exécutable (appelé *effet externe*). Vous pouvez associer les outils à une classe d'alerte, et les grouper en menus d'outils.

Un outil peut contenir une fenêtre d'invite ou un menu contextuel pour permettre à l'utilisateur d'entrer des informations.

Personnalisation des menus

Utilisez Netcool/OMNIbus Administrator pour personnaliser les menus du bureau Tivoli Netcool/OMNIbus pour la liste d'événements et Conductor.

Pourquoi et quand exécuter cette tâche

Vous pouvez personnaliser un menu de l'une des manières suivantes :

- Ajoutez des outils, des sous-menus et des séparateurs à un menu
- Editez des éléments de menu
- Modifiez l'ordre des éléments de menu
- Supprimez des outils, des sous-menus ou de séparateurs d'un menu

Après avoir personnalisé un menu, vous pouvez afficher un aperçu de sa structure pour savoir comment elle va apparaître dans la liste d'événements ou Conductor.

La table suivante affiche les menus que vous pouvez personnaliser.

Tableau 22. Menus de listes d'événements personnalisables

Nom de menu dans Netcool/OMNIbus Administrator - fenêtre Menus	**Nom de menu dans le bureau Tivoli Netcool/OMNIbus**
AlertsMenu	menu **Alertes** et menu contextuel dans la liste d'événements, lorsqu'une alerte est sélectionnée
MainEventListMenu	Menu **Outils** dans la fenêtre d'écrans de surveillance Liste d'événements
SubEventListMenu	Menu **Outils** pour toute fenêtre de liste d'événements
Menu Conductor	Menu **Outils** dans Conductor

Ajouts d'outils, de sous-menus et de séparateurs à un menu

Vous pouvez ajouter des outils, de sous-menus et de séparateurs comme éléments de menu dans un menu de bureau.

Pourquoi et quand exécuter cette tâche

Pour ajouter un outil, un sous-menu ou un séparateur à un menu, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton **Menu**.
2. Cliquez sur **Menus**. Le panneau Menus s'ouvre.

   Conseil : Vous pouvez cliquer deux fois sur le menu pour afficher les éléments de menu et les outils existants. Vous pouvez également cliquer sur le symbole circulaire (sous UNIX) ou le symbole plus (+) (sous Windows) à gauche du menu.
3. Dans la structure arborescente, sélectionnez le menu auquel vous souhaitez ajouter des éléments, puis cliquez sur **Add Item** (Ajouter l'élément) dans la barre d'outils. La fenêtre Menu Item Details (Détails de l'élément de menu) s'ouvre.
4. Complétez cette fenêtre comme suit :

Menu Item Type (Type d'élément de menu)
Si vous ajoutez un élément de menu, sélectionnez le type d'élément de menu à ajouter au menu sélectionné. Si vous modifiez un outil ou un sous-menu, cette zone est en lecture seule.

Remarque : Si vous ajoutez un séparateur, il s'agit de la seule entrée requise dans cette fenêtre.

Outil Sélectionnez l'outil à ajouter. Si vous modifiez un outil, vous pouvez sélectionner un autre outil dans cette liste déroulante.

Remarque : Cette zone est disponible pour les outils uniquement.

Si vous souhaitez créer un nouvel outil qui peut ensuite être ajouté en tant qu'élément du menu sélectionné, cliquez sur **Add Tool** (Ajouter un outil). La fenêtre Tool Details (Détails de l'outil) s'ouvre. Complétez cette fenêtre et enregistrez vos modifications. Lorsque vous revenez sur la fenêtre Menu Item Details (Détails des éléments de menu), l'outil pourra être sélectionné dans la liste déroulante **Tool** (Outil).

Si vous souhaitez modifier les paramètres de l'outil sélectionné, cliquez sur **Edit Tool** (Modifier l'outil). La fenêtre Tool Details (Détails de l'outil) s'ouvre. Complétez cette fenêtre et enregistrez vos modifications. Vous revenez alors sur la fenêtre Menu Item Details (Détails des éléments de menu).

Titre Si vous ajoutez ou modifiez un outil ou un sous-menu, acceptez le texte par défaut de cette zone (s'il apparaît), ou entrez un autre titre. Insérez une perluète (&) dans le titre pour créer un mnémonique. Ce texte s'affiche dans le menu comme nom d'élément de menu.

Remarque : Si vous ajoutez un outil, la zone **Titre** est définie par défaut sur le nom de l'outil sélectionné. Cependant, si vous sélectionnez ensuite un outil différent, l'entrée initiale de la zone **Titre** reste inchangée, sauf si vous la mettez à jour manuellement.

5. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails d'éléments de menu et fermer la fenêtre. Le panneau Menus est mis à jour et affiche les modifications effectuées. Les modifications apparaissent également dans la liste d'événements ou le menu Conductor pertinents au prochain démarrage du bureau ou à la prochaine synchronisation de la liste d'événement avec le serveur ObjectServer.

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Que faire ensuite

Conseil : Si vous avez ajouté un sous-menu au menu sélectionné, vous devez à présent ajouter des éléments de menu. Sélectionnez le sous-menu dans le panneau Menus, puis suivez les étapes 3, à la page 103, 4, à la page 103 et 5 pour chaque élément de menu à ajouter.

Tâches associées:

«Création et édition d'outils», à la page 108
Lors de sa création, un outil est ajouté à la base de données d'outils. Les outils qui s'affichent dans le panneau Tools (Outils) sont des liens vers des entrées de cette

base de données.

Edition d'éléments de menu

Vous pouvez modifier le titre des outils et des sous-menus de la liste d'événements et des menus Conductor. Vous pouvez également remplacer un outil par un autre et éditer sa définition.

Pourquoi et quand exécuter cette tâche

Pour modifier un élément de menu, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton **Menu**.
2. Cliquez sur **Menus**. Le panneau Menus s'ouvre.
3. Dans la structure arborescente, sélectionnez l'outil ou le sous-menu que vous souhaitez éditer, puis cliquez sur **Edit Item** (Editer l'élément) dans la barre d'outils. La fenêtre Menu Item Details (Détails de l'élément de menu) s'ouvre.
4. Editez le menu comme suit :

   Menu Item Type (Type d'élément de menu)
   Si vous ajoutez un élément de menu, sélectionnez le type d'élément de menu à ajouter au menu sélectionné. Si vous modifiez un outil ou un sous-menu, cette zone est en lecture seule.

   Remarque : Si vous ajoutez un séparateur, il s'agit de la seule entrée requise dans cette fenêtre.

   Outil Sélectionnez l'outil à ajouter. Si vous modifiez un outil, vous pouvez sélectionner un autre outil dans cette liste déroulante.

   Remarque : Cette zone est disponible pour les outils uniquement.

   Si vous souhaitez créer un nouvel outil qui peut ensuite être ajouté en tant qu'élément du menu sélectionné, cliquez sur **Add Tool** (Ajouter un outil). La fenêtre Tool Details (Détails de l'outil) s'ouvre. Complétez cette fenêtre et enregistrez vos modifications. Lorsque vous revenez sur la fenêtre Menu Item Details (Détails des éléments de menu), l'outil pourra être sélectionné dans la liste déroulante **Tool** (Outil).

   Si vous souhaitez modifier les paramètres de l'outil sélectionné, cliquez sur **Edit Tool** (Modifier l'outil). La fenêtre Tool Details (Détails de l'outil) s'ouvre. Complétez cette fenêtre et enregistrez vos modifications. Vous revenez alors sur la fenêtre Menu Item Details (Détails des éléments de menu).

   Titre Si vous ajoutez ou modifiez un outil ou un sous-menu, acceptez le texte par défaut de cette zone (s'il apparaît), ou entrez un autre titre. Insérez une perluète (&) dans le titre pour créer un mnémonique. Ce texte s'affiche dans le menu comme nom d'élément de menu.

   Remarque : Si vous ajoutez un outil, la zone **Titre** est définie par défaut sur le nom de l'outil sélectionné. Cependant, si vous sélectionnez ensuite un outil différent, l'entrée initiale de la zone **Titre** reste inchangée, sauf si vous la mettez à jour manuellement.
5. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder les détails d'éléments de menu et fermer la fenêtre. Le panneau Menus est mis à jour et affiche les

modifications effectuées. Les modifications apparaissent également dans la liste d'événements ou le menu Conductor pertinents au prochain démarrage du bureau ou à la prochaine synchronisation de la liste d'événement avec le serveur ObjectServer.

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Création et édition d'outils», à la page 108
Lors de sa création, un outil est ajouté à la base de données d'outils. Les outils qui s'affichent dans le panneau Tools (Outils) sont des liens vers des entrées de cette base de données.

Modification de l'ordre des éléments de menu

Vous pouvez réorganiser les éléments d'un menu afin d'améliorer les flux de travaux de l'utilisateur.

Pourquoi et quand exécuter cette tâche

Pour modifier l'ordre des éléments de menu du bureau, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton **Menu**.
2. Cliquez sur **Menus**. Le panneau Menus s'ouvre.
3. Sélectionnez l'élément de menu que vous souhaitez repositionner et déplacez-le vers son nouvel emplacement en procédant de l'une des façons suivantes :
   - Pour déplacer un élément vers le haut du menu, sélectionnez **Move To Top** (Déplacer vers le haut) dans le menu **Item** (Elément), la barre d'outils ou le menu contextuel qui s'ouvre lorsque vous cliquez sur l'élément à l'aide du bouton droit de la souris.
   - Pour déplacer l'élément d'une position vers le haut, sélectionnez **Move Up** (Vers le haut) dans le menu **Item** (Elément), la barre d'outils ou le menu contextuel qui s'ouvre lorsque vous cliquez sur l'élément à l'aide du bouton droit de la souris.
   - Pour déplacer l'élément d'une position vers le bas, sélectionnez **Move Down** (Vers le bas) dans le menu **Item** (Elément), la barre d'outils ou le menu contextuel qui s'ouvre lorsque vous cliquez sur l'élément à l'aide du bouton droit de la souris.
   - Pour déplacer un élément vers le bas du menu, sélectionnez **Move To Bottom** (Déplacer vers le bas) dans le menu **Item** (Elément), la barre d'outils ou le menu contextuel qui s'ouvre lorsque vous cliquez sur l'élément à l'aide du bouton droit de la souris.

   Les éléments de menu repositionnés s'affichent dans le nouvel ordre au prochain démarrage du bureau ou de la prochaine resynchronisation de la liste d'événements avec le serveur ObjectServer.

Suppression d'outils, de sous-menus ou de séparateurs d'un menu

Pourquoi et quand exécuter cette tâche

Pour supprimer un élément de menu du bureau, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton **Menu**.
2. Cliquez sur **Menus**. Le panneau Menus s'ouvre.
3. Sélectionnez l'élément de menu que vous souhaitez supprimer. Pour les sous-menus, vous pouvez sélectionner un seul élément à supprimer ou sélectionner le nom du sous-menu à supprimer.
4. Dans la barre d'outils, cliquez sur **Delete** (Supprimer).

Résultats

L'élément de menu est supprimé du bureau au démarrage suivant du bureau ou lors de la resynchronisation avec le serveur ObjectServer.

Prévisualisation de la structure des menus personnalisés

Vous pouvez prévisualiser les menus personnalisés pour savoir comment ils apparaissent dans le bureau.

Pourquoi et quand exécuter cette tâche

Pour prévisualiser les menus, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton **Menu**.
2. Cliquez sur **Menus**. Le panneau Menus s'ouvre.
3. Dans la barre d'outils, cliquez sur **Show Menu Structure** (Afficher la structure de menus). La fenêtre d'aperçu s'ouvre avec les menus affichés horizontalement. Vous pouvez cliquer sur chaque nom de menu pour développer son contenu.

Résultats

Remarque : Les noms de menu tels qu'ils apparaissent dans Netcool/OMNIbus Administrator sont différents de ceux présents sur le bureau.

Configuration des outils

Les outils permettent aux opérateurs de gérer des alertes dans la liste d'événements. Vous pouvez créer des outils que les opérateurs peuvent utiliser pour exécuter des commandes SQL sur le serveur ObjectServer ou exécuter des commandes externes qui démarrent une application, un fichier de traitement par lots ou un script local.

Un outil peut posséder une instruction et/ou des commandes exécutables associée(s). Un outil peut également inclure une fenêtre d'invite ou un menu contextuel permettant aux opérateurs d'entrer ou de sélectionner des informations.

Vous pouvez ajouter des outils aux menus de la liste d'événements et associer des outils aux classes d'alerte. Ces outils sont disponibles dans les menus uniquement lorsque les alertes des classes associées sont sélectionnées dans la liste d'événements.

Remarque : Vous pouvez exécuter l'utilitaire **nco_elct** dans un outil pour ouvrir une liste d'événements personnalisée et transitoire. Par exemple, vous pouvez ouvrir une liste d'événements et appliquer un filtre pour afficher toutes les alertes critiques d'un serveur ObjectServer particulier. Pour plus d'informations sur **nco_elct**, voir *IBM Tivoli Netcool/OMNIbus User's Guide*.

Tâches associées:

«Ajouts d'outils, de sous-menus et de séparateurs à un menu», à la page 103
Vous pouvez ajouter des outils, de sous-menus et de séparateurs comme éléments de menu dans un menu de bureau.

Création et édition d'outils

Lors de sa création, un outil est ajouté à la base de données d'outils. Les outils qui s'affichent dans le panneau Tools (Outils) sont des liens vers des entrées de cette base de données.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un outil, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton **Menu**.
2. Cliquez sur **Tools** (Outils). Le panneau Tools (Outils) s'ouvre.
3. Pour ajouter un outil, cliquez sur **Add Tool** (Ajouter un outil) dans la barre d'outils. La fenêtre Tool Details (Détails de l'outil) s'ouvre.
4. Pour éditer un outil, sélectionnez-le, puis cliquez sur **Edit Tool** (Editer l'outil) dans la barre d'outils. La fenêtre Tool Details (Détails de l'outil) s'ouvre.
5. Définissez l'outil comme suit :

   Nom Entrez un nom d'outil unique. Si vous modifiez un outil, vous ne pouvez pas le renommer.

   Activée
   Cochez cette case pour activer l'outil sur le système et le rendre disponible pour utilisation. Décochez cette case pour créer l'outil sans l'activer, ou le rendre indisponible.
6. Dans l'onglet **SQL**, entrez les commandes SQL à exécuter sur le serveur ObjectServer lors de la sélection de l'outil à utiliser. Renseignez l'onglet comme suit :

   Activée
   Cochez cette case pour indiquer que les commandes SQL saisies doivent s'exécuter lorsque l'outil est utilisé.

   Exécuter pour chaque ligne sélectionnée
   Cochez cette case pour indiquer que les commandes SQL doivent s'exécuter une fois pour chaque ligne dans une sélection de ligne de liste d'événements.

   Commandes SQL
   Entrez les commandes SQL à exécuter lorsque l'outil est utilisé.

Pour générer les commandes SQL, utilisez les boutons de l'assistant SQL situés à droite de cette zone.

Conseil : Vous pouvez inclure une invite dans une instruction SQL ; ainsi, lorsque l'outil s'exécute, il appelle une fenêtre d'invite ou un menu contextuel, pour que les utilisateurs puissent entrer ou sélectionner des informations.

7. Dans l'onglet **Executable** (Exécutable), entrez les commandes externes à exécuter lors de la sélection de l'outil à utiliser. Renseignez l'onglet comme suit :

   Activée
   Cochez cette case pour indiquer que les commandes exécutables saisies doivent s'exécuter lorsque l'outil est utilisé.

   Exécuter pour chaque ligne sélectionnée
   Cochez cette case pour indiquer que les commandes exécutables doivent s'exécuter une fois par ligne dans une sélection de lignes de liste d'événements.

   Redirect output (Rediriger la sortie)
   Cochez cette case pour indiquer que la sortie doit être répercutée via une fenêtre en lecture seule lors de l'exécution des commandes. Décochez cette case pour supprimer la sortie.

   Redirect errors (Rediriger les erreurs)
   Cochez cette case pour indiquer que tous les messages d'erreur doivent être répercutés via une fenêtre en lecture seule lors de l'exécution des commandes. Décochez cette case pour supprimer les messages d'erreur.

   Executable Commands (Commandes exécutables)
   Entrez les commandes à exécuter lorsque cet outil est utilisé.

   Pour générer les commandes exécutables, utilisez les boutons de l'assistant SQL situés à droite de cette zone.

   Conseil : Vous pouvez inclure une invite dans une commande externe ; ainsi, lorsque l'outil s'exécute, il appelle une fenêtre d'invite ou un menu contextuel pour que les utilisateurs puissent entrer ou sélectionner des informations.

8. Dans l'onglet **Journal**, indiquez les paramètres du journal à appliquer lors de la sélection de l'outil à utiliser. Renseignez l'onglet comme suit :

   Force Journal Entry (Forcer l'entrée de journal)
   Cochez cette case pour forcer les utilisateurs à entrer le texte de journal avant d'exécuter l'outil. Ce texte est ajouté au journal de l'alerte ou des alertes sélectionnées lorsque l'outil est sélectionné.

   Exécuter pour chaque ligne sélectionnée
   Cochez cette case pour entrer les informations de journal une fois par ligne dans la sélection de liste d'événements.

   Entrée de journal
   Entrez le texte de l'entrée de journal.

   Pour générer le texte de journal, utilisez les boutons d'assistant situés à droite de cette zone.

Conseil : Vous pouvez inclure une invite dans une entrée de journal ; ainsi, lorsque l'outil s'exécute, il appelle une fenêtre d'invite ou un menu contextuel pour que les utilisateurs puissent entrer ou sélectionner des informations.

9. Dans l'onglet **Access** (Accès), indiquez les classes d'alerte auxquelles l'outil s'applique, ainsi que les groupes d'utilisateurs autorisés à utiliser l'outil. Renseignez l'onglet comme suit :

   Class Access (Accès aux classes)
   Sélectionnez les classes d'alerte pour lesquelles l'outil peut être utilisé. Pour sélectionner toutes les classes d'alerte, cliquez sur **Tick All** (Sélectionner tout) immédiatement à droite de la liste. Pour annuler toutes les sélections, cliquez sur **Tick None** (Sélectionner aucun) immédiatement à droite de la liste. Vous pouvez également cochez chaque case individuellement pour chaque classe requise.

   Group Access (Accès des groupes)
   Sélectionnez les groupes d'utilisateur pouvant utiliser cet outil. Pour sélectionner tous les groupes, cliquez sur **Tick All** (Sélectionner tout) immédiatement à droite de la liste. Pour annuler toutes les sélections, cliquez sur **Sélectionner aucun** immédiatement à droite de la liste. Vous pouvez également cochez chaque case individuellement pour chaque classe requise.

10. Dans l'onglet **Plateforme**, indiquez les plateformes du système d'exploitation pour lesquelles l'outil va être mis à disposition. Renseignez l'onglet comme suit :

    Available Platforms (Plateformes disponibles)
    Cochez les cases des systèmes d'exploitation sur lesquels l'outil est disponible. Pour sélectionner tous les systèmes d'exploitation, cliquez sur **Tick All** (Sélectionner tout) à droite de la liste. Pour annuler toutes les sélections, cliquez sur **Tick None** (Sélectionner aucun) à droite de cette liste.

    Important : Vous *devez* toujours indiquer les systèmes d'exploitation sur lesquels l'outil peut être utilisé. L'outil est uniquement disponible sur les plateformes sélectionnées.

11. Dans l'onglet **Description**, entrez une description facultative de cet outil. Cela peut s'avérer utile à quiconque souhaite comprendre le fonctionnement de l'outil.

12. Sauvegardez ou annulez les modifications comme suit :

    OK Cliquez sur ce bouton pour sauvegarder les détails de l'outil et fermer la fenêtre. Les nouveaux outils sont ajoutés au panneau Outils.

    Annuler
    Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Que faire ensuite

Les outils affichés dans le panneau Outils peuvent être ajoutés aux menus de bureau de la liste d'événements et de Conductor pour vous assister dans la gestion des alertes.

Tâches associées:

«Ajouts d'outils, de sous-menus et de séparateurs à un menu», à la page 103
Vous pouvez ajouter des outils, de sous-menus et de séparateurs comme éléments

de menu dans un menu de bureau.

«Création et édition d'invites», à la page 112
Vous pouvez créer des invites à utiliser avec des outils.

Référence associée:

Annexe C, «Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires», à la page 411
Vous pouvez utiliser un certain nombre de commandes SQL, d'expressions variables et de boutons d'aide pour récupérer des informations d'une liste d'événements en cours, de l'événement actuel ou de l'environnement du système d'exploitation. Vous pouvez utiliser ces expressions lors de la création d'un outil, d'un déclencheur ou d'une procédure SQL, ainsi que dans les paramètres transmis à une liste d'événements transitoires.

Suppression d'outils

Pourquoi et quand exécuter cette tâche

Pour supprimer un outil, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton **Menu**.
2. Cliquez sur **Tools** (Outils). Le panneau Tools (Outils) s'ouvre.
3. Sélectionnez l'outil à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. L'outil est supprimé.

Configuration des invites

Un outil liste d'événements peut inclure une fenêtre d'invite ou un menu contextuel afin de permettre aux utilisateurs d'entrer ou de sélectionner des informations.

Lors de la création ou de l'édition d'un outil, vous pouvez inclure une invite dans une instruction SQL, une commande externe ou une entrée de journal.

Les types d'invite suivants peuvent être créés :

- Chaîne : ce type crée une fenêtre d'invite qui accepte un ou plusieurs caractères.
- Entier : ce type crée une fenêtre d'invite qui accepte une valeur d'entier.
- Virgule flottante : ce type crée une fenêtre d'invite qui accepte un nombre en virgule flottante pouvant contenir un séparateur décimal.
- Heure : ce type crée une fenêtre d'invite qui accepte une heure.
- Choix fixe : ce type crée un menu contextuel alimenté par les options que vous spécifiez.
- Recherche : ce type crée un menu contextuel ou une liste déroulante alimentée par les valeurs stockées dans un fichier spécifié.
- Mot de passe : ce type crée une fenêtre d'invite qui accepte un ou plusieurs caractères comme mot de passe.
- Choix dynamique : ce type crée un menu contextuel ou une liste déroulante qui est alimentée par les résultats d'une recherche de base de données.

Création et édition d'invites

Vous pouvez créer des invites à utiliser avec des outils.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer une invite, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton **Menu**.
2. Cliquez sur **Prompts** (Invites). Le panneau Prompts (Invites) s'ouvre.
3. Pour ajouter une invite, cliquez sur **Add Prompt** (Ajouter une invite) dans la barre d'outils. La fenêtre Prompt Details (Détails de l'invite) s'ouvre. Les zones de cette fenêtre varient en fonction du type d'invite en cours de création ou d'édition.
4. Pour éditer une invite, sélectionnez l'outil à éditer, puis cliquez sur **Edit Prompt** (Editer l'invite) dans la barre d'outils. La fenêtre Prompt Details (Détails de l'invite) s'ouvre. Les zones de cette fenêtre varient en fonction du type d'invite en cours de création ou d'édition.
5. Pour créer ou éditer une invite de chaîne, renseignez la fenêtre comme suit :

   Nom Entrez un nom d'invite unique. Ce nom doit être conforme aux conventions de dénomination du serveur ObjectServer. Si vous modifiez une invite, vous ne pouvez pas la renommer.

   Invite Entrez le texte de l'invite devant apparaître lorsque l'outil demande des informations à l'utilisateur.

   Type Sélectionnez `Chaîne` pour créer une fenêtre d'invite qui accepte un ou plusieurs caractères.

   Valeur par défaut
   Entrez une valeur par défaut pour l'invite à afficher.
6. Pour créer ou éditer une invite d'entier, renseignez la fenêtre comme suit :

   Nom Entrez un nom d'invite unique. Ce nom doit être conforme aux conventions de dénomination du serveur ObjectServer. Si vous modifiez une invite, vous ne pouvez pas la renommer.

   Invite Entrez le texte de l'invite devant apparaître lorsque l'outil demande des informations à l'utilisateur.

   Type Sélectionnez `Entier` pour créer une fenêtre d'invite qui accepte une valeur entière.

   Valeur par défaut
   Entrez une valeur par défaut pour l'invite à afficher.
7. Pour créer ou éditer une invite à variable flottante, renseignez la fenêtre comme suit :

   Nom Entrez un nom d'invite unique. Ce nom doit être conforme aux conventions de dénomination du serveur ObjectServer. Si vous modifiez une invite, vous ne pouvez pas la renommer.

   Invite Entrez le texte de l'invite devant apparaître lorsque l'outil demande des informations à l'utilisateur.

   Type Sélectionnez `Float` (flottante) pour créer une fenêtre d'invite qui accepte un nombre en virgule flottante, qui contient un point décimal.

Valeur par défaut
Entrez une valeur par défaut pour l'invite à afficher.

8. Pour créer ou éditer une invite d'heure, renseignez la fenêtre comme suit :

   Nom Entrez un nom d'invite unique. Ce nom doit être conforme aux conventions de dénomination du serveur ObjectServer. Si vous modifiez une invite, vous ne pouvez pas la renommer.

   Invite Entrez le texte de l'invite devant apparaître lorsque l'outil demande des informations à l'utilisateur.

   Type Sélectionnez `Heure` pour créer une fenêtre d'invite qui accepte une heure. Pour une invite horaire, la valeur qui s'affiche par défaut est l'heure actuelle.

9. Pour créer ou éditer une invite de choix fixe, renseignez la fenêtre comme suit :

   Nom Entrez un nom d'invite unique. Ce nom doit être conforme aux conventions de dénomination du serveur ObjectServer. Si vous modifiez une invite, vous ne pouvez pas la renommer.

   Invite Entrez le texte de l'invite devant apparaître lorsque l'outil demande des informations à l'utilisateur.

   Type Sélectionnez `Fixed Choice` (Choix fixe) pour créer un menu contextuel peuplé par les options que vous spécifiez.

   Options
   Cette zone permet de spécifier des options de menu à inclure dans le menu contextuel. Pour créer une option, cliquez sur **Add option** (Ajouter une option) et entrez le nom de l'option dans la zone affichée dans la liste **Options**. Cliquez en dehors de la zone pour valider l'entrée. Vous pouvez également modifier une option existante en cliquant sur l'entrée dans la liste **Options** pus en cliquant sur **Editer l'option**. Pour supprimer une option, cliquez sur l'entrée, puis cliquez sur **Delete option** (Supprimer l'option).

10. Pour créer ou éditer une invite de recherche, renseignez la fenêtre comme suit :

    Nom Entrez un nom d'invite unique. Ce nom doit être conforme aux conventions de dénomination du serveur ObjectServer. Si vous modifiez une invite, vous ne pouvez pas la renommer.

    Invite Entrez le texte de l'invite devant apparaître lorsque l'outil demande des informations à l'utilisateur.

    Type Sélectionnez `Recherche` pour créer un menu contextuel ou une liste déroulante contenant les valeurs d'un fichier spécifié.

    Fichier
    Entrez le nom et le chemin d'accès au fichier dont le contenu doit être utilisé pour peupler le menu contextuel ou la liste déroulante associée à l'invite. Vous pouvez aussi cliquer sur **Parcourir** pour ouvrir une fenêtre de sélection de fichiers, accéder à l'emplacement approprié et sélectionner le fichier.

11. Pour créer ou éditer une invite de mot de passe, renseignez la fenêtre comme suit :

    Nom Entrez un nom d'invite unique. Ce nom doit être conforme aux conventions de dénomination du serveur ObjectServer. Si vous modifiez une invite, vous ne pouvez pas la renommer.

Invite Entrez le texte de l'invite devant apparaître lorsque l'outil demande des informations à l'utilisateur.

Type Sélectionnez `Mot de passe` pour créer une fenêtre d'invite qui accepte un mot de passe d'un ou plusieurs caractères. Pour une invite de mot de passe, les caractères s'affichent sous la forme d'astérisques lorsque l'utilisateur renseigne la zone.

12. Pour créer ou éditer une invite de choix dynamique, renseignez la fenêtre comme suit :

    Nom Entrez un nom d'invite unique. Ce nom doit être conforme aux conventions de dénomination du serveur ObjectServer. Si vous modifiez une invite, vous ne pouvez pas la renommer.

    Invite Entrez le texte de l'invite devant apparaître lorsque l'outil demande des informations à l'utilisateur.

    Type Sélectionnez `Dynamic Choice` (Choix dynamique) pour créer un menu contextuel ou une liste déroulante contenant les résultats d'une requête de base de données.

    Base de données
    Sélectionnez une base de données dans cette liste déroulante.

    Table Dans la liste déroulante, sélectionnez une table de la base de données.

    Afficher
    Sélectionnez un nom de colonne dans la liste déroulante. Vous définissez ainsi la colonne utilisée pour renseigner le menu d'invite.

    Assign (Attribuer)
    Sélectionnez un nom de colonne dans la liste déroulante. Vous définissez ainsi la colonne utilisée pour renvoyer une valeur dans la commande SQL, la commande externe ou l'entrée de journal contenant l'invite.

    Where Entrez une condition de recherche SQL. Par exemple : `Colname='Severity'`.

    Order By (Trier par)
    Sélectionnez un nom de colonne dans la liste déroulante. Vous définissez ainsi la colonne de tri des éléments dans le menu d'invite. Pour indiquer le sens de tri, utilisez les options **Ordre croissant** et **Ordre décroissant**.

13. Sauvegardez ou annulez les modifications comme suit :

    OK Cliquez sur ce bouton pour sauvegarder les détails d'invite et fermer la fenêtre. Les nouvelles invites sont ajoutées au panneau Invites.

    Annuler
    Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Que faire ensuite

Les invites affiché dans le panneau Invites peuvent être ajoutées aux outils configurés pour aider les opérateurs à gérer les alertes dans la liste d'événements. Lorsque cet outil est exécuté, une fenêtre d'invite ou un menu contextuel s'affiche pour que vous puissiez entrer ou sélectionner des informations.

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

Tâches associées:

«Création et édition d'outils», à la page 108
Lors de sa création, un outil est ajouté à la base de données d'outils. Les outils qui s'affichent dans le panneau Tools (Outils) sont des liens vers des entrées de cette base de données.

Suppression d'invites

Pourquoi et quand exécuter cette tâche

Pour supprimer une invite, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton **Menu**.
2. Cliquez sur **Prompts** (Invites). Le panneau Prompts (Invites) s'ouvre.
3. Sélectionnez l'invite à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. L'invite est supprimée.

Configuration des automatisations

Vous pouvez utiliser une automatisation pour détecter les modifications apportées au serveur ObjectServer et exécuter des réponses automatisées à ces modifications. Cela permet au serveur ObjectServer de traiter les alertes sans l'intervention d'un opérateur.

Pourquoi et quand exécuter cette tâche

Par exemple, les alertes réseau contiennent souvent le message `Link Down` (Liaison interrompue), qui signale un problème dans le réseau. Lorsque le problème est résolu, le serveur ObjectServer reçoit une autre alerte contenant le message `Link Up` (Liaison établie). Une automatisation correctement configurée permet au serveur ObjectServer d'associer automatiquement les deux alertes, de reconnaître que le message `Link Up` (Liaison établie) indique que le problème est résolu, puis de supprimer les deux alertes.

Vous pouvez également utiliser l'automatisation pour gérer le dédoublonnage, qui permet de réduire la quantité de données traitées dans le serveur ObjectServer en éliminant les événements en double. Netcool/OMNIbus contient un certain nombre d'automatisations standard.

Les *déclencheurs* sont la base du sous-système d'automatisation du serveur ObjectServer. Ils lancent automatiquement une action de déclenchement ou se *mettent en application* lorsque le serveur ObjectServer détecte un problème lié à un déclencheur. Dans un déclencheur, vous pouvez exécuter des commandes SQL, puis appeler des procédures en réponse à la modification.

Les *signaux* et *procédures* font également partie intégrante du sous-système d'automatisation. Les signaux peuvent être associés à des déclencheurs, de sorte que le serveur ObjectServer puisse automatiquement répondre lorsqu'un signal est émis. Les procédures sont des programmes exécutables créés pour réaliser des opérations communes. Vous pouvez les exécuter dans un déclencheur ou à partir de l'interface SQL interactive.

Concepts associés:

«interface interactive SQL», à la page 229
Vous pouvez utiliser l'utilitaire de ligne de commande de l'interface interactive SQL fourni avec Tivoli Netcool/OMNIbus pour émettre des commandes SQL en direction d'un serveur ObjectServer.

Référence associée:

«EXECUTE PROCEDURE», à la page 389
Après avoir créé une procédure, vous devez l'exécuter à l'aide de la commande `EXECUTE PROCEDURE` pour réaliser les actions de la procédure. Pour ce faire, utilisez l'interface interactive SQL (**nco_sql**) ou un déclencheur ou une procédure.

«Automatisations Tivoli Netcool/OMNIbus standard», à la page 196
Un ensemble d'automatisations standard est inclus à Tivoli Netcool/OMNIbus. Ces automatisations sont créées lors de l'initialisation de la base de données.

Configuration des déclencheurs

Vous pouvez créer et modifier des déclencheurs à partir des fenêtres Netcool/OMNIbus Administrator par défaut ou en utilisant un script SQL. Pour créer le script SQL, vous utilisez un éditeur de texte externe. Il existe trois types de déclencheur : déclencheurs de base de données, déclencheurs de signal et déclencheurs temporels.

Les déclencheurs de base de données sont activés lors du déclenchement d'un incident de base de données. Les déclencheurs de signal sont activés lorsqu'un système ou un signal défini par l'utilisateur est mis au premier plan. Les déclencheurs temporels sont mis en application de manière répétée selon une fréquence spécifiée.

Chaque déclencheur doit appartenir à un *groupe de déclenchement*, qui est une collecte de déclencheurs associés. Vous pouvez activer ou désactiver en une seule action tous les déclencheurs d'un groupe de déclenchement donné.

Création et édition de groupes de déclencheur

Lors de la création d'un déclencheur, vous devez l'attribuer à un groupe de déclencheurs. Vous pouvez créer des groupes de déclencheurs *avant* de créer vos déclencheurs (action séparée) ou *pendant* leur création. Cette tâche explique comment créer un groupe de déclencheurs dans une action séparée.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un groupe de déclencheurs, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **Trigger Groups** (Groupes de déclencheurs). Le panneau Trigger Groups (Groupes de déclencheurs) s'ouvre.
3. Pour ajouter un groupe de déclencheurs, cliquez sur **Add Trigger Group** (Ajouter un groupe de déclencheurs) dans la barre d'outils. La fenêtre Trigger Group Details (Détails du groupe de déclencheurs) s'ouvre.
4. Pour éditer un groupe de déclencheurs, sélectionnez-le, puis cliquez sur **Edit Trigger Group** (Editer le groupe de déclencheurs) dans la barre d'outils. La fenêtre Trigger Group Details (Détails du groupe de déclencheurs) s'ouvre.
5. Renseignez les zones comme suit :

Nom de groupe
Entrez un nom de groupe de déclencheurs unique. Si vous modifiez un groupe de déclencheurs, vous ne pouvez pas le renommer.

Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

Activée
Cochez cette case pour activer le groupe de déclencheurs.

Décochez cette case pour créer un groupe de déclencheurs sans l'activer, ou pour le rendre indisponible. Lorsque le groupe de déclencheurs est indisponible, tous ses déclencheurs sont également indisponibles. Vous pouvez également rendre des déclencheurs individuels disponibles ou indisponibles lorsque vous les créez ou les modifiez.

6. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails du groupe de déclencheurs et fermer la fenêtre. Les nouveaux groupes sont ajoutés au panneau Trigger Groups (Groupes de déclencheurs).

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Création et édition de déclencheurs de base de données»
Un déclencheur de base de données se met en application en cas de modification d'une base de données de déclenchement. Par exemple, vous pouvez créer un déclencheur pour réaliser une action à chaque insertion dans la table alerts.status.

«Création et édition de déclencheurs de signal», à la page 121
Les déclencheurs de signal sont activés lorsqu'un système ou un signal défini par l'utilisateur est mis au premier plan. Les signaux *système* sont émis de manière spontanée par le serveur ObjectServer lorsqu'il détecte des modifications apportées au système. Les signaux *définis par l'utilisateur* sont créés, émis et supprimés de manière explicite.

«Création et édition de déclencheurs temporels», à la page 124
Les déclencheurs temporels sont mis en application de manière répétée selon une fréquence spécifiée.

Création et édition de déclencheurs de base de données

Un déclencheur de base de données se met en application en cas de modification d'une base de données de déclenchement. Par exemple, vous pouvez créer un déclencheur pour réaliser une action à chaque insertion dans la table alerts.status.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un déclencheur de base de données, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).

2. Cliquez sur **Triggers** (Déclencheurs). Le panneau Triggers (Déclencheurs) s'ouvre.

   Il répertorie tous les déclencheurs de base de données, de signal et déclencheurs temporels configurés.

   Conseil : Pour afficher un seul type de déclencheur, cliquez sur **Show Database Triggers Only** (Afficher les déclencheurs de base de données uniquement), **Show Temporal Triggers Only** (Afficher les déclencheurs temporaires uniquement), ou **Show Signal Triggers Only** (Afficher les déclencheurs de signal uniquement) dans la barre d'outils.
3. Pour ajouter un déclencheur de base de données, cliquez sur **Add Database Trigger** (Ajouter un déclencheur de base de données) dans la barre d'outils. La fenêtre Database Trigger Details (Détails du déclencheur de base de données) s'ouvre.
4. Pour éditer un déclencheur de base de données, sélectionnez-le, puis cliquez sur **Edit Trigger** (Editer le déclencheur) dans la barre d'outils. La fenêtre Database Trigger Details (Détails du déclencheur de base de données) s'ouvre.
5. Définissez ou éditez les détails de configuration du déclencheur comme suit :

   Nom Entrez un nom de déclencheur unique. Si vous modifiez un déclencheur, vous ne pouvez pas le renommer.

   Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

   Groupe
   Sélectionnez le groupe auquel vous souhaitez affecter le déclencheur.

   Add new trigger group (Ajouter un nouveau groupe de déclencheurs)
   Cliquez sur ce bouton pour créer un groupe de déclencheurs auquel le déclencheur sera affecté. La fenêtre Trigger Group Details (Détails du groupe de déclencheurs) s'ouvre. Complétez cette fenêtre et enregistrez vos modifications.

   Lorsque vous revenez sur la fenêtre Database Trigger Details (Détails sur le déclencheur de bases de données), le nouveau groupe de déclencheurs s'affiche en tant que groupe sélectionné.
6. Renseignez l'onglet **Settings** (Paramètres) comme suit :

   On (Sur)
   Sélectionnez la base de données du serveur ObjectServer et la table de base de données qui exécute le déclencheur.

   Priority (Priorité)
   Sélectionnez un niveau de priorité qui définit l'ordre dans lequel le serveur ObjectServer exécute les déclencheurs, si la modification de la base de données entraîne l'activation de plusieurs déclencheurs. Vous pouvez sélectionnez les chiffres de 1 à 20, 1 étant le niveau de priorité le plus élevé.

Pre database action/Post database action (action pré-base de données/post-base de données)
Cliquez sur **Pre database action** (action pré-base de données) pour indiquer que l'action du déclencheur doit s'effectuer avant la modification de base de données qui a généré l'exécution du déclencheur. Cliquez sur **Post database action** (action post-base de données) pour indiquer que l'action du déclencheur doit s'effectuer après la modification de base de données qui a généré l'exécution du déclencheur.

Par exemple, cliquez sur **Pre database action** pour évaluer un nom d'utilisateur avant qu'une ligne de la table alerts.status soit supprimée. Dans le déclencheur, vous pouvez détecter si l'utilisateur est autorisé à effectuer des suppressions dans la table alerts.status, et, si ce n'est pas le cas, vous pouvez empêcher les modifications. Si vous cliquez sur **Post database action** (action post-base de données), la modification de base de données s'effectue toujours.

Supprimer/Insérer/Réinsérer/Mettre à jour
Utilisez ces options pour spécifier les modifications de base de données qui peuvent être effectuées.

Ligne/Instruction
Cliquez sur **Ligne** pour définir le déclencheur à exécuter une fois pour chaque ligne correspondant à la condition de déclencheur. Cliquez sur **Instruction** pour définir le déclencheur à exécuter une fois quel que soit le nombre de lignes correspondantes dans la table.

Debug (Débogage)
Cochez cette case pour envoyer des informations de débogage au journal de messages du serveur ObjectServer chaque fois que le déclencheur est exécuté.

Activé Cochez cette case pour activer le déclencheur et le rendre disponible pour utilisation. Décochez cette case pour créer le déclencheur sans l'activer, ou le rendre indisponible. Un déclencheur désactivé ne s'exécute pas lorsque la modification de la base de données associée se produit.

7. Dans l'onglet **When** (clause WHEN), indiquez une clause WHEN facultative permettant de tester une condition particulière avant de réaliser l'action. Si la condition n'est pas satisfaite, l'action n'est pas réalisée. Vous pouvez utiliser les boutons d'aide situés à droite de la zone pour construire la clause WHEN.
8. Dans l'onglet **Action**, entrez les commandes SQL correspondant au déclencheur.

   Le corps d'un déclencheur contient un ensemble de commandes SQL et de constructions de programmation qui manipulent les données dans le serveur ObjectServer. Le corps du déclencheur est encadré par les mots clés BEGIN et END. Chaque instruction, sauf la dernière, doit se terminer par un point virgule (;).

   Vous pouvez éventuellement définir (déclarer) les variables locales à insérer dans un déclencheur. Une variable locale est une marque de réservation pour les valeurs utilisées lors de l'exécution du déclencheur. Les déclarations de variables locales dans un déclencheur doivent être séparées par des points virgules (;).

   Le corps du déclencheur présente la syntaxe suivante :

```
[ DECLARE déclaration_variable;...[;] ]
BEGIN
 liste_instructions_déclencheur
END;
```

Pour générer les commandes SQL, utilisez les boutons de l'assistant SQL affichés à droite du panneau de l'éditeur SQL.

9. Dans l'onglet **Comment** (Commentaire), entrez un commentaire facultatif pour le déclencheur. Cela peut être utile aux personnes souhaitant savoir comment fonctionne le déclencheur.
10. Sauvegardez ou annulez les modifications comme suit :

    OK Cliquez sur ce bouton pour sauvegarder les détails du déclencheur et fermer la fenêtre. Les nouveaux déclencheurs sont ajoutés au panneau Triggers (Déclencheurs).

    Annuler
    Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

Tâches associées:

«Création et édition de groupes de déclencheur», à la page 116
Lors de la création d'un déclencheur, vous devez l'attribuer à un groupe de déclencheurs. Vous pouvez créer des groupes de déclencheurs *avant* de créer vos déclencheurs (action séparée) ou *pendant* leur création. Cette tâche explique comment créer un groupe de déclencheurs dans une action séparée.

Référence associée:

«CREATE TRIGGER (déclencheur de base de données)», à la page 373
La commande `CREATE TRIGGER` permet de créer les déclencheurs de base de données qui s'activent lorsqu'une modification ou une tentative de modification dans une table du serveur ObjectServer survient (ou lorsqu'une modification ou une tentative de modification dans une vue affecte une table de base).

«Valeurs recommandées lors de la création de déclencheurs», à la page 340
Lorsque vous créez ou modifiez des déclencheurs, vérifiez que ces derniers sont aussi efficaces que possible et qu'ils possèdent le plus faible temps d'exécution possible.

Annexe C, «Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires», à la page 411
Vous pouvez utiliser un certain nombre de commandes SQL, d'expressions variables et de boutons d'aide pour récupérer des informations d'une liste d'événements en cours, de l'événement actuel ou de l'environnement du système d'exploitation. Vous pouvez utiliser ces expressions lors de la création d'un outil, d'un déclencheur ou d'une procédure SQL, ainsi que dans les paramètres transmis à une liste d'événements transitoires.

Création et édition de déclencheurs de signal

Les déclencheurs de signal sont activés lorsqu'un système ou un signal défini par l'utilisateur est mis au premier plan. Les signaux *système* sont émis de manière spontanée par le serveur ObjectServer lorsqu'il détecte des modifications apportées au système. Les signaux *définis par l'utilisateur* sont créés, émis et supprimés de manière explicite.

Pourquoi et quand exécuter cette tâche

Par exemple, vous pouvez créer un déclencheur de signal pour envoyer un message électronique à un opérateur au démarrage ou à l'arrêt du serveur ObjectServer, étant donné qu'un signal système est généré dans ce cas.

Pour créer ou éditer un déclencheur de signal, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **Triggers** (Déclencheurs). Le panneau Triggers (Déclencheurs) s'ouvre.

   Il répertorie tous les déclencheurs de base de données, de signal et déclencheurs temporels configurés.

   Conseil : Pour afficher un seul type de déclencheur, cliquez sur **Show Database Triggers Only** (Afficher les déclencheurs de base de données uniquement), **Show Temporal Triggers Only** (Afficher les déclencheurs temporaires uniquement), ou **Show Signal Triggers Only** (Afficher les déclencheurs de signal uniquement) dans la barre d'outils.
3. Pour ajouter un déclencheur de signal, cliquez sur **Add Signal Trigger** (Ajouter un déclencheur de signal) dans la barre d'outils. La fenêtre Signal Trigger Details (Détails du déclencheur de signal) s'ouvre.
4. Pour éditer un déclencheur de signal, sélectionnez-le, puis cliquez sur **Edit Trigger** (Editer le déclencheur) dans la barre d'outils. La fenêtre Signal Trigger Details (Détails du déclencheur de signal) s'ouvre.
5. Définissez ou éditez les détails de configuration du déclencheur comme suit :

   Nom Entrez un nom de déclencheur unique. Si vous modifiez un déclencheur, vous ne pouvez pas le renommer.

   Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

   Groupe
   Sélectionnez le groupe auquel vous souhaitez affecter le déclencheur.

   Add New Trigger Group (Ajouter un nouveau groupe de déclencheurs)
   Cliquez sur ce bouton si vous souhaitez créer un nouveau groupe de déclencheurs auquel le déclencheur doit être affecté. La fenêtre Trigger Group Details (Détails du groupe de déclencheurs) s'ouvre. Complétez cette fenêtre et sauvegardez vos modifications.

Lorsque vous revenez sur la fenêtre Signal Trigger Details (Détails du déclencheur de signal), le nouveau groupe de déclencheur est affiché comme le groupe de déclencheurs sélectionné.

6. Renseignez l'onglet **Settings** (Paramètres) comme suit :

   Signal Sélectionnez le signal qui doit exécuter le déclencheur.

   Priority (Priorité)
   Sélectionnez une priorité afin de déterminer l'ordre dans lequel le serveur ObjectServer exécute les déclencheurs, si plusieurs sont exécutés suite au signal. Vous pouvez sélectionnez les chiffres de 1 à 20, 1 étant le niveau de priorité le plus élevé.

   Debug (Débogage)
   Cochez cette case pour envoyer des informations de débogage au journal de messages du serveur ObjectServer chaque fois que le déclencheur est exécuté.

   Activée
   Cochez cette case pour activer le déclencheur et le rendre disponible pour utilisation. Décochez cette case pour créer le déclencheur sans l'activer, ou le rendre indisponible. Un déclencher désactivé ne s'exécute pas lorsque le signal associé est émis.

7. Dans l'onglet **When** (clause WHEN), indiquez une clause WHEN facultative permettant de tester une condition particulière avant de réaliser l'action. Si la condition n'est pas satisfaite, l'action n'est pas réalisée. Vous pouvez utiliser les boutons d'aide situés à droite de la zone pour construire la clause WHEN.

8. Dans l'onglet **Evaluate** (Evaluation), construisez éventuellement un ensemble de résultats temporaire à partir d'une seule instruction SELECT traitée au cours de l'action du déclencheur définie dans l'onglet **Action**. Renseignez l'onglet comme suit :

   Bind As (Associer à)
   Entrez le nom de la table temporaire où l'ensemble de résultats doit être stocké.

   panneau Editeur SQL
   Entrez l'instructions sous le format suivant :

   ```
   EVALUATE SELECT_cmd
   ```

   Pour générer l'instruction, utilisez les boutons de l'assistant SQL affichés à droite de la zone.

9. Dans l'onglet **Action**, entrez les commandes SQL correspondant au déclencheur.

   Le corps d'un déclencheur contient un ensemble de commandes SQL et de constructions de programmation qui manipulent les données dans le serveur ObjectServer. Le corps du déclencheur est encadré par les mots clés BEGIN et END. Chaque instruction, sauf la dernière, doit se terminer par un point virgule (;).

   Vous pouvez éventuellement définir (déclarer) les variables locales à insérer dans un déclencheur. Une variable locale est une marque de réservation pour les valeurs utilisées lors de l'exécution du déclencheur. Les déclarations de variables locales dans un déclencheur doivent être séparées par des points virgules (;).

   Le corps du déclencheur présente la syntaxe suivante :

```
[ DECLARE déclaration_variable;...[;] ]
BEGIN
 liste_instructions_déclencheur
END;
```

Pour générer les commandes SQL, utilisez les boutons de l'assistant SQL affichés à droite du panneau de l'éditeur SQL.

10. Dans l'onglet **Comment** (Commentaire), entrez un commentaire facultatif pour le déclencheur. Cela peut s'avérer utile à quiconque souhaite comprendre le fonctionnement du déclencheur.
11. Sauvegardez ou annulez les modifications comme suit :

    OK Cliquez sur ce bouton pour sauvegarder les détails du déclencheur et fermer la fenêtre. Les nouveaux déclencheurs sont ajoutés au panneau Déclencheurs.

    Annuler
    Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Concepts associés:

«Configuration des signaux», à la page 139
Les signaux sont des occurrences situées dans le serveur ObjectServer qui peuvent être détectés et respectés. Les signaux font partie du sous-système d'automatisation.

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

Tâches associées:

«Création et édition de groupes de déclencheur», à la page 116
Lors de la création d'un déclencheur, vous devez l'attribuer à un groupe de déclencheurs. Vous pouvez créer des groupes de déclencheurs *avant* de créer vos déclencheurs (action séparée) ou *pendant* leur création. Cette tâche explique comment créer un groupe de déclencheurs dans une action séparée.

Référence associée:

«CREATE TRIGGER (déclencheur de signal)», à la page 375
La commande `CREATE TRIGGER` permet de créer un déclencheur de signal qui s'active en réponse à des incidents dans le serveur ObjectServer ou en réponse à un signal défini par l'utilisateur.

«Signaux système et leurs attributs», à la page 181
Si un signal système est émis, les attributs qui identifient la cause sont définis. Ces attributs sont transmis sous forme de variables implicites dans le déclencheur de signal associé.

«Valeurs recommandées lors de la création de déclencheurs», à la page 340
Lorsque vous créez ou modifiez des déclencheurs, vérifiez que ces derniers sont aussi efficaces que possible et qu'ils possèdent le plus faible temps d'exécution possible.

Annexe C, «Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires», à la page 411
Vous pouvez utiliser un certain nombre de commandes SQL, d'expressions variables et de boutons d'aide pour récupérer des informations d'une liste d'événements en cours, de l'événement actuel ou de l'environnement du système d'exploitation. Vous pouvez utiliser ces expressions lors de la création d'un outil, d'un déclencheur ou d'une procédure SQL, ainsi que dans les paramètres transmis à une liste d'événements transitoires.

Création et édition de déclencheurs temporels

Les déclencheurs temporels sont mis en application de manière répétée selon une fréquence spécifiée.

Pourquoi et quand exécuter cette tâche

Par exemple, un déclencheur temporel permet de supprimer toutes les lignes en clair (`Severity = 0`) de la table alerts.status qui n'ont pas été modifiées depuis un certain temps.

Pour créer ou éditer un déclencheur temporel, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **Triggers** (Déclencheurs). Le panneau Triggers (Déclencheurs) s'ouvre.

   Il répertorie tous les déclencheurs de base de données, de signal et déclencheurs temporels configurés.

   Conseil : Pour afficher un seul type de déclencheur, cliquez sur **Show Database Triggers Only** (Afficher les déclencheurs de base de données uniquement), **Show Temporal Triggers Only** (Afficher les déclencheurs temporaires uniquement), ou **Show Signal Triggers Only** (Afficher les déclencheurs de signal uniquement) dans la barre d'outils.
3. Pour ajouter un déclencheur temporel, cliquez sur **Add Temporal Trigger** (Ajouter un déclencheur temporel) dans la barre d'outils. La fenêtre Temporal Trigger Details (Détails du déclencheur temporel) s'ouvre.
4. Pour éditer un déclencheur temporel, sélectionnez-le, puis cliquez sur **Edit Trigger** (Editer le déclencheur) dans la barre d'outils. La fenêtre Temporal Trigger Details (Détails du déclencheur temporel) s'ouvre.
5. Définissez ou éditez les détails de configuration du déclencheur comme suit :

   Nom Entrez un nom de déclencheur unique. Si vous modifiez un déclencheur, vous ne pouvez pas le renommer.

   Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

   Groupe
   Sélectionnez le groupe auquel vous souhaitez affecter le déclencheur.

   Add New Trigger Group (Ajouter un nouveau groupe de déclencheurs)
   Cliquez sur ce bouton si vous souhaitez créer un nouveau groupe de déclencheurs auquel le déclencheur doit être affecté. La fenêtre Trigger Group Details (Détails du groupe de déclencheurs) s'ouvre. Complétez cette fenêtre et sauvegardez vos modifications.

Lorsque vous revenez sur la fenêtre Temporal Trigger Details (Détail du déclencheur temporaire), le nouveau groupe de déclencheurs temporaires est ajouté à la liste **Groupe**. Vous pouvez alors affecter le déclencheur à ce groupe.

6. Renseignez l'onglet **Settings** (Paramètres) comme suit :

   Every (Tous les)
   Définit la fréquence d'exécution du déclencheur. Indiquez une valeur numérique et sélectionnez **heures**, **minutes**, ou **secondes** dans la liste déroulante.

   Priority (Priorité)
   Sélectionnez une priorité pour définir l'ordre d'exécution des déclencheurs par le serveur ObjectServer, si plusieurs déclencheurs sont définis pour une fréquence. Vous pouvez sélectionnez les chiffres de 1 à 20, 1 étant le niveau de priorité le plus élevé.

   Debug (Débogage)
   Cochez cette case pour envoyer des informations de débogage au journal de messages du serveur ObjectServer chaque fois que le déclencheur est exécuté.

   Activée
   Cochez cette case pour activer le déclencheur et le rendre disponible pour utilisation. Décochez cette case pour créer le déclencheur sans l'activer, ou le rendre indisponible. Un déclencheur désactivé ne s'exécute pas.

7. Dans l'onglet **When** (clause WHEN), indiquez une clause WHEN facultative permettant de tester une condition particulière avant de réaliser l'action. Si la condition n'est pas satisfaite, l'action n'est pas réalisée. Vous pouvez utiliser les boutons d'aide situés à droite de la zone pour construire la clause WHEN.

8. Dans l'onglet **Evaluate** (Evaluation), construisez éventuellement un ensemble de résultats temporaire à partir d'une seule instruction SELECT traitée au cours de l'action du déclencheur définie dans l'onglet **Action**. Renseignez l'onglet comme suit :

   Bind As (Associer à)
   Entrez le nom de la table temporaire où l'ensemble de résultats doit être stocké.

   panneau Editeur SQL
   Entrez l'instruction à l'aide du format suivant :

   ```
   EVALUATE commande_SELECT
   ```

   Pour générer l'instruction, utilisez les boutons de l'assistant SQL affichés à droite du panneau de l'éditeur SQL.

9. Dans l'onglet **Action**, entrez les commandes SQL correspondant au déclencheur.

   Le corps d'un déclencheur contient un ensemble de commandes SQL et de constructions de programmation qui manipulent les données dans le serveur ObjectServer. Le corps du déclencheur est encadré par les mots clés BEGIN et END. Chaque instruction, sauf la dernière, doit se terminer par un point virgule (;).

   Vous pouvez éventuellement définir (déclarer) les variables locales à insérer dans un déclencheur. Une variable locale est une marque de réservation pour

les valeurs utilisées lors de l'exécution du déclencheur. Les déclarations de variables locales dans un déclencheur doivent être séparées par des points virgules (;).

Le corps du déclencheur présente la syntaxe suivante :

```
[ DECLARE déclaration_variable;...[;] ]
BEGIN
 liste_instructions_déclencheur
END;
```

Pour générer les commandes SQL, utilisez les boutons de l'assistant SQL affichés à droite du panneau de l'éditeur SQL.

10. Dans l'onglet **Comment** (Commentaire), entrez un commentaire facultatif pour le déclencheur. Cela peut s'avérer utile à quiconque souhaite comprendre le fonctionnement du déclencheur.
11. Sauvegardez ou annulez les modifications comme suit :

    OK Cliquez sur ce bouton pour sauvegarder les détails du déclencheur et fermer la fenêtre. Les nouveaux déclencheurs sont ajoutés au panneau Déclencheurs.

    Annuler
    Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

Tâches associées:

«Création et édition de groupes de déclencheur», à la page 116
Lors de la création d'un déclencheur, vous devez l'attribuer à un groupe de déclencheurs. Vous pouvez créer des groupes de déclencheurs *avant* de créer vos déclencheurs (action séparée) ou *pendant* leur création. Cette tâche explique comment créer un groupe de déclencheurs dans une action séparée.

Référence associée:

«CREATE TRIGGER (déclencheur temporel)», à la page 377
La commande `CREATE TRIGGER` permet de créer des délencheurs temporels qui s'activent selon une fréquence spécifiée.

«Valeurs recommandées lors de la création de déclencheurs», à la page 340
Lorsque vous créez ou modifiez des déclencheurs, vérifiez que ces derniers sont aussi efficaces que possible et qu'ils possèdent le plus faible temps d'exécution possible.

Annexe C, «Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires», à la page 411
Vous pouvez utiliser un certain nombre de commandes SQL, d'expressions variables et de boutons d'aide pour récupérer des informations d'une liste d'événements en cours, de l'événement actuel ou de l'environnement du système d'exploitation. Vous pouvez utiliser ces expressions lors de la création d'un outil, d'un déclencheur ou d'une procédure SQL, ainsi que dans les paramètres transmis à une liste d'événements transitoires.

Utilisation d'un éditeur externe pour créer et éditer des déclencheurs

Dans Netcool/OMNIbus Administrator, vous pouvez configurer un éditeur de texte externe et l'utiliser pour créer ou éditer des déclencheurs.

Pourquoi et quand exécuter cette tâche

Configuration d'un éditeur externe pour les déclencheurs :

Le cas échéant, vous pouvez créer et éditer des déclencheurs à partir d'un éditeur de texte externe de votre choix. Vous devez en premier lieu configurer l'éditeur que vous souhaitez utiliser.

Pourquoi et quand exécuter cette tâche

Pour configurer un éditeur externe, procédez comme suit :

Procédure

1. Dans Netcool/OMNIbus Administrator, cliquez sur **Tools** > **Configure Tools** (Outils > Configurer les outils). La fenêtre Choose Tool (Choisir l'outil) s'ouvre.
2. Sélectionnez **Text Editor** (Editeur de texte). La fenêtre External Program (Programme externe) s'ouvre.
3. Renseignez cette fenêtre comme suit :

   Nom d'outil
   Cette zone affiche le type d'outil que vous configurez.

   Executable (Exécutable)
   Entrez le nom et le chemin complet du programme exécutable correspondant au type d'outil. Vous pouvez cliquez sur le bouton situé à droite de la zone pour rechercher et sélectionner le programme exécutable.

   Arguments
   Entrez les arguments de ligne de commande à exécuter avec ce programme exécutable.

   Run time environment (Environnement d'exécution)
   Sélectionnez l'environnement d'exécution.
4. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder les détails du programme et fermer la fenêtre.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Création et édition de déclencheurs dans un éditeur externe», à la page 128
Vous pouvez utiliser un éditeur externe pour créer ou éditer une base de données, un signal ou un déclencheur temporel.

Création et édition de déclencheurs dans un éditeur externe :

Vous pouvez utiliser un éditeur externe pour créer ou éditer une base de données, un signal ou un déclencheur temporel.

Avant de commencer

Un éditeur externe à utiliser doit être configuré pour la création ou l'édition de déclencheurs.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un déclencheur, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **Triggers** (Déclencheurs). Le panneau Triggers (Déclencheurs) s'ouvre. Il répertorie tous les déclencheurs de base de données, de signal et déclencheurs temporels configurés.

   Conseil : Pour afficher un seul type de déclencheur, cliquez sur **Show Database Triggers Only** (Afficher les déclencheurs de base de données uniquement), **Show Temporal Triggers Only** (Afficher les déclencheurs temporaires uniquement), ou **Show Signal Triggers Only** (Afficher les déclencheurs de signal uniquement) dans la barre d'outils.
3. Pour créer un déclencheur, vérifiez qu'aucune ligne n'est sélectionnée dans le panneau Triggers (Déclencheurs), puis cliquez sur **Edit in External Editor** (Editer dans l'éditeur externe) dans la barre d'outils (vous pouvez désélectionner une ligne en appuyant sur Ctrl et en cliquant cliquant sur la ligne). La boîte de dialogue Select Trigger Type (Sélectionner le type de déclencheur) s'ouvre. Procédez comme suit :
   a. Dans la liste **Trigger Template** (Modèle de déclencheur), sélectionnez le modèle du type de déclencheur que vous souhaitez créer. Si vous sélectionnez **Database** (Base de données), **Signal** ou **Temporal** (Temporel), l'éditeur externe s'ouvre avec la syntaxe correspondant au type de procédure. Si vous sélectionnez **<vierge>**, l'éditeur externe s'ouvre sans afficher de texte.
   b. Complétez la syntaxe du déclencheur. Si vous utilisez un modèle, remplacez la variable *nom_déclencheur* et *nom_groupe* par des valeurs réelles. De plus, pour un déclencheur de signal, remplacez la variable *nomSignal* par une valeur réelle. Dans le cas d'un déclencheur de base de données, remplacez la variable *nom_base_de_données.nom_table* par une valeur réelle. Ajoutez des instructions spécifiques au déclencheur, des clauses facultatives et des déclarations de variable, selon le cas. Ensuite, ajoutez le corps du déclencheur entre les mots clés Begin et End. Le modèle contient des commentaires (précédés par --) pour le placement du texte.
   c. Sauvegardez vos entrées. Le déclencheur est sauvegardé dans le serveur ObjectServer. En cas d'erreurs de syntaxe, vous êtes invité à recharger le contenu de l'éditeur externe.
   d. Fermez l'éditeur externe.
4. Pour éditer un déclencheur, sélectionnez-le, puis cliquez sur **Edit in External Editor** (Editer dans l'éditeur externe) dans la barre d'outils ou cliquez à l'aide du bouton droit de la souris et sélectionnez **Edit in External Editor** (Editer

dans l'éditeur externe) dans le menu contextuel. L'éditeur externe s'ouvre et affiche la syntaxe du déclencheur. Procédez comme suit :

a. Editez la syntaxe du déclencheur et enregistrez vos modifications. Le déclencheur est sauvegardé dans le serveur ObjectServer. En cas d'erreurs de syntaxe, vous êtes invité à recharger le contenu de l'éditeur externe.
b. Fermez l'éditeur externe.

Résultats

Conseil : L'instruction SQL que vous entrez dans un éditeur externe est sauvegardée dans le serveur ObjectServer dans un fichier `.ed`. Vous pouvez vérifier la validité de la syntaxe dans le fichier `.ed` et les autres fichiers `.sql` issus de l'interface interactive SQL (s'exécutant en mode interface graphique).

Tâches associées:

«Configuration d'un éditeur externe pour les déclencheurs», à la page 127
Le cas échéant, vous pouvez créer et éditer des déclencheurs à partir d'un éditeur de texte externe de votre choix. Vous devez en premier lieu configurer l'éditeur que vous souhaitez utiliser.

«Utilisation de l'interface interactive SQL en mode interface graphique», à la page 164
Vous pouvez utiliser l'interface interactive SQL pour configurer le serveur ObjectServer en émettant des commandes SQL.

Référence associée:

«CREATE TRIGGER (déclencheur de base de données)», à la page 373
La commande `CREATE TRIGGER` permet de créer les déclencheurs de base de données qui s'activent lorsqu'une modification ou une tentative de modification dans une table du serveur ObjectServer survient (ou lorsqu'une modification ou une tentative de modification dans une vue affecte une table de base).

«CREATE TRIGGER (déclencheur de signal)», à la page 375
La commande `CREATE TRIGGER` permet de créer un déclencheur de signal qui s'active en réponse à des incidents dans le serveur ObjectServer ou en réponse à un signal défini par l'utilisateur.

«CREATE TRIGGER (déclencheur temporel)», à la page 377
La commande `CREATE TRIGGER` permet de créer des délencheurs temporels qui s'activent selon une fréquence spécifiée.

Suppression de groupes de déclencheurs

Vous ne pouvez pas supprimer un groupe de déclencheurs contenant des déclencheurs.

Pourquoi et quand exécuter cette tâche

Pour supprimer un groupe de déclencheurs, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **Trigger Groups** (Groupes de déclencheurs). Le panneau Trigger Groups (Groupes de déclencheurs) s'ouvre.
3. Sélectionnez le groupe de déclencheurs à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. Le groupe de déclencheurs est supprimé.

Suppression de déclencheurs

Pourquoi et quand exécuter cette tâche

Pour supprimer un déclencheur, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **Triggers** (Déclencheurs). Le panneau Triggers (Déclencheurs) s'ouvre.
3. Sélectionnez le déclencheur à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. Le déclencheur est supprimé.

Configuration de procédures

Une procédure est un objet SQL exécutable qui peut être appelé pour effectuer des opérations communes.

Les types de procédure sont les suivantes :

- Procédures SQL, qui manipulent les données dans une base de données du serveur ObjectServer
- Procédures externes, qui exécutent un fichier exécutable sur un système local ou distant

Conseil : Vous pouvez créer et modifier des procédures à partir des fenêtres Netcool/OMNIbus Administrator ou à partir de l'éditeur de texte externe de votre choix.

Création et édition de procédures SQL

Les procédures SQL se composent essentiellement de paramètres, de déclarations de variable locale et du corps de la procédure.

Pourquoi et quand exécuter cette tâche

Les paramètres dont des valeurs transmises vers ou à partir d'une procédure. Vous déclarez les paramètres lors de la création de la procédure, et indiquez les valeurs transmises en tant que paramètres lors de l'exécution de la procédure.

Les variables locales sont déclarées avant le corps de la procédure et peuvent permettre de définir les valeurs utilisées lors de l'exécution de la procédure.

Pour créer ou éditer une procédure SQL, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **Procedures** (Procédures). Le panneau Procedures (Procédures) s'ouvre.
3. Pour ajouter une procédure SQL, cliquez sur **Add SQL Procedure** (Ajouter une procédure SQL) dans la barre d'outils. La fenêtre SQL Procedure Details (Détails de la procédure SQL) s'ouvre.

4. Pour éditer une procédure SQL, sélectionnez-la, puis cliquez sur **Edit Procedure** (Editer la procédure) dans la barre d'outils. La fenêtre SQL Procedure Details (Détails de la procédure SQL) s'ouvre.
5. Complétez cette fenêtre comme suit :

Nom Entrez un nom de procédure unique. Si vous modifiez une procédure, vous ne pouvez pas la renommer.

Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

Paramètres
Cette zone affiche les paramètres créés pour accéder à et sortir d'une procédure.

Utilisez les flèches de déplacement situées à droite de la zone de liste pour modifier l'ordre d'un paramètre sélectionné. Vous pouvez également cliquer sur **Remove parameter** (Supprimer le paramètre) à droite de la zone de liste pour supprimer de la liste tout paramètre sélectionné.

Pour créer un paramètre, utilisez les zones **In/Out** (Interne/Externe), **Nom**, et **Data Type** (Type de données), la case à cocher**Array** (Tableau) (si nécessaire), et le bouton **Add parameter to the list** (Ajouter un paramètre à la liste).

In/Out (Entrée/Sortie)
Sélectionnez un mode de création pour le paramètre. Chaque paramètre de procédure a un mode, qui peut être **in** (interne), **out** (externe) ou **in out** (interne externe). En fonction du mode choisi pour vos paramètres, vous pouvez les utiliser de différentes façons.

Nom Entrez le nom du paramètre à créer. Les noms de paramètre doivent être uniques dans une procédure.

Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

Data Type (Type de données)
Sélectionnez le type de données que le paramètre peut insérer ou supprimer de la procédure. Le type de données peut être tout type de données valide du serveur ObjectServer, sauf VARCHAR et INCR.

Array (Tableau)
Si vous avez sélectionné le mode **in** dans la liste déroulante

In/Out (Interne/Externe), vous pouvez cocher cette case pour passer un tableau dans le type de données sélectionné dans la procédure.

Add parameter to the list (Ajouter un paramètre à la liste)
Après avoir renseigné les zones **In/Out** (Interne/Externe), **Nom** et **Data Type** (Type de données), et éventuellement coché la case **Array** (Tableau), cliquez sur ce bouton pour ajouter un paramètre à la liste de paramètres.

Actions
Entrez les commandes SQL de cette procédure. Le corps d'une procédure contient un ensemble de commandes SQL et de constructions de programmation qui manipulent les données dans le serveur ObjectServer. Le corps d'une procédure est encadré par les mots clés BEGIN et END. Chaque instruction, sauf la dernière, doit se terminer par un point virgule (;).

Vous pouvez éventuellement définir (déclarer) les variables locales à insérer dans la procédure. Une variable locale est une marque de réservation pour les valeurs utilisées lors de l'exécution de la procédure. Les déclarations de variables locales dans une procédure doivent être séparées par des points virgules (;).

Pour générer les commandes SQL, utilisez les boutons de l'assistant SQL affichés à droite du panneau de l'éditeur SQL.

6. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder la procédure SQL et fermer la fenêtre. Les nouvelles procédures SQL sont ajoutées au panneau Procedures (Procédures).

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Que faire ensuite

Après avoir créé une procédure dans le serveur ObjectServer, vous pouvez l'exécuter à partir de l'interface SQL (**iSQL**) ou dans un déclencheur à l'aide de la commande EXECUTE PROCEDURE.

Référence associée:

«CREATE PROCEDURE (procédures SQL)», à la page 367
La commande `CREATE PROCEDURE` permet de créer des procédures SQL.

«EXECUTE PROCEDURE», à la page 389
Après avoir créé une procédure, vous devez l'exécuter à l'aide de la commande `EXECUTE PROCEDURE` pour réaliser les actions de la procédure. Pour ce faire, utilisez l'interface interactive SQL (**nco_sql**) ou un déclencheur ou une procédure.

«Spécification des types de données des colonnes», à la page 171
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.

Annexe C, «Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires», à la page 411
Vous pouvez utiliser un certain nombre de commandes SQL, d'expressions variables et de boutons d'aide pour récupérer des informations d'une liste d'événements en cours, de l'événement actuel ou de l'environnement du système d'exploitation. Vous pouvez utiliser ces expressions lors de la création d'un outil, d'un déclencheur ou d'une procédure SQL, ainsi que dans les paramètres transmis à une liste d'événements transitoires.

Exemple : procédure externe

Cet exemple de procédure SQL génère un rapport sur le nombre total d'alertes reçues et d'alertes en double supprimées pour un noeud donné.

Dans la fenêtre SQL Procedure Details, la procédure SQL nommée node_report est créée avec les entrées suivantes :

Tableau 23. Entrées de la procédure SQL node_report dans la fenêtre SQL Procedure Details

Zone	Entrée
Nom	`node_report`
Paramètres	`in node_name Char(255)` Cette entrée en lecture seule de la liste Parameters est générée à partir d'entrées effectuées dans les zones **In/Out**, **Name**, **Data Type** et **Length** de la section **Parameters**. Par exemple : • **In/Out** : `in` • **Name** : `node_name` • **Data Type** : `Char` • **Length** : `255`

Actions	`declare`
	`  tally_total integer;`
	`begin`
	`  set tally_total = 0;`
	`  for each row tmprow in alerts.status where tmprow.Node = node_name`
	`  begin`
	`  set tally_total = tally_total + tmprow.Tally;`
	`  end;`
	`  write into node_report_file values ( 'Total tally for node ', node_name, ' = ', tally_total );`
	`end`

La commande SQL permettant de créer le fichier du serveur ObjectServer `node_report_file` et le texte SQL complet de la même procédure `node_report` se présente comme suit :

```
create file node_report_file '/tmp/node_report';

create procedure node_report( node_name char(255) )
declare
     tally_total integer;
begin
     set tally_total = 0;
     for each row tmprow in alerts.status where tmprow.Node = node_name
     begin
     set tally_total = tally_total + tmprow.Tally;
     end;
     write into node_report_file values ( 'Total tally for node ', node_name,
 ' = ', tally_total );
end;
```

Création et édition des procédures externes

Vous pouvez créer des procédures externes afin d'exécuter un programme exécutable sur un système local ou distant.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer une procédure externe, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **Procedures** (Procédures). Le panneau Procedures (Procédures) s'ouvre.
3. Pour ajouter une procédure externe, cliquez sur **Add External Procedure** (Ajouter une procédure externe) dans la barre d'outils. La fenêtre External Procedure Details (Détails de la procédure externe) s'ouvre.
4. Pour éditer une procédure externe, sélectionnez-la, puis cliquez sur **Edit Procedure** (Editer la procédure) dans la barre d'outils. La fenêtre External Procedure Details (Détails de la procédure externe) s'ouvre.
5. Complétez cette fenêtre comme suit :

   Nom Entrez un nom unique pour la procédure. Si vous modifiez une procédure, vous ne pouvez pas la renommer.

   Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

   Paramètres
   Cette zone affiche les paramètres qui ont été créé pour la procédure. Ces paramètres peuvent être des variables de type de base, des tableaux de valeurs de type de base, ou des lignes de tables désignées.

   Vous pouvez utiliser les flèches de déplacement situées à droite de la zone de liste pour modifier la position d'un paramètre sélectionné. Vous pouvez également cliquer sur **Remove parameter** (Supprimer le paramètre) à droite de la zone de liste pour supprimer tout paramètre sélectionné de la liste.

   Pour créer un paramètre, utilisez les zones **In/Out** (interne/externe), **Nom** et **Data Type** (Type de données), la case à cocher **Array** (Tableau) et le bouton **Add parameter to the list** (Ajouter un paramètre à la liste).

   In/Out (Interne/Externe)
   Le mode **in** (interne) est sélectionné par défaut. Vous ne pouvez pas modifier cette valeur car les paramètres de procédures externe sont toujours de type `IN`. Vous pouvez utiliser un paramètre `IN` dans les expressions pour vous aider à calculer une valeur, mais vous ne pouvez pas attribuer de valeur au paramètre.

   Nom Entrez un nom pour le paramètre créé. Les noms de paramètres doivent être uniques dans la procédure.

Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

Data Type (Type de données)
Sélectionnez le type de données que le paramètre peut transmettre dans la procédure. Le type de données peut être tout type de données valide du serveur ObjectServer, sauf VARCHAR et INCR.

Array (Tableau)
Cochez cette case pour faire passer un tableau du type de données sélectionné dans la procédure.

Add parameter to the list (Ajouter un paramètre à la liste)
Après avoir renseigné les zones **In/Out** (Interne/Externe), **Nom** et **Data Type** (Type de données), et coché la case **Array** (Tableau), cliquez sur ce bouton pour ajouter un paramètre à la liste de paramètres.

Executable (Exécutable)
Entrez le chemin complet de la commande à exécuter. Pour rechercher et sélectionner le fichier, cliquez sur **Parcourir**.

Arguments
Entrez les arguments de ligne de commande pour la commande à exécuter.

Hôte Entrez le nom de l'ordinateur hôte sur lequel la procédure dot être exécutée. Le nom de l'ordinateur connecté s'affiche par défaut.

ID utilisateur
Indiquez l'ID utilisateur (UNIX) sous lequel le programme exécutable doit être exécuté.

ID groupe
Indiquez l'ID de groupe (UNIX) sous lequel le programme exécutable doit être exécuté.

6. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder la procédure externe et fermer la fenêtre. Les nouvelles procédures externes sont ajoutées au panneau Procedures (Procédures).

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Que faire ensuite

Après avoir créé une procédure dans le serveur ObjectServer, vous pouvez l'exécuter à partir de l'interface SQL (**iSQL**) ou dans un déclencheur à l'aide de la commande EXECUTE PROCEDURE.

Remarque : Pour exécuter une procédure externe, un démon d'agent de contrôle de processus (**nco_pad**) doit être exécuté sur l'hôte sur lequel est enregistré le fichier exécutable.

Concepts associés:

Chapitre 7, «Utilisation du contrôle de processus pour gérer des processus et des procédures externes», à la page 247
Le système de contrôle Tivoli Netcool/OMNIbus effectue deux tâches principales. Il gère les processus locaux et distants, puis exécute les processus externes spécifiés dans les automatisations.

Référence associée:

«Spécification des types de données des colonnes», à la page 171
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.

«EXECUTE PROCEDURE», à la page 389
Après avoir créé une procédure, vous devez l'exécuter à l'aide de la commande `EXECUTE PROCEDURE` pour réaliser les actions de la procédure. Pour ce faire, utilisez l'interface interactive SQL (**nco_sql**) ou un déclencheur ou une procédure.

Exemple : procédure externe

Cet exemple de procédure externe appelle un programme nommé **nco_mail**, qui envoie un e-mail sur les alertes critiques non reconnues.

Dans la fenêtre External Procedure Details, la procédure externe nommée send_email est créée avec les entrées suivantes :

Tableau 24. Entrées de la procédure externe send-email de la fenêtre External Procedure Details

Zone	Entrée
Nom	`send_email`

Paramètres	`in node Char(255)`
	`in severity Integer`
	`in subject Char(255)`
	`in email Char(255)`
	`in summary Char(255)`
	`in hostname Char(255)`
	
	Ces entrées en lecture seule de la liste Parameters sont générées à partir d'entrées effectuées dans les zones **In/Out**, **Name**, **Data Type** et **Length** de la section **Parameters**. Par exemple, pour `in node Char(255)`, les entrées sont les suivantes :
	• **In/Out** : `in`
	• **Name** : `node`
	• **Data Type** : `Char`
	• **Length** : `255`

Executable	`$NCHOME/omnibus/utils/nco_mail`
Arguments	`'\''+node+'\'', severity,'\''+subject+'\'','\''+email+'\'','\''+summary+'\''`
Host	`localhost`

Le texte SQL complet de la même procédure send_email est la suivante :

```
create or replace procedure send_email
(in node character(255), in severity integer, in subject character(255),
in email character(255), in summary character(255), in hostname character(255))
executable '$NCHOME/omnibus/utils/nco_mail'
host 'localhost'
user 0
group 0
arguments '\'' + node + '\'', severity, '\'' + subject + '\'',
'\'' + email + '\'', '\'' + summary + '\'';
```

Cet exemple présente également comment transmettre les chaînes de texte à un exécutable. Les chaînes doivent être incluses entre guillemets, qui doivent être ignorés à l'aide de barres obliques inverses. Tous les guillemets de cet exemple sont des guillemets simples.

Utilisation d'un éditeur externe pour créer et éditer des procédures

Dans Netcool/OMNIbus Administrator, vous pouvez configurer un éditeur de texte externe et l'utiliser pour créer ou éditer des procédures.

Pourquoi et quand exécuter cette tâche

Configuration d'un éditeur externe pour les procédures :

Le cas échéant, vous pouvez créer et éditer des procédures à partir d'un éditeur de texte externe de votre choix. Vous devez en premier lieu configurer l'éditeur que vous souhaitez utiliser.

Pourquoi et quand exécuter cette tâche

Pour configurer un éditeur externe, procédez comme suit :

Procédure

1. Dans Netcool/OMNIbus Administrator, sélectionnez **Tools** > **Configure Tools** (Outils > Configurer les outils). La fenêtre Choose Tool (Choisir l'outil) s'ouvre.
2. Sélectionnez **Text Editor** (Editeur de texte). La fenêtre External Program (Programme externe) s'ouvre.
3. Renseignez cette fenêtre comme suit :

   Nom d'outil
   Cette zone affiche le type d'outil que vous configurez.

   Executable (Exécutable)
   Entrez le nom et le chemin complet du programme exécutable correspondant au type d'outil. Vous pouvez cliquez sur le bouton situé à droite de la zone pour rechercher et sélectionner le programme exécutable.

   Arguments
   Entrez les arguments de ligne de commande à exécuter avec ce programme exécutable.

   Run time environment (Environnement d'exécution)
   Sélectionnez l'environnement d'exécution.
4. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder les détails du programme et fermer la fenêtre.

Annuler

Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Création et édition des procédures dans un éditeur externe»
Une fois que vous avez configuré un éditeur externe à utiliser pour la création ou l'édition de procédures, utilisez-le pour créer ou éditer une procédure SQL ou externe.

Création et édition des procédures dans un éditeur externe :

Une fois que vous avez configuré un éditeur externe à utiliser pour la création ou l'édition de procédures, utilisez-le pour créer ou éditer une procédure SQL ou externe.

Procédure

Pour créer ou éditer une procédure, sélectionnez le bouton de menu **Automation** (Automatisation) dans la fenêtre Netcool/OMNIbus Administrator et cliquez sur **Procedures** (Procédures). Procédez ensuite comme suit :

- Pour créer une procédure :
  1. Dans le panneau Procedures (Procédures), vérifiez qu'aucune ligne n'est sélectionnée. Cliquez ensuite sur **Edit in External Editor** (Edition dans l'éditeur externe).

     Conseil : Vous pouvez désélectionner une ligne en appuyant sur Ctrl et en cliquant sur la ligne.
  2. Dans la liste **Procedure Template** (Modèle de procédure) de la boîte de dialogue Select Procedure Type (Sélection du type de procédure), sélectionnez le modèle de procédure. Si vous sélectionnez **SQL** ou **External** (Externe), l'éditeur externe s'ouvre avec la syntaxe correspondant au type de procédure. Si vous sélectionnez **<vierge>**, l'éditeur externe s'ouvre sans afficher de texte.
  3. Complétez la syntaxe de la procédure. Si vous utilisez un modèle, remplacez la variable *nom_procédure* par une valeur réelle. De plus, dans le cas d'une procédure externe, remplacez les variables *nomExécutable*, *nomHôte*, *IDutilisateur* et *IDgroupe* par des valeurs réelles. Ajoutez des instructions spécifiques à la procédure et d'autres déclarations, le cas échéant. Le modèle peut contenir des commentaires (précédés par --) pour le placement du texte.
  4. Sauvegardez vos entrées. Si la procédure contient des erreurs de syntaxe, vous êtes invité à charger de nouveau le contenu de l'éditeur externe.
  5. Fermez l'éditeur externe.
- Pour modifier une procédure :
  1. Sélectionnez une procédure et cliquez sur **Edit in External Editor** (Edition dans l'éditeur externe). L'éditeur externe s'ouvre et affiche la syntaxe de la procédure.
  2. Editez la syntaxe de la procédure et enregistrez vos modifications. Si la procédure contient des erreurs de syntaxe, vous êtes invité à charger de nouveau le contenu de l'éditeur externe.
  3. Fermez l'éditeur externe.

Résultats

L'instruction SQL que vous entrez dans un éditeur externe est sauvegardée dans le serveur ObjectServer dans un fichier `.ed`.

Que faire ensuite

Vous pouvez vérifier la validité de la syntaxe dans le fichier `.ed` et dans les autres fichiers `.sql` issus de l'interface interactive SQL qui s'exécute en mode interface graphique.

Tâches associées:

«Configuration d'un éditeur externe pour les procédures», à la page 137
Le cas échéant, vous pouvez créer et éditer des procédures à partir d'un éditeur de texte externe de votre choix. Vous devez en premier lieu configurer l'éditeur que vous souhaitez utiliser.

«Utilisation de l'interface interactive SQL en mode interface graphique», à la page 164
Vous pouvez utiliser l'interface interactive SQL pour configurer le serveur ObjectServer en émettant des commandes SQL.

Référence associée:

«CREATE PROCEDURE (procédures externes)», à la page 365
La commande `CREATE PROCEDURE` permet de créer des procédures externes.

«CREATE PROCEDURE (procédures SQL)», à la page 367
La commande `CREATE PROCEDURE` permet de créer des procédures SQL.

Suppression de procédures

Vous ne pouvez pas supprimer une procédure en cours d'utilisation dans un déclencheur.

Pourquoi et quand exécuter cette tâche

Pour supprimer une procédure, procédez comme suit :

Procédure

1. Dans Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **Procedures** (Procédures). Le panneau Procedures (Procédures) s'ouvre.
3. Sélectionnez la procédure à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. La procédure est supprimée.

Configuration des signaux

Les signaux sont des occurrences situées dans le serveur ObjectServer qui peuvent être détectés et respectés. Les signaux font partie du sous-système d'automatisation.

Les types de signaux sont les suivants :

- Signaux système
- Signaux définis par l'utilisateur

Un serveur ObjectServer met au premier plan automatiquement les signaux système lorsque certaines modifications surviennent dans le système (par exemple, pendant le démarrage du système ou un échec de connexion. Vous ne pouvez ni

créer, ni modifier ces signaux. Vous pouvez associer des déclencheurs aux signaux système pour créer des réponses automatiques aux problèmes survenant dans le serveur ObjectServer.

Les signaux définis par l'utilisateur sont définis par vous-même. Vous pouvez utiliser Netcool/OMNIbus Administrator pour créer vos propres signaux définis par l'utilisateur.

Tâches associées:

«Création et édition de déclencheurs de signal», à la page 121
Les déclencheurs de signal sont activés lorsqu'un système ou un signal défini par l'utilisateur est mis au premier plan. Les signaux *système* sont émis de manière spontanée par le serveur ObjectServer lorsqu'il détecte des modifications apportées au système. Les signaux *définis par l'utilisateur* sont créés, émis et supprimés de manière explicite.

Création et édition de signaux définis par l'utilisateur

A l'inverse des signaux système, qui sont prédéfinis et ne peuvent pas être configurés, les signaux définis par l'utilisateur doivent être créés ou supprimés de manière explicite.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un signal défini par l'utilisateur, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **User Defined Signals** (Signaux définis par l'utilisateur). Le panneau User Defined Signals (Signaux définis par l'utilisateur) s'ouvre.
3. Pour ajouter un signal défini par l'utilisateur, cliquez sur **Add User Defined Signal** (Ajouter un signal défini par l'utilisateur) dans la barre d'outils. La fenêtre User Defined Signal Details (Détails du signal défini par l'utilisateur) s'ouvre.
4. Pour éditer un signal défini par l'utilisateur, sélectionnez-le, puis cliquez sur **Edit User Defined Signal** (Editer le signal défini par l'utilisateur) dans la barre d'outils. La fenêtre User Defined Signal Details (Détails du signal défini par l'utilisateur) s'ouvre.
5. Complétez cette fenêtre comme suit :

   Signal Name (Nom de signal)
   Entrez un nom de signal unique. Si vous modifiez un signal, vous ne pouvez pas le renommer.

   Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

   Commentaire
   Entrez un commentaire pour le signal. Par exemple, vous pouvez ajouter un commentaire pour indiquer le déclencheur qui s'exécute lorsque ce signal est émis.

Paramètres
Cette zone affiche les paramètres propres au signal défini par l'utilisateur.

L'ordre d'apparition des paramètres doit correspondre à leur ordre d'apparition dans la commande RAISE SIGNAL destinée au déclencheur. Utilisez les flèches de défilement à droite de la zone de liste pour modifier l'emplacement d'un paramètre sélectionné. Vous pouvez également cliquer sur **Remove parameter** (Supprimer le paramètre) à droite de la zone de liste pour supprimer de la liste un paramètre sélectionné.

Pour créer un paramètre, utilisez les zones **Nom** et **Data Type** (Type de données), et le bouton **Add parameter to the list** (Ajouter le paramètre à la liste).

Nom Entrez le nom du paramètre à créer. Les noms de paramètres doivent être uniques dans le signal.

Data Type (Type de données)
Sélectionnez le type de données que le paramètre peut transmettre au signal. Le type de données peut être tout type de données valide du serveur ObjectServer, sauf VARCHAR et INCR.

Data Length(Longueur des données)
Pour les types de données **Char** uniquement, entrez la longueur de paramètre.

Add parameter to the list (Ajouter le paramètre à la liste)
Après avoir renseigné les zones **Nom**, **Data Type** (Type de données), et, le cas échéant, **Data Length** (Longueur de données), cliquez sur ce bouton pour ajouter le paramètre à la liste de paramètres.

6. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder le signal défini par l'utilisateur et fermer la fenêtre. Les nouveaux signaux sont ajoutés au panneau User Defined Signals (Signaux définis par l'utilisateur).

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Résultats

Exemple : signal et déclencheur définis par l'utilisateur

Cet exemple présente un signal défini par l'utilisateur nommé `illegal_delete`, et le déclencheur de base de données `DETECT_AN_ILLEGAL_DELETE` dans laquelle il est utilisé. Le déclencheur de base de données utilise le signal pour interrompre les suppressions survenant en dehors des heures de bureau.

Dans la fenêtre User Defined Signal Details, le signal défini par l'utilisateur nommé illegal_delete est créé avec les entrées suivantes :

Tableau 25. Entrées du signal défini par l'utilisateur illegal_delete de la fenêtre User Defined Signal Details

Zone	Entrée
Signal Name	`illegal_delete`
Comment	`S'utilise avec le déclencheur DETECT_AN_ILLEGAL_DELETE.`

Tableau 25. Entrées du signal défini par l'utilisateur illegal_delete de la fenêtre User Defined Signal Details (suite)

Zone	Entrée

Paramètres	`user_name Char(20)`
	`row_summary Char(20)`
	
	Ces entrées en lecture seule de la liste Parameters sont générées à partir d'entrées effectuées dans les zones **Name**, **Data Type** et **Data Length** de la section **Parameters**. Par exemple, pour `user_name Char(20)`, les entrées sont les suivantes :
	• **Name** : `user_name`
	• **Data Type** : `Char`
	• **Data Length** : `20`

Dans le texte SQL suivant du déclencheur de base de données de pré-insertion `DETECT_AN_ILLEGAL_DELETE`, la commande `raise signal` s'affiche en gras.

```
create trigger DETECT_AN_ILLEGAL_DELETE
group default_triggers
priority 1
before delete on alerts.status
for each row
begin
   if( ( (hourofday() > 17) and (minuteofhour() > 30) ) or (hourofday() < 9) ) then
      raise signal ILLEGAL_DELETE %user.user_name, old.Summary;
         cancel;
      end if;
end;
```

Ce déclencheur met au premier plan le signal défini par l'utilisateur `illegal_delete`. Normalement, le signal mis au premier plan est ensuite détecté et utilisé, par exemple, par un autre déclencheur.

Suppression de signaux définis par l'utilisateur

Vous ne pouvez pas supprimer un signal défini par l'utilisateur en cours d'utilisation par un déclencheur de signal.

Pourquoi et quand exécuter cette tâche

Pour supprimer un signal défini par l'utilisateur, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Automation** (Automatisation).
2. Cliquez sur **User Defined Signals** (Signaux définis par l'utilisateur). Le panneau User Defined Signals (Signaux définis par l'utilisateur) s'ouvre.
3. Sélectionnez le signal défini par l'utilisateur à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. Le signal défini par l'utilisateur est supprimé.

Configuration de l'apparence visuelle de la liste d'événements

Les conversions, les couleurs, les visuels de colonne et les classes déterminent l'affichage des informations d'alerte dans la liste d'événements.

Pourquoi et quand exécuter cette tâche

Création et édition des conversions

Les conversions sont associées aux colonnes de la table alerts.status du serveur ObjectServer et mettent en correspondance les valeurs entières des colonnes avec les chaînes. Les conversions configurées dans Netcool/OMNIbus Administrator sont utilisées dans la liste d'événements afin de convertir les valeurs entières en chaîne, pour des raisons de lisibilité.

Pourquoi et quand exécuter cette tâche

Par exemple, il existe des conversions par défaut pour les gravités d'événement. Si la gravité d'un événement est de `4`, le texte `Major` s'affiche pour la gravité d'événement de la liste d'événements.

Pour créer ou éditer une conversion, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Visual** (Visuel).
2. Cliquez sur **Conversions**. Le panneau Conversions s'ouvre.

   Pour afficher les conversions existantes d'une colonne, cliquez deux fois sur le nom de la colonne. Vous pouvez également cliquer sur le symbole circulaire (sous UNIX) ou le symbole plus (+) (sous Windows) à gauche du nom de la colonne.
3. Pour ajouter une conversion, cliquez sur **Add Conversion** (Ajouter une conversion) dans la barre d'outils. La fenêtre Conversion Details (Détails de la conversion) s'ouvre.
4. Pour éditer une conversion, sélectionnez-la, puis cliquez sur **Edit Conversion** (Editer la conversion) dans la barre d'outils. La fenêtre Conversion Details (Détails de la conversion) s'ouvre.
5. Complétez cette fenêtre comme suit :

   Colonne
   Sélectionnez le nom de la colonne contenant les données à convertir.

   Valeur Spécifiez la valeur entière à convertir.

   Conversion
   Entrez la chaîne qui doit remplacer la valeur.
6. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder la conversion et fermer la fenêtre. Les nouvelles conversions sont ajoutées au panneau Conversions.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Suppression de conversions

Pourquoi et quand exécuter cette tâche

Pour supprimer une conversion, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Visual** (Visuel).
2. Cliquez sur **Conversions**. Le panneau Conversions s'ouvre.
3. Sélectionnez la conversion à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. La conversion est supprimée.

Création et édition de couleurs de gravité d'événement pour les listes d'événements Windows

Dans les listes d'événements, différentes couleurs permettent de représenter les différents degrés de gravité d'événement. Vous pouvez afficher, créer et modifier les couleurs de gravité utilisées dans les listes d'événements Windows. Vous pouvez sélectionner différentes couleurs pour les alertes dont la réception a été accusée ou pas.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer des couleurs de gravité d'événement dans les listes d'événements Windows, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Visual** (Visuel).
2. Cliquez sur **Colors** (Couleurs). Le panneau Colors (Couleurs) s'ouvre.
3. Pour ajouter une couleur de gravité, cliquez sur **Add Color Definition** (Ajouter une définition de couleur) dans la barre d'outils. La fenêtre Color Details (Détails de la couleur) s'ouvre.
4. Pour éditer une couleur de gravité, sélectionnez-la, puis cliquez sur **Edit Color Definition** (Editer la définition de couleur) dans la barre d'outils. La fenêtre Color Details (Détails de la couleur) s'ouvre.
5. Complétez cette fenêtre comme suit :

   Gravité
   Si vous créez une couleur, indiquez la valeur de gravité des alertes.

   Conversion
   Cette zone affiche la conversion de cette gravité des alertes (le cas échéant). Les conversions permettent de traduire des valeurs d'entiers en chaînes pour une meilleure lisibilité. Par exemple, la conversion par défaut d'une gravité de `4` est `Majeure`.

   Unacknowledged (Sans acc de récep)
   Cette zone affiche la couleur de la gravité des alertes, lorsqu'aucun accusé de réception n'a été émis pour celle-ci dans les listes d'événements. Par défaut, les couleurs de la gravité des alertes pour lesquelles aucun accusé de réception n'a été émis sont les suivantes :

   - 0 - Vert
   - 1 - Violet

- 2 - Bleu
- 3 - Jaune
- 4 - Orange
- 5 - Rouge

Cliquez sur le bouton **Show color picker** (Afficher la palette de couleurs) pour sélectionner la couleur des alertes pour lesquelles aucun accusé de réception n'a été émis de cette gravité. A partir de la boîte de dialogue Palette de couleurs qui s'affiche, sélectionnez une couleur à l'aide de son échantillon, ainsi que de ses valeurs HSB et RVB, puis cliquez sur OK.

Avec acc de récep
Cette zone affiche la couleur de la gravité des alertes, lorsqu'un accusé de réception a été émis pour celle-ci dans les listes d'événements. Par défaut, les couleurs de la gravité des alertes pour lesquelles un accusé de réception a été émis sont les suivantes :

- 0 - Vert foncé
- 1 - Violet foncé
- 2 - Bleu foncé
- 3 - Jaune foncé
- 4 - Orange foncé
- 5 - Rouge foncé

Cliquez sur le bouton **Show color picker** (Afficher la palette de couleurs) pour sélectionner la couleur des alertes pour lesquelles un accusé de réception a été émis de cette gravité. A partir de la boîte de dialogue Palette de couleurs qui s'affiche, sélectionnez une couleur à l'aide de son échantillon, ainsi que de ses valeurs HSB et RVB, puis cliquez sur OK.

6. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails des couleurs et fermer la fenêtre. Les nouvelles couleurs de gravité sont ajoutées dans le panneau Couleurs.

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Création et édition de visuels de colonne

L'apparence visuelle des listes d'événements est définies par les paramètres des visuels de colonne. Pour chaque colonne de la liste d'événements, vous pouvez définir le texte du titre, les largeurs par défaut et maximale et la justification du titre et des données.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un visuel de colonnes, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Visual** (Visuel).
2. Cliquez sur **Column Visuals** (Visuels de colonne). Le panneau Column Visuals (Visuels de colonne) s'ouvre.

3. Pour ajouter un visuel de colonne, cliquez sur **Add Column Visual** (Ajouter un visuel de colonne) dans la barre d'outils. La fenêtre Column Visual Details (Détails du visuel de colonne) s'ouvre.
4. Pour éditer un visuel de colonne, sélectionnez-le, puis cliquez sur **Edit Column Visual** (Editer le visuel de colonne) dans la barre d'outils. La fenêtre Column Visual Details (Détails du visuel de colonne) s'ouvre.
5. Complétez cette fenêtre comme suit :

   Colonne
   Si vous créez un nouveau visuel de colonne, sélectionnez la colonne à laquelle vous voulez ajouter le visuel.

   Titre Entrez le titre que vous souhaitez afficher en tant qu'en-tête de colonne dans les listes d'événements Tivoli Netcool/OMNIbus.

   Default (Valeur par défaut)
   Indiquez la largeur de colonne par défaut (en caractères).

   Maximum
   Indiquez la largeur maximale de la colonne (en caractères).

   Titre Définissez la justification ou l'alignement du titre de la colonne.

   Colonne
   Définissez la justification ou l'alignement du contenu de la colonne.
6. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder les détails du visuel de colonne et fermer la fenêtre. Les nouveaux visuels de colonne sont ajoutés dans le panneau Column Visuals (Visuels de colonne).

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Suppression de visuels de colonne

Pourquoi et quand exécuter cette tâche

Pour supprimer un visuel de colonne, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Visual** (Visuel).
2. Cliquez sur **Column Visuals** (Visuels de colonne). Le panneau Column Visuals (Visuels de colonne) s'ouvre.
3. Sélectionnez le visuel de colonne à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. Le visuel de colonne est supprimé.

Création et édition de classes

Les événements du serveur ObjectServer comportent une classe attribuée par la sonde. Chaque classe peut être associée à un menu d'outils liste d'événements contenant les outils utiles aux événements de cette classe.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer une classe, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Visual** (Visuel).
2. Cliquez sur **Classes**. Le panneau Classes s'ouvre.
3. Pour ajouter une classe, cliquez sur **Add Class** (Ajouter une classe) dans la barre d'outils. La fenêtre Class Details (Détails de la classe) s'ouvre.
4. Pour éditer une classe, sélectionnez-la, puis cliquez sur **Edit Class** (Editer la classe) dans la barre d'outils. La fenêtre Class Details (Détails de la classe) s'ouvre.
5. Complétez cette fenêtre comme suit :

   Identificateur
   Si vous créez une classe, indiquez son identificateur. La sonde affecte un identificateur de classe aux alertes du serveur ObjectServer.

   IBM définit des identificateurs de classe pour des types d'équipements particuliers. Contactez le support IBM si vous souhaitez réserver une plage de classes pour le type de votre équipement. Une plage réservée au client de 88 000 à 89 000 est également disponible ; tous les utilisateurs sont libres de l'utiliser.

   Nom Entrez un libellé pour le type d'équipement à associer au numéro de classe.
6. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder les détails de la classe et fermer la fenêtre. Les nouvelles classes sont ajoutées au panneau Classes.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Suppression de classes

Pourquoi et quand exécuter cette tâche

Pour supprimer une classe, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Visual** (Visuel).
2. Cliquez sur **Classes**. Le panneau Classes s'ouvre.
3. Sélectionnez la classe à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. La classe est supprimée.

Configuration de bases de données, fichiers, propriétés, connexions et canaux ObjectServer

Vous pouvez configurer les composants système ObjectServer suivants : structures de base de données, fichiers, propriétés, connexions et canaux ObjectServer.

Pourquoi et quand exécuter cette tâche

Une base de données est une collection structurée de données permettant d'accéder rapidement aux informations requises. Une base de données relationnelle utilise les tables comme des conteneurs logiques pour enregistrer ces données dans des lignes et des colonnes.

Un fichier ObjectServer offre un moyen de consigner ou de renvoyer des informations relatives aux événements ObjectServer.

Les propriétés ObjectServer contrôlent le comportement du serveur ObjectServer.

Il est possible d'afficher et de gérer les connexions au serveur ObjectServer.

Les canaux permettent de définir le type de données d'événement à diffuser dans les notifications d'événement accéléré, ainsi que les destinataires des données.

Concepts associés:

Chapitre 6, «Configuration de la notification des événements accélérés», à la page 235
Vous pouvez configurer Tivoli Netcool/OMNIbus pour accélérer la notification des événements pouvant présenter un risque pour le système. Le système Accelerated Event Notification (AEN) permet d'accélérer les événements de priorité élevée afin de garantir une exécution des systèmes sans interruption.

Configuration des bases de données

La configuration des bases de données implique la création et la maintenance des tables de base de données et des colonnes de ces tables.

Vous pouvez créer et supprimer des bases de données. Vous pouvez également créer, supprimer et éditer (ou altérer) les tables et les colonnes de base de données.

Restriction : Vous n'êtes pas autorisé à apporter des modifications aux bases de données système. Dans Netcool/OMNIbus Administrator, les bases de données système possèdent un verrou en regard de leurs noms.

Concepts associés:

«Bases de données initialisées par le système», à la page 169
Lorsque vous initialisez un serveur ObjectServer, plusieurs bases de données par défaut sont créées.

Création de bases de données

Vous pouvez utiliser Netcool/OMNIbus Administrator pour créer et gérer des bases de données ObjectServer.

Pourquoi et quand exécuter cette tâche

Pour créer une base de données ObjectServer, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).

   Vous pouvez afficher et configurer les informations de table et de colonne pour chaque base de données répertoriée n'appartenant pas au système. Utilisez l'onglet **Data View** (Vue des données) pour afficher les données de la table. Utilisez l'onglet **Column Definitions** (Définitions de colonne) pour afficher les informations relatives aux colonnes de la table (leurs types de données et attributs, par exemple).

   Pour régénérer les informations affichées, cliquez sur l'icône de la base de données ou de la table sélectionnée ou sur le bouton de la barre d'outils **Refresh** (Régénérer).
3. Dans la barre d'outils, cliquez sur **Create Database** (Créer la base de données). La fenêtre Database Details (Détails de la base de données) s'ouvre.
4. Complétez cette fenêtre comme suit :

   Nom Entrez un nom unique pour la base de données.

   Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

   OK Cliquez sur ce bouton pour sauvegarder la base de données et fermer la fenêtre. Les nouvelles bases de données sont ajoutées au serveur ObjectServer et au panneau Databases, Tables and Columns (Bases de données, tables et colonnes).

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Que faire ensuite

Vous pouvez désormais ajouter des tables à la base de données.

Tâches associées:

«Création de tables», à la page 150
Une table possède un nombre fixe de colonnes de données. Chaque nom de colonne est unique dans la table. Une table ne contient aucune ou plusieurs lignes au format défini par la liste de colonnes de la table. Le nom de table qualifié complet est composé du nom de la base de données et du nom de la table, séparés par un point.

Création de tables

Une table possède un nombre fixe de colonnes de données. Chaque nom de colonne est unique dans la table. Une table ne contient aucune ou plusieurs lignes au format défini par la liste de colonnes de la table. Le nom de table qualifié complet est composé du nom de la base de données et du nom de la table, séparés par un point.

Pourquoi et quand exécuter cette tâche

Par exemple, la table `status` de la base de données des alertes est identifié par alerts.status.

Pour créer une table, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).
3. Sélectionnez la base de données dans laquelle vous créez la table.
4. Dans la barre d'outils, cliquez sur **Create Table** (Créer la table). La fenêtre Table Details (Détails de la table) s'ouvre.
5. Complétez cette fenêtre comme suit :

   Nom Entrez le nom de la table.

   Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

   Type Sélectionnez un des types de table suivants :

   - **Persistent** (Permanente) : Lorsque le serveur ObjectServer redémarre, une table permanente est recréée avec toutes les données.
   - **Virtual** (Virtuelle) : Lorsque le serveur ObjectServer redémarre, une table virtuelle est recréée avec la même description de table, mais sans données.

   zone Table
   Cette zone liste les détails de toutes les colonnes de la table. Utilisez les flèches de déplacement situées à droite pour changer l'emplacement d'une colonne sélectionnée dans la table.

   Add column (Ajouter une colonne)
   Cliquez sur ce bouton pour ajouter une colonne à la table. La fenêtre Column Details (Détails de la colonne) s'ouvre. Complétez cette fenêtre et sauvegardez vos modifications.

   Lorsque vous revenez sur la fenêtre Table Details (Détails de la table), la nouvelle colonne est ajoutée à la liste de colonnes.

   Edit column (Editer la colonne)
   Cliquez sur ce bouton si vous voulez modifier les détails de la colonne

sélectionnée. La fenêtre Column Details (Détails de la colonne) s'ouvre. Modifiez les détails et enregistrez vos modifications.

Lorsque vous revenez sur la fenêtre Table Details (Détails de la table), les mises à jour des détails de la colonnes apparaissent dans la liste de colonnes.

Delete column (Supprimer la colonne)
Cliquez sur ce bouton pour supprimer une colonne de la table. Aucune confirmation de suppression n'est requise.

6. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails de la table et fermer la fenêtre. Les nouvelles tables sont ajoutées au panneau Databases, Tables and Columns (Bases de données, tables et colonnes).

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Résultats

Conseil : Vous pouvez utiliser l'onglet **Data View** (Vue des données) de la fenêtre Databases, Tables and Columns (Bases de données, tables et colonnes) pour afficher les données de la table, puis cliquez sur l'onglet **Column Definitions** (Définitions de colonne) pour afficher des informations détaillées relatives aux colonnes de la table.

Tâches associées:

«Ajout et édition de colonnes d'une table»
Vous pouvez ajouter de nouvelles colonnes aux tables ObjectServer ou modifier des colonnes existantes. Une table peut contenir au maximum 512 colonnes, à l'exclusion des colonnes gérées par le système. La taille de ligne maximale d'une table, qui est la somme des longueurs de colonne de la ligne, est de 64 Ko. Vous ne pouvez pas ajouter de colonnes de clé primaire aux tables existantes.

Ajout et édition de colonnes d'une table

Vous pouvez ajouter de nouvelles colonnes aux tables ObjectServer ou modifier des colonnes existantes. Une table peut contenir au maximum 512 colonnes, à l'exclusion des colonnes gérées par le système. La taille de ligne maximale d'une table, qui est la somme des longueurs de colonne de la ligne, est de 64 Ko. Vous ne pouvez pas ajouter de colonnes de clé primaire aux tables existantes.

Procédure

Pour ajouter ou éditer une colonne de table, procédez comme suit :

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).
3. Dans le panneau Databases, Tables and Columns (Bases de données, tables et colonnes), sélectionnez la table dans laquelle vous ajoutez ou éditez une colonne. Cliquez ensuite sur l'onglet **Column Definitions** (Définitions de colonne).
4. Ajoutez ou éditez une colonne :
   - Pour ajouter une colonne, cliquez sur **Add Column** (Ajouter une colonne) dans la barre d'outils.

- Pour éditer une colonne, sélectionnez la colonne, puis cliquez sur **Edit Column** (Editer la colonne) dans la barre d'outils.

La fenêtre Column Details (Détails de la colonne) s'affiche et vous permet de spécifier les détails de la colonne.

5. Complétez cette fenêtre comme suit :

Column Name (Nom de colonne)
Entrez le nom de la colonne. Si vous éditez une colonne, vous ne pouvez pas modifier son nom.

Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

Data Type (Type de données)
Sélectionnez un type de données pour la colonne. Le type de données détermine la manière dont le serveur ObjectServer traite les données de la colonne. Si vous éditez une colonne, vous ne pouvez pas modifier son type de données. Vous pouvez choisir parmi les types de données suivants :

- `Entier` : entier signé 32 bits
- `UTC` : temps stocké sous la forme du nombre de secondes écoulées depuis le 1er janvier 1970 à minuit
- `VarChar` : chaîne de caractères de taille variable (longueur pouvant atteindre 8 192 octets)
- `Incr` (Incr.): entier non signé 32 bits à incrémentation automatique ne pouvant être mis à jour que par le système
- `Char` (Carac.) : chaîne de caractères de taille fixe (longueur pouvant atteindre 8 192 octets)
- `Unsigned` (Non signé) : entier non signé 32 bits
- `Boolean` (Booléen) : TRUE ou FALSE
- `Real` (Réel): nombre en virgule flottante signé 64 bits
- `Integer64` : entier signé 64 bits
- `Unsigned64` : entier non signé 64 bits

Length (Longueur)
Cette zone n'est disponible que lorsque vous sélectionnez `VarChar` ou `Char` (Carac.) dans la liste **Data Type** (Type de données). Indiquez la longueur de la colonne.

Primary Key (Clé primaire)
Cochez cette case pour indiquer que la colonne est une clé primaire. La ou les colonnes de clés primaires permettent d'identifier chaque ligne de manière unique. Une colonne de clé primaire doit afficher une valeur par défaut.

No Modify (Pas de modification)
Cochez cette case pour indiquer que les utilisateurs ne peuvent pas modifier de données dans cette colonne.

No Default (Pas de valeur par défaut)
Cochez cette case pour indiquer qu'une valeur doit être indiquée pour cette colonne dans une commande INSERT.

6. Sauvegardez ou annulez les modifications comme suit :

Appliquer
Si vous souhaitez ajouter plusieurs colonnes sans fermer la fenêtre Column Details (Détails des colonnes), cliquez sur ce bouton pour sauvegarder les détails de colonnes après avoir ajouté chaque ensemble d'entrées.

OK Après avoir entré les valeurs correspondant à la dernière colonne que vous souhaitez ajouter pour le moment, cliquez sur ce bouton pour sauvegarder les détails de colonnes et fermer la fenêtre. Les nouvelles colonnes sont ajoutées au panneau Databases, Tables and Columns (Bases de données, tables et colonnes).

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Que faire ensuite

Vous pouvez utiliser l'onglet **Data View** (Vue des données) de la fenêtre Databases, Tables and Columns (Bases de données, tables et colonnes) pour afficher les données de la table.

Indexation des colonnes de table

Vous pouvez utiliser des index pour améliorer les performances des bases de données du serveur ObjectServer. L'utilisation d'index correctement conçus permet de réduire ou d'éliminer le recours aux analyses de table pendant l'exécution de requêtes SQL et accélère l'extraction des données. Pour déterminer l'efficacité d'un index, consultez sa sélectivité.

Avant de commencer

Veillez à vous familiariser avec le fonctionnement des index en lisant les instructions d'indexation. N'oubliez pas que, bien qu'il n'y ait aucune limite concernant le nombre d'index que vous pouvez créer sur une table, vous devez utiliser les index avec modération.

Pourquoi et quand exécuter cette tâche

La sélectivité d'index est un classement qui détermine l'efficacité d'un index. Le classement de sélectivité d'un index est déterminé par le rapport du nombre de valeurs distinctes dans la table au nombre de lignes de la table.

Les classements possibles sont les suivants :

- Elevé : au moins 90 % des valeurs de la table sont uniques. Cela représente le classement de sélectivité idéal.
- Moyen : entre 50 % et 89 % des valeurs sont uniques.
- Bas : entre 10 % et 49 % des valeurs sont uniques.
- Inconnu : moins de 9 % des valeurs sont uniques.

Le classement de sélectivité d'un index est élevé si peu de lignes contiennent la même valeur et il est faible si plusieurs lignes contiennent la même valeur. Lorsque

vous devez déterminer les colonnes de table devant être indexées, tenez également compte du type de demandes SQL qui sont exécutées sur la table.

Dans le cas de tables volumineuses, l'ajout et la suppression d'index, ainsi que la visualisation de la sélectivité des index, peuvent consommer beaucoup de mémoire.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Référence associée:

«Instructions d'indexation», à la page 336
L'indexation peut affecter les performances de vos requêtes SQL. Sans indexation, une analyse de base de données complète est généralement effectuée lors de l'exécution d'une requête SQL. Utilisez l'indexation pour limiter le nombre de lignes à examiner.

Comment utiliser la sélectivité comme critère d'indexation des colonnes de table :

La sélectivité d'un index n'est pas le seul critère définissant si une colonne de table constitue un index effectif. Vous devez également prendre en compte les types de requêtes que vous souhaitez exécuter par rapport à l'index. Basez-vous sur les exemples ci-après pour décider des colonnes de table à indexer.

Les exemples suivants indiquent quand un index peut avoir un classement de sélectivité initial élevé ou bas, mais montrent également la façon dont le choix de création d'un index est impacté par le type de requêtes utilisées pour accéder à la table.

Exemple 1

L'exemple suivant renvoie un classement de sélectivité élevé : une table avec 30000 lignes et avec une colonne (A) contenant des valeurs comprises entre 1 et 15000 et dans laquelle aucune valeur n'intervient plus de deux fois.

Notez la façon dont l'utilisation de requêtes destinées à l'index peut déterminer l'efficacité de ce dernier.

Par exemple, si la clause WHERE suivante est utilisée dans une requête d'accès aux lignes de la table, le résultat est un classement de sélectivité élevé et la requête fait un très bon usage de l'index :

```
where A = 5 or A = 10
```

Toutefois, si la clause WHERE suivante est utilisée pour accéder aux lignes de la table, seules deux lignes parmi les 30000 lignes seront ignorées par l'index. Cela constitue alors un mauvais critère de sélection d'index.

```
where A > 1
```

Exemple 2

L'exemple suivant renvoie une classement de sélectivité faible : une table avec 20000 lignes et avec une colonne (B) contenant des valeurs 1 ou 2.

Notez la façon dont l'utilisation de requêtes destinées à l'index peut déterminer l'efficacité de ce dernier.

Par exemple, si la clause WHERE suivante est utilisée pour accéder aux lignes de la table, toutes les valeurs ont une correspondance avec la clause et il n'y a donc aucun intérêt à créer un index.

```
where B = 1 or B = 2
```

Toutefois, si la clause WHERE suivante est utilisée pour accéder aux lignes de la table, l'efficacité de l'index dépend de la répartition des valeurs 1 :

```
where B = 1
```

Les facteurs permettant de déterminer l'efficacité d'un index sont les suivants :

- S'il n'existe que peu d'instances de la valeur 1, un index réduit le nombre de lignes traitées et est donc justifié.
- Si la majorité des lignes contiennent la valeur 1, une colonne indexée ne présente aucun avantage.
- S'il existe une répartition égale des deux valeurs, un index permet de réduire le nombre de lignes traitées de 50 %. Il est alors justifié de créer un index.

Création d'index :

Utilisez Netcool/OMNIbus Administrator pour créer des index sur les colonnes des tables de base de données.

Pourquoi et quand exécuter cette tâche

Tivoli Netcool/OMNIbus prend en charge les structures d'index de hachage et d'arborescence d'index. L'index de hachage prend en charge les comparaisons d'égalité dans les requêtes SQL. L'arborescence d'index est un index organisé qui stocke les valeurs de colonne dans une structure triée et élargit les comparaisons aux égalités dans les requêtes SQL. Par conséquent, une arborescence d'index peut être utilisée dans des requêtes de plage et dans des requêtes avec clause ORDER BY.

Les noms d'index doivent commencer par un caractère alphabétique en majuscules ou minuscules. Chaque nom d'index doit être unique sur le serveur ObjectServer. Afin de faciliter l'identification des index et l'unicité de leur nom, utilisez une convention de dénomination.

Restriction : Vous ne pouvez pas créer d'index sur les colonnes des tables du système. Ces tables contiennent des métadonnées relatives aux objets du serveur ObjectServer et sont enregistrées dans la base de données de catalogue. De plus, vous ne pouvez pas créer d'index de hachage sur une seule zone de clé primaire. Vous ne pouvez pas créer d'arborescence d'index avec des valeurs de données booléennes.

Procédure

Pour créer des index de colonne :

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).

3. Sélectionnez la table contenant la colonne à indexer et cliquez sur l'onglet **Data View** (Vue des données). Toutes les colonnes actuellement indexées sont signalées par une icône.
4. Pour indexer une colonne, cliquez avec le bouton droit de la souris sur la colonne, puis cliquez sur **Indexes** > **Add Column Index**. Une fenêtre s'affiche et présente le classement de sélectivité de l'index de colonne.
5. Pour créer l'index, cliquez sur **Yes** (Oui).
6. Entrez le nom de l'index, spécifiez son type (index de hachage ou index d'arborescence), puis cliquez sur **OK**. L'index est alors créé. Sous l'onglet **Data View** (Vue des données), la colonne indexée est signalée par une icône d'index.

Résultats

Vous pouvez savoir si la table est indexée en examinant la table catalog.indexes, soit à partir de l'interface Netcool/OMNIbus Administrator soit en utilisant la commande SELECT. Les requêtes SQL portant sur la colonne indexée doivent à présent s'exécuter plus vite et permettre une extraction des données plus rapide.

Tâches associées:

«Affichage des détails de l'index», à la page 175
Pour connaître les colonnes qui ont été indexées, vous pouvez examiner le contenu de la table catalog.indexes dans l'interface Netcool/OMNIbus Administrator ou à l'aide de la commande SELECT.

«Utilisation de l'interface interactive SQL en mode interface graphique», à la page 164
Vous pouvez utiliser l'interface interactive SQL pour configurer le serveur ObjectServer en émettant des commandes SQL.

Affichage de la sélectivité des index :

Vous pouvez consulter la sélectivité d'un index d'une table indexée ou d'une table à indexer.

Procédure

Pour afficher la sélectivité des index de colonne :

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).
3. Sélectionnez la table contenant la colonne choisie et cliquez sur l'onglet **Data View** (Vue des données).
4. Cliquez à l'aide du bouton droit de la souris sur la colonne, puis cliquez sur **Indexes** > **Column Selectivity**. Une fenêtre s'affiche et présente le classement de sélectivité.

   Remarque : Toutes les colonnes actuellement indexées sont signalées par une icône sous l'onglet **Data View** (Vue des données) ou peuvent être consultées dans la table catalog.indexes.

Suppression d'index :

Si un index n'est plus utile, vous pouvez le supprimer de la table catalog.indexes.

Procédure

Pour supprimer des index de colonne :

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).
3. Pour ouvrir la table catalog.indexes et afficher tous les index, cliquez sur **indexes** (Index) puis sur l'onglet **Data View** (Vue des données). Chaque index est représenté sous forme d'une ligne de la table. Vous pouvez utiliser la colonne **IndexName** pour identifier l'index recherché.
4. Cliquez avec le bouton droit de la souris sur la ligne choisie, puis cliquez sur **Indexes** > **Delete Column Index** et cliquez sur **Yes** (Oui) pour confirmer la suppression de l'index.

Suppression de bases de données

Vous n'êtes pas autorisé à supprimer des bases de données système, à côté desquelles est placée un icône de verrouillage.

Pourquoi et quand exécuter cette tâche

Avertissement : Lorsque vous supprimez une base de données contenant des données de table, les tables sont en premier lieu vidées et supprimées.

Pour supprimer une base de données, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).
3. Sélectionnez la base de données à supprimer et cliquez sur **Drop Database** (Supprimer la base de données) dans la barre d'outils. La base de données est supprimée du serveur ObjectServer.

Suppression de tables

Vous n'êtes pas autorisé à supprimer les tables de bases de données système.

Pourquoi et quand exécuter cette tâche

Pour supprimer une table, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).
3. Sélectionnez la base de données contenant la colonne à supprimer.

4. Sélectionnez la table à supprimer et cliquez sur **Drop Table** (Supprimer la table) dans la barre d'outils. Toutes les données sont supprimées de la table, laquelle est ensuite supprimée de la base de données.

Suppression de colonnes d'une table

Vous n'êtes pas autorisé à supprimer les colonnes de clé primaire ou les colonnes de tables système.

Pourquoi et quand exécuter cette tâche

Avertissement : Lors de la suppression d'une colonne, il est nécessaire de réaliser un certain nombre d'actions préliminaires afin d'identifier et de supprimer les dépendances externes dont fait l'objet la colonne. Cela implique de rechercher toutes les références à la colonne dans les déclencheurs, les procédures, les vues, les filtres de restriction, les fichiers de règles de sonde et les fichiers de mappage de passerelle. De même, n'oubliez pas que si vous supprimez un colonne de laquelle dépendent des déclencheurs, des procédures, des vues ou des filtres de restriction, ces objets dépendants sont également supprimés, un avertissement étant écrit dans le fichier journal du serveur ObjectServer.

Pour supprimer une colonne de table après avoir confirmé l'absence de dépendance externe, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).
3. Sélectionnez la table contenant la colonne à supprimer.
4. Cliquez sur l'onglet **Column Definitions** (Définitions de colonne).
5. Sélectionnez la colonne à supprimer et cliquez sur **Drop Column** (Supprimer la colonne) dans la barre d'outils. La colonne est supprimée de la table.

Référence associée:

«DROP COLUMN», à la page 383
Pour supprimer des colonnes d'une table existante, utilisez le paramètre `DROP COLUMN` avec la commande `ALTER TABLE`.

Affichage et modification des propriétés du serveur ObjectServer

Les propriétés du serveur ObjectServer permettent de déterminer son comportement. Vous pour les afficher et les modifier à l'aide de Netcool/OMNIbus Administrator. Vous ne pouvez pas ajouter des propriétés, mais uniquement éditer celles qui existent.

Pourquoi et quand exécuter cette tâche

L'emplacement par défaut du fichier de propriétés ObjectServer est `$NCHOME/omnibus/etc/`*nom_serveur*.props. Le serveur ObjectServer lit le fichier de propriétés lorsqu'il démarre.

Important : Il est essentiel de bien connaître les propriétés du serveur ObjectServer avant de les modifier. Une configuration incorrecte peut avoir des conséquences néfastes sur les performances et les fonctionnalités du système.

Pour modifier la valeur d'une propriété du serveur ObjectServer, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Propriétés**. Le panneau ObjectServer Properties (Propriétés du serveur ObjectServer) s'ouvre.

   Conseil : Le texte `false` des propriétés en affichage seulement s'affiche dans la colonne **Editable**.
3. Pour éditer une propriété, sélectionnez-la, puis cliquez sur **Edit Property** (Editer la propriété) dans la barre d'outils. La fenêtre Property Details (Détails de la propriété) s'ouvre.
4. Complétez cette fenêtre comme suit :

   Nom Le nom unique attribué à la propriété ObjectServer est affiché ici. Vous ne pouvez pas changer ce nom.

   Description
   Une description de la propriété ObjectServer est affichée ici. Vous ne pouvez pas changer cette valeur.

   Valeur Modifiez la valeur de propriété selon les besoins.
5. Sauvegardez ou annulez les modifications comme suit :

   OK Cliquez sur ce bouton pour sauvegarder les détails de la propriété et fermer la fenêtre. La valeur mise à jour s'affiche dans le panneau Propriétés ObjectServer.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Résultats

Conseil : Les modifications apportées à certaines propriétés ne prennent pas effet tant que vous n'avez pas redémarré le serveur ObjectServer. La colonne **Immediate** (Immédiat) de ces propriétés comporte le texte `false`.

Référence associée:

«Propriétés et options de ligne de commande du serveur ObjectServer», à la page 4
Utilisez les propriétés ou les options de ligne de commande ObjectServer pour configurer les paramètres du serveur ObjectServer. Pour éviter les erreurs, ajoutez autant de propriétés que possible au fichier de propriétés plutôt que des options de ligne de commande. D'autres utilitaires sont fournis que vous pouvez utiliser pour chiffrer les valeurs de propriété.

Configuration des fichiers du serveur ObjectServer

Les fichiers du serveur ObjectServer sont des objets de stockage définis par l'utilisateur qui contiennent des données de journal ou de rapport.

Un fichier du serveur ObjectServer est un fichier logique possédant un fichier ou un ensemble de fichiers correspondant sur le système de fichiers physique. Vous pouvez définir les tailles de fichier du serveur ObjectServer et le nombre de fichiers physiques dans un ensemble.

Séquence de création des fichiers du serveur ObjectServer

Un numéro est ajouté au nom (ou extension) de chaque fichier d'un ensemble de fichiers.

Par exemple, si vous créez un fichier nommé `logfile` dans le répertoire `/log`, spécifiez sa taille maximale à 20 Ko et le nombre maximum de fichiers à 3, la séquence des fichiers suivante est créée et utilisée :

1. Lorsque vous cliquez sur **OK** pour créer le fichier, le serveur ObjectServer crée un fichier vide nommé `logfile1` dans le répertoire `/log`.
2. Le serveur ObjectServer écrit des données dans `logfile1` jusqu'au dépassement de la taille maximale du fichier (20 Ko).
3. Le serveur ObjectServer remplace le nom du fichier `logfile1` par `logfile2`. Il crée ensuite un nouveau fichier `logfile1` et y insère des données jusqu'au dépassement de la taille maximale.
4. Le serveur ObjectServer remplace le nom du fichier `logfile2` par `logfile3` et le nom du fichier `logfile1` par `logfile2`. Il crée ensuite un nouveau fichier `logfile1` et y insère des données jusqu'au dépassement de la taille maximale.
5. Le serveur ObjectServer supprime l'ancien fichier (`logfile3`). Il remplace ensuite le nom du fichier `logfile2` par `logfile3` et le nom du fichier `logfile1` par `logfile2`. Il crée ensuite un nouveau fichier nommé `logfile1` et y insère des données jusqu'au dépassement de la taille maximale.

Cette séquence est répétée jusqu'à la modification ou la suppression du fichier.

Création et édition de fichiers ObjectServer

Un fichier ObjectServer offre un moyen de consigner ou de renvoyer des informations relatives aux événements ObjectServer.

Pourquoi et quand exécuter cette tâche

Par exemple, vous pouvez créer un déclencheur qui écrit une entrée dans un fichier ObjectServer à chaque fois qu'un utilisateur se connecte à un serveur ObjectServer.

Pour créer ou éditer un fichier ObjectServer, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Log Files** (Fichiers journaux). Le panneau Log Files (Fichiers journaux) s'ouvre.
3. Pour ajouter un fichier, cliquez sur **Add Log File** (Ajouter un fichier journal) dans la barre d'outils. La fenêtre File Details (Détails du fichier) s'ouvre.

4. Pour éditer les détails du fichier, sélectionnez le fichier en question, puis cliquez sur **Edit Log File** (Editer le fichier journal) dans la barre d'outils. La fenêtre File Details (Détails du fichier) s'ouvre.
5. Complétez cette fenêtre comme suit :

Nom Entrez un nom de fichier ObjectServer unique ; par exemple, indiquez un nom expliquant son utilité. Remarque : il ne s'agit pas du nom de fichier créé sur le système de fichiers ; pour définir ce nom, renseignez la zone **Full File Path** (Chemin d'accès complet au fichier). Si vous modifiez un fichier, vous ne pouvez pas le renommer.

Conseil : Lors de la création d'objets du serveur ObjectServer, les noms des objets doivent commencer par une majuscule ou une minuscule suivie de majuscules ou de minuscules, de nombres ou de caractères de soulignement (_) et ne doivent pas dépasser 40 caractères. *Les noms d'utilisateur, de groupe et de rôle peuvent être composés de n'importe quelle chaîne de texte ne dépassant pas 64 caractères et peuvent inclure des espaces.* Les noms des objets du serveur ObjectServer sont sensibles à la casse.

Full File Path (Chemin d'accès complet au fichier)
Entrez le chemin complet et le nom du fichier physique ; par exemple, `/opt/netcool/omnibus/log/status.log`.

Remarque : Un numéro est automatiquement ajouté au nom de fichier dans le système de fichiers.

Activé Cochez cette case pour activer le fichier ObjectServer. S'il n'est pas activé, le fichier existera sur le système de fichiers, mais ne sera pas accessible en écriture. Vous pouvez ajouter des informations sur le fichier ObjectServer puis activer le fichier ultérieurement.

Unlimited File Size (Taille de fichier illimitée)
Cochez cette case si vous souhaitez écrire les informations dans un seul fichier de taille illimitée. Si vous activez ce paramètre, les zones **Max. Size** (Taille maximale) et **Max. Files** (Nombre maximum de fichiers) n'apparaîtront pas dans la fenêtre.

Si cette case n'est pas cochée, les informations peuvent être écrites dans un fichier unique, ou dans un pool de fichiers qui seront utilisés au fur et à mesure que leur taille maximale est atteinte. Si vous activez ce paramètre, vous devez spécifier les valeurs correspondantes dans les zones **Max. Size** (Taille maximale) et **Max. Files** (Nombre maximum de fichiers).

Truncate (Tronquer)
Cliquez sur ce bouton pour effacer les informations écrites dans le fichier physique. Le fichier ne sera pas supprimé. Dans les situations où un jeu contient plusieurs fichiers, seul le fichier en cours d'écriture sur le système de fichiers est tronqué.

Remarque : Ce bouton apparaît uniquement lorsque vous modifiez des détails de fichiers.

Max. Size (Taille max.)
Indiquez la taille maximale de fichier ObjectServer et sélectionnez une unité de mesure. La taille minimale de fichier est 1 Ko et la taille maximale est 4 Go.

Remarque : Il est possible que le système d'exploitation impose une limite inférieure pour la taille maximale de fichier unique.

Max. Files (Nombre maximum de fichiers)
Indiquez le nombre maximal de fichiers ObjectServer à créer.

6. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails de fichier et fermer la fenêtre. Les nouveaux détails de fichier sont ajoutés au panneau Log Files (Fichiers journaux).

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Concepts associés:

«Séquence de création des fichiers du serveur ObjectServer», à la page 160
Un numéro est ajouté au nom (ou extension) de chaque fichier d'un ensemble de fichiers.

Suppression de fichiers ObjectServer

Vous ne pouvez pas supprimer un fichier en cours d'utilisation (dans un déclencheur, par exemple).

Pourquoi et quand exécuter cette tâche

Pour supprimer un fichier ObjectServer, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Log Files** (Fichiers journaux). Le panneau Log Files (Fichiers journaux) s'ouvre.
3. Sélectionnez le fichier ObjectServer à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. Le fichier ObjectServer est supprimé. Le serveur ObjectServer n'écrit plus d'informations dans ce fichier.

Résultats

Lorsque vous supprimez un fichier, seul le fichier ObjectServer est supprimé. Les fichiers physiques créés dans le système de fichiers ne le sont pas.

Surveillance des connexions au serveur ObjectServer

Vous pouvez afficher toutes les connexions en cours au serveur ObjectServer et en déconnecter une ou plusieurs. Vous devez détenir le droit ALTER SYSTEM DROP CONNECTION pour pouvoir arrêter les connexions au serveur ObjectServer.

Pourquoi et quand exécuter cette tâche

Pour afficher et interrompre des connexions au serveur ObjectServer, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).

2. Cliquez sur **Connections** (Connexions). La fenêtre ObjectServer Connections (Connexions au serveur ObjectServer) s'ouvre. Elle présente une ligne d'informations pour chaque connectée.
3. Sélectionnez les lignes que vous souhaitez déconnecter. Vous pouvez utiliser la touche majuscule pour les sélections consécutives ou la touche Ctrl pour les sélections non consécutives.
4. Cliquez sur **Disconnect** (Déconnecter) dans la barre d'outils. Vous êtes invité à confirmer la déconnexion des applications que vous avez sélectionnées.
5. Cliquez sur **Yes** (Oui) pour chaque application à déconnecter, et sur **No** (Non) pour annuler la déconnexion.

Tâches associées:

«Affichage des connexions utilisateur au serveur ObjectServer», à la page 98
Vous pouvez afficher les détails de la connexion d'un utilisateur connecté au serveur ObjectServer.

Configuration des canaux

Le système de notification d'événement accéléré permet d'accélérer les événements à priorité élevée afin de permettre aux systèmes de s'exécuter sans interruption. Utilisez les canaux pour définir le type de données d'événement à intégrer aux notifications d'événement accéléré, ainsi que les destinataires de ces données.

Concepts associés:

Chapitre 6, «Configuration de la notification des événements accélérés», à la page 235
Vous pouvez configurer Tivoli Netcool/OMNIbus pour accélérer la notification des événements pouvant présenter un risque pour le système. Le système Accelerated Event Notification (AEN) permet d'accélérer les événements de priorité élevée afin de garantir une exécution des systèmes sans interruption.

«Configuration des canaux afin de diffuser les données d'événement», à la page 238
Lors de la configuration des notifications d'événement accéléré, vous devez utiliser les canaux pour définir le type de données d'événement à diffuser dans les notifications d'événement accéléré et les destinataires de ces données. Vous pouvez configurer plusieurs canaux possédant des données d'événement et des destinataires variés.

Tâches associées:

«Création et édition de canaux», à la page 238
Vous devez créer des canaux sur un serveur ObjectServer à partir duquel les événements accélérés vont être transférés.

«Copie et collage de canaux», à la page 242
Vous pouvez utiliser un canal comme modèle pour un autre en copiant et collant sa définition. Cela est très utile si vous souhaitez créer des canaux avec de légères variations de leurs définitions.

«Suppression d'un canal», à la page 242

«Envoi de messages aux destinataires de canal», à la page 242
Vous pouvez envoyer des messages aux destinataires du canal qui exécutent le client de notification d'événement accéléré.

«Déconnexion des clients de notification d'événement accéléré», à la page 243
Si vous souhaitez procéder à des opérations de maintenance mineures sur le serveur ObjectServer (une resynchronisation, par exemple), vous pouvez déconnecter (ou fermer la session) à distance les clients de notification d'événement accéléré en cours d'exécution. Lors de la déconnexion, vous pouvez entrer un bref

message contenant les informations pertinentes à l'intention des utilisateurs.

«Arrêt des clients de notification d'événement accéléré», à la page 244
Si vous souhaitez arrêter le serveur ObjectServer, vous pouvez arrêter à distance les clients de notification d'événement accéléré en cours d'exécution. Lors de la procédure d'arrêt, vous pouvez entrer un bref message contenant les informations pertinentes à l'intention des utilisateurs.

Utilisation de l'interface interactive SQL en mode interface graphique

Vous pouvez utiliser l'interface interactive SQL pour configurer le serveur ObjectServer en émettant des commandes SQL.

Pourquoi et quand exécuter cette tâche

Remarque : Seuls les utilisateurs appartenant à un groupe auquel a été accordé le rôle ISQL peuvent accéder à un serveur ObjectServer à l'aide de l'interface interactive SQL. Seuls les utilisateurs appartenant à un groupe auquel a été accordé le rôle ISQLWrite peuvent mettre à jour les données du serveur ObjectServer à l'aide de l'interface interactive SQL.

Pour ouvrir l'interface interactive SQL en mode interface graphique, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **SQL**. Le panneau SQL s'ouvre.
3. Complétez cette fenêtre comme suit :

   Editeur SQL

   Utilisez les boutons et la zone de texte de cette zone pour émettre des commandes. Entrez le langage SQL dans la zone de texte, et utilisez un point virgule pour séparer plusieurs commandes. Les boutons de l'assistant SQL et les boutons supplémentaires facilitent la création des commandes SQL.

   Lorsque vous entrez des commandes SQL dans les panneaux d'édition SQL de Tivoli Netcool/OMNIbus, vous pouvez saisir un ou plusieurs caractères et appuyer sur Ctrl+F1 pour ouvrir une boîte de dialogue contenant une liste de mots clés susceptibles de correspondre à votre entrée. Sélectionnez le mot clé requis, puis cliquez sur **OK** pour compléter votre entrée. Si un seul mot clé correspond aux caractères que vous avez entrés, il est automatiquement inséré. Si vous appuyez sur Ctrl+F1 après avoir entré un mot clé associé à une base de données, la boîte de dialogue propose une liste de bases de données ObjectServer possibles dans laquelle vous pouvez faire votre sélection. Si vous appuyez sur Ctrl+F1 après avoir entré un nom de base de données suivi d'un point (`alerts.`, par exemple), vous pouvez de nouveau appuyer sur Ctrl+F1 pour afficher une liste de tables dans la base de données et faire votre sélection.

   La table suivante décrit les boutons de l'assistant.

Tableau 26. Boutons de l'interface interactive SQL

Bouton	Description
	Cliquez sur ce bouton pour sélectionner une commande SQL dans le menu contextuel. Selon la commande que vous avez sélectionnée, renseignez la fenêtre comme suit : • **Select** (SELECT) : sélectionnez la base de données et la table sur laquelle exécuter la commande SELECT. Ensuite, choisissez les colonnes de table à sélectionner. • **Insert** (INSERT) : sélectionnez la base de données et la table sur laquelle exécuter la commande INSERT. Ensuite, sélectionnez les colonnes de table dans lesquelles insérer les valeurs. Pour chaque colonne sélectionnée, entrez la valeur à insérer. Pour les instructions INSERT, vous devez inclure la clé primaire. Les clés primaires sont indiquées par un astérisque (*). • **Update** (UPDATE) : sélectionnez la base de données et la table sur lesquelles exécuter la commande. Ensuite, choisissez les colonnes de table à mettre à jour. Pour chaque colonne sélectionnée, entrez la nouvelle valeur. Pour les instructions UPDATE, vous devez exclure la clé primaire. Les clés primaires sont indiquées par un astérisque (*). **Remarque :** Pour les insertions et mises à jour de la table alerts.status, toutes les conversions existantes apparaissent dans les listes déroulantes. • **Delete** (Supprimer) : sélectionnez la table à supprimer. • **Use** (Utiliser) : sélectionnez la table à utiliser. • **Service** : sélectionnez un nom de service et une valeur. Les valeurs peuvent être `Good` (Correct), `Marginal` ou `Bad` (Incorrect).
	Cliquez sur ce bouton pour sélectionner un nom de colonne de table à ajouter à la commande. Le nom de colonne est remplacé par la valeur de la ligne correspondante de la liste d'événements lors de l'exécution de l'outil. Lorsque le nom de colonne est précédé du symbole @, il est remplacé par la valeur de ligne correspondante de la liste d'événements lors de l'exécution. Cela peut être utilisé dans une requête SQL ou un fichier de restriction (`RemoteNodeAlias = '@LocalNodeAlias'`, par exemple)
	Cliquez sur ce bouton pour procéder à une sélection dans une liste de conversions disponibles. Cliquez deux fois pour ajouter la conversion.
	Cliquez sur ce bouton pour effacer la commande SQL entrée.
	Cliquez sur ce bouton pour concevoir une liste de mots clés qui complète l'instruction SQL entrée.
	Cliquez sur ce bouton pour vérifier la validité de la syntaxe SQL entrée.
	Cliquez sur ce bouton pour rechercher un fichier du type `.sql` ou `.ed` et vérifier la validité de sa syntaxe. A l'issue, les résultats s'affichent (si vous utilisez un éditeur externe pour créer ou éditer des déclencheurs et procédures, ces derniers sont sauvegardés dans des fichiers `.ed`).
	Cliquez sur ce bouton pour soumettre les commandes SQL.

Après avoir défini la commande SQL, cliquez sur **Soumettre**.

Historique
Cette liste déroulante fournit l'historique des commandes SQL saisies. Vous pouvez sélectionner une commande émise précédemment dans la liste. Vous pouvez également effacer la liste des commandes émises précédemment en cliquant avec le bouton droit sur la liste et en sélectionnant **Clear History** (Effacer l'historique).

Result View (Vue Résultats)
Après avoir émis la commande, une représentation visuelle de la table sur laquelle vous exécutez la commande SQL s'affiche dans cet onglet.

Console View (Vue Console)
Un historique de commande s'affiche dans cet onglet.

Concepts associés:

Chapitre 5, «SQL ObjectServer», à la page 167
Le serveur ObjectServer possède une interface SQL permettant de définir et de manipuler les objets de base de données relationnelle tels que les tables et les vues.

Tâches associées:

«Création d'index», à la page 155
Utilisez Netcool/OMNIbus Administrator pour créer des index sur les colonnes des tables de base de données.

«Création et édition des conversions», à la page 143
Les conversions sont associées aux colonnes de la table alerts.status du serveur ObjectServer et mettent en correspondance les valeurs entières des colonnes avec les chaînes. Les conversions configurées dans Netcool/OMNIbus Administrator sont utilisées dans la liste d'événements afin de convertir les valeurs entières en chaîne, pour des raisons de lisibilité.

Chapitre 5. SQL ObjectServer

Le serveur ObjectServer possède une interface SQL permettant de définir et de manipuler les objets de base de données relationnelle tels que les tables et les vues.

Les commandes SQL ObjectServer sont les suivantes :

- Commandes Data Definition Language (DDL) pour créer, modifier et supprimer les objets de base de données
- Commandes Data Manipulation Language (DML) pour interroger et manipuler les données dans les objets de base de données existants
- Commandes système pour modifier la configuration d'un serveur ObjectServer
- Commandes de contrôle de session pour modifier les paramètres dans les sessions clientes
- Commandes de sécurité pour contrôler l'accès des utilisateurs aux objets de base de données

Le serveur ObjectServer possède également des commandes de langage de procédure qui permettent de programmer les constructions pour définir des actions survenant lors de l'occurrence d'incidents spécifiés et de conditions définies par vos soins. Vous pouvez utiliser les procédures et les déclencheurs pour former des automatisations et ainsi traiter automatiquement les événements.

Vous pouvez utiliser l'interface interactive SQL pour vous connecter à un serveur ObjectServer et exécuter les commandes SQL de serveur ObjectServer.

Conseil : De nombreuses tâches exécutées via les commandes SQL de serveur ObjectServer à partir de l'interface interactive SQL peuvent également être effectuées à partir de l'interface Netcool/OMNIbus Administrator.

Tâches associées:

«Utilisation de l'interface interactive SQL en mode interface graphique», à la page 164
Vous pouvez utiliser l'interface interactive SQL pour configurer le serveur ObjectServer en émettant des commandes SQL.

Objets du serveur ObjectServer

Le serveur ObjectServer stocke, gère et traite les données d'événement collectées par des applications externes telles que des sondes et des passerelles. Les structures (ou objets) de stockage par défaut sont créées par des fichiers de définition SQL. Vous pouvez créer de nouveaux objets ou modifier des objets existants à l'aide de l'interface interactive SQL.

Les principaux objets ObjectServer sont les bases de données, les tables, les index, les vues, les filtres de restriction et les fichiers.

Bases de données
Les bases de données ObjectServer sont des bases de données relationnelles qui stockent les données dans des tables. Lorsque vous initialisez un serveur ObjectServer, plusieurs bases de données par défaut sont créées.

Tables Les tables possèdent un nombre fixe de colonnes de type données. Chaque

© Copyright IBM Corp. 1994, 2014

nom de colonne est unique dans la table. Les tables ne contiennent aucune ligne de données ou plusieurs lignes de données au format défini par la liste de colonnes de la table.

Le nom de table qualifié complet est composé du nom de la base de données et du nom de la table, séparés par un point. Par exemple, la table status de la base de données des alertes (alerts) est nommée par alerts.status.

Index Vous pouvez utiliser des index pour améliorer les performances des bases de données ObjectServer. L'utilisation d'index correctement conçus permet de réduire ou d'éliminer le recours aux analyses complètes de table pendant l'exécution de requêtes SQL et accélère l'extraction des données.

Vues Les vues sont des tables virtuelles projetées à partir de lignes et de colonnes particulières de tables réelles. Elles permettent d'afficher et de manipuler les sous-ensembles de données de table.

Par exemple, vous pouvez limiter un groupe d'utilisateurs afin qu'ils ne puissent afficher qu'une vue d'une table au lieu de toutes les colonnes de la table. Vous pouvez également disposer de colonnes virtuelles, composées à l'aide d'expressions provenant des colonnes de la table sous-jacente.

Remarque : Les vues sont principalement destinées à un usage interne. N'utilisez pas de vues dans les automatisations.

Filtres de restriction
Les filtres de restriction permettent de limiter les lignes affichées lorsqu'un utilisateur affiche une table.

Une fois le filtre de restriction affecté à un utilisateur ou à un groupe, il contrôle les données pouvant être affichées ou modifiées à partir des applications client, ainsi que les données pouvant être modifiées dans les instructions `INSERT`, `UPDATE` et `DELETE`. Seules les lignes conformes aux critères spécifiés dans les conditions du filtre de restriction sont retournées. Les filtres de restriction sont automatiquement appliqués aux commandes `SELECT`, `INSERT`, `UPDATE` et `DELETE`.

Vous ne pouvez affecter à un utilisateur ou à un groupe qu'un filtre de restriction par table. Cependant, plusieurs filtres de restriction peuvent s'appliquer à un utilisateur. Par exemple, vous pouvez affecter un filtre de restriction unique à un utilisateur mais également affecter plusieurs filtres aux groupes auxquels cet utilisateur appartient. Si vous affectez un filtre à un utilisateur et un filtre à chacun des trois groupes auxquels il appartient, c'est le total des quatre filtres qui s'applique à l'utilisateur. Si plusieurs filtres de restriction s'appliquent à un utilisateur, les données obtenues sont une combinaison de tous les filtres de restriction applicables pour l'utilisateur ou le groupe.

Si vous utilisez plusieurs filtres de restriction, indiquez la valeur appropriée pour la propriété ObjectServer **`RestrictionFiltersAND`**.

Files (Nombre maximum de fichiers)
Les fichiers du serveur ObjectServer sont des objets de stockage définis par l'utilisateur pour les données de journal ou de rapport. Un fichier du serveur ObjectServer est un fichier logique possédant un fichier ou un ensemble de fichiers correspondants sur le système de fichiers physique. Vous pouvez définir les tailles de fichier du serveur ObjectServer et le nombre de fichiers physiques dans un ensemble.

Héritage de droits d'accès d'un objet

Lorsqu'un objet ObjectServer est créé, les droits d'accès lui sont automatiquement accordés en fonction de ceux de son parent.

Par exemple, si SuperAdmin détient les droits d'accès `CREATE_DATABASE` et que LondonAdmin crée une base de données, SuperAdmin détient par défaut tous les droits d'accès d'objet sur la base de données que LondonAdmin a créée.

Si les droits d'accès sur un objet parent sont modifiés après la création de l'objet enfant, cela n'a aucun effet sur les droits d'accès de l'objet enfant.

Le tableau suivant répertorie les objets enfant pour chaque objet parent ObjectServer.

Tableau 27. Héritage de droits d'accès d'un objet

Objet parent	Objets enfant
Système	Database Groupe de déclencheurs Fichier Procédure SQL Procédure externe Signal Filtre de restriction
Base de données	Table Vue
Table	Index
Groupe de déclencheurs	Déclencheur

Bases de données initialisées par le système

Lorsque vous initialisez un serveur ObjectServer, plusieurs bases de données par défaut sont créées.

La table suivante décrit ces bases de données initialisées par le système.

Tableau 28. Bases de données initialisées par le système

Nom de base de données	Type de base de données	Description
security	Système	Contient des informations sur la sécurité du système, notamment les utilisateurs, les rôles, les groupes et les droits d'accès.
catalog	Système	Contient des métadonnées sur les objets du serveur ObjectServer.
alerts	Utilisateur	Contient des informations sur le statut de l'alerte, transmises au serveur ObjectServer par des sondes et des passerelles.

Tableau 28. Bases de données initialisées par le système (suite)

Nom de base de données	Type de base de données	Description
service	Utilisateur	Utilisée pour prendre en charge IBM Tivoli Composite Application Manager for Internet Service Monitoring.
custom	Utilisateur	Utilisée pour les tables ajoutées par les utilisateurs.
persist	Système	Enregistre les informations internes sur l'état du serveur ObjectServer.
transfer	Système	Utilisée en interne par les passerelles unidirectionnelles et bidirectionnelles du serveur ObjectServer pour synchroniser les informations de sécurité entre les serveurs ObjectServer.
master	Utilisateur	Utilisée à des fins de compatibilité avec les versions précédentes de Tivoli Netcool/OMNIbus. Les tables de la base de données maître prennent également en charge l'architecture du serveur ObjectServer de bureau. Pour plus d'informations sur l'architecture du serveur ObjectServer de bureau, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.
tools	Utilisateur	Utilisée à des fins de compatibilité avec les versions précédentes de Tivoli Netcool/OMNIbus.
registry	Utilisateur	Contient des informations sur les configurations réparties Tivoli Netcool/OMNIbus.
iduc_system	Utilisateur	Contient toutes les tables de prise en charge d'application IDUC requises pour accélérer la notification des événements, l'envoi de messages d'informations et l'appel de commandes.
precision	Utilisateur	Utilisée par IBM Tivoli Network Manager IP Edition pour implémenter l'application d'événements affectés par le service.

Restriction : Le serveur ObjectServer gère les bases de données du système. Vous pouvez visualiser, mais pas modifier, les données qui s'y trouvent.

Concepts associés:

«Configuration des bases de données», à la page 148
La configuration des bases de données implique la création et la maintenance des tables de base de données et des colonnes de ces tables.

Référence associée:

Annexe D, «Tables du serveur ObjectServer», à la page 415
La base de données du serveur ObjectServer contient les tables suivantes : les tables d'alertes, les tables de services, les tables de catalogues système, les tables de statistiques, les tables de prise en charge d'outils client, les tables d'outils de bureau, les tables du serveur ObjectServer de bureau, les tables de sécurité, les tables de canaux IDUC et les tables d'événements affectés par un service.

Tables système

Les tables système sont des tables spéciales gérées par le serveur ObjectServer et contiennent des métadonnées sur les objets du serveur ObjectServer.

Les tables système sont identifiées par le nom de la base de données `catalog`. Par exemple, la table catalog.columns contient les métadonnées de toutes les colonnes des tables du serveur ObjectServer.

Vous pouvez afficher les informations dans les tables système. Pour ce faire, utilisez les commandes SELECT et DESCRIBE, mais vous ne pouvez ni ajouter, ni modifier, ni supprimer les tables système ou leur contenu à l'aide du langage SQL du serveur ObjectServer.

Référence associée:

«DESCRIBE», à la page 382
La commande `DESCRIBE` permet d'afficher des informations sur les colonnes de la table ou de la vue spécifiée.

«SELECT (scalaire)», à la page 401
La commande scalaire `SELECT` extrait les colonnes et lignes d'une table en fonction de critères spécifiques.

«SELECT (aggregate)», à la page 403
Une commande `SELECT` d'agrégation calcule le nombre de lignes et renvoie une valeur unique.

«SELECT (GROUP BY)», à la page 405
Vous pouvez utiliser une commande `SELECT` avec une clause `GROUP BY` pour regrouper en une seule ligne toutes les lignes ayant des valeurs identiques dans une colonne ou une combinaison de colonnes spécifiques. Vous pouvez également rechercher la valeur agrégée de chaque groupe de valeurs de colonne.

«Tables de catalogues système», à la page 428
La base de données de catalogue contient les tables système créées et gérées par le serveur ObjectServer. Les tables système contiennent des métadonnées sur les objets du serveur ObjectServer.

Spécification des types de données des colonnes

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.

Lors de la création d'une table à l'aide de la commande CREATE TABLE, vous devez spécifier un type de données pour chaque colonne que vous définissez.

Important : Vous pouvez n'afficher que les colonnes de type CHAR, VARCHAR, INCR, INTEGER et TIME dans la liste d'événements. N'ajoutez pas les colonnes d'un autre type dans la table alerts.status. Si vous ajoutez des colonnes qui ne sont

pas de type CHAR, VARCHAR, INCR, INTEGER ou TIME, les sondes ne peuvent pas écrire dans la table alerts.status. Il en résulte alors que les sondes ne peuvent pas démarrer.

Les types de données pris en charge par le serveur ObjectServer figurent dans le tableau ci-dessous.

Tableau 29. 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 uniquement aux colonnes de table. Les colonnes de ce type sont définies par le système. Vous pouvez remplacer l'incrémentation automatique par une autre valeur en utilisant la commande ALTER TABLE SET INCR. Vous pouvez aussi utiliser cette commande pour définir quelles sont les valeurs affectées aux nouvelles lignes. Vous ne pouvez pas modifier les colonnes INCR dans les lignes existantes.	`1`	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 29. Types de données ObjectServer (suite)

Type SQL	Description	Valeur par défaut	ID du serveur ObjectServer correspondant au type de données
CHAR(*entier*)	Chaîne de caractères à taille fixe, composée de *entier* 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(*entier*)	Chaîne de caractères à taille variable, composée de *entier* 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, d'analyse, d'insertion et de suppression.	''	2
INTEGER64	Entier signé 64 bits.	0	16
UNSIGNED64	Entier non signé 64 bits.	0	17

Référence associée:

«CREATE TABLE», à la page 372
La commande CREATE TABLE permet de créer une table.

«Table alerts.status», à la page 415
La table alerts.status contient des informations de statut sur les problèmes détectés par les sondes.

«ALTER TABLE», à la page 357
La commande ALTER TABLE permet de modifier les caractéristiques d'une table existante et de ses colonnes. Vous pouvez ajouter, supprimer et modifier des colonnes ; vous pouvez également définir manuellement la valeur des zones de type INCR.

«CREATE SIGNAL», à la page 371
La commande CREATE SIGNAL permet de créer un signal défini par l'utilisateur. Il s'agit de définir une liste d'attributs de type données.

Spécification des propriétés facultatives des colonnes

Vous pouvez spécifier les propriétés facultatives des colonnes que vous définissez lors de la création d'une table.

Les propriétés de colonne facultatives sont décrites dans le tableau suivant.

Tableau 30. Propriétés de colonne

Propriété de colonne	Description
PRIMARY KEY	La colonne est créée en tant que clé primaire. Les colonnes de clé primaire identifient chaque ligne de manière unique. Elles peuvent contenir une valeur par défaut et ne peuvent pas être masquées.
NODEFAULT	La valeur de cette colonne doit être spécifiée dans la commande INSERT initiale. Vous pouvez utiliser cette commande pour insérer une nouvelle ligne de données dans une table existante.
NOMODIFY	La valeur de cette colonne ne peut pas être modifiée après l'exécution de la commande INSERT initiale.
HIDDEN	Les données ne sont pas écrites ni lues dans une colonne masquée lors de l'insertion ou de la sélection d'une ligne. Le nom de colonne doit être indiqué de manière explicite pour y insérer les données ou les sélectionner. Les colonnes masquées contiennent des informations sur le système ou des informations qui ne sont pas applicables à la plupart des utilisateurs.

Dans la commande CREATE TABLE, la syntaxe permettant de spécifier les colonnes faisant office de clé primaire se présente comme suit :

```
(nom_colonne type_données [ PRIMARY KEY | NODEFAULT | NOMODIFY | HIDDEN ],...
[, PRIMARY KEY(nom_colonne,...) ] );
```

En fonction de cette syntaxe, vous pouvez créer des colonnes faisant office de clés primaires de l'une et/ou l'autre des manières suivantes :

- Spécifiez la propriété de colonne PRIMARY KEY dans le cadre d'une définition de colonne.
- Spécifiez une ou plusieurs colonnes composant la clé primaire en insérant une liste séparée par des virgules de colonnes dans une clause PRIMARY KEY à la suite des définitions de colonne.

Référence associée:

«CREATE TABLE», à la page 372
La commande CREATE TABLE permet de créer une table.

«INSERT», à la page 397
La commande INSERT permet d'insérer une nouvelle ligne de données dans une table existante.

Affichage des détails de l'index

Pour connaître les colonnes qui ont été indexées, vous pouvez examiner le contenu de la table catalog.indexes dans l'interface Netcool/OMNIbus Administrator ou à l'aide de la commande SELECT.

Pourquoi et quand exécuter cette tâche

Pour afficher les détails de l'index dans la table catalog.indexes, suivez l'une des étapes suivantes :

Procédure

- Dans la fenêtre Netcool/OMNIbus Administrator :
  1. Sélectionnez le bouton de menu **System** (Système).
  2. Cliquez sur **Databases** (Bases de données). Le panneau Databases, Tables and Columns (Bases de données, tables et colonnes).
  3. Sélectionnez **catalog.indexes**.
  4. Cliquez sur l'onglet **Data View** (Vue des données) du panneau Databases, Tables and Columns (Bases de données, tables et colonnes) pour afficher les données de la table.
- Dans l'interface interactive SQL, entrez la commande suivante :

```
select * from catalog.indexes;
```

Concepts associés:

«interface interactive SQL», à la page 229
Vous pouvez utiliser l'utilitaire de ligne de commande de l'interface interactive SQL fourni avec Tivoli Netcool/OMNIbus pour émettre des commandes SQL en direction d'un serveur ObjectServer.

Tâches associées:

«Création d'index», à la page 155
Utilisez Netcool/OMNIbus Administrator pour créer des index sur les colonnes des tables de base de données.

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Référence associée:

«Table catalog.indexes», à la page 439
La table catalog.indexes stocke des informations sur les index, y compris la colonne et la table de base de données sur lesquelles l'index se base, et le type d'index.

«SELECT (scalaire)», à la page 401
La commande scalaire SELECT extrait les colonnes et lignes d'une table en fonction de critères spécifiques.

«SELECT (aggregate)», à la page 403
Une commande SELECT d'agrégation calcule le nombre de lignes et renvoie une valeur unique.

«SELECT (GROUP BY)», à la page 405
Vous pouvez utiliser une commande SELECT avec une clause GROUP BY pour regrouper en une seule ligne toutes les lignes ayant des valeurs identiques dans une colonne ou une combinaison de colonnes spécifiques. Vous pouvez également rechercher la valeur agrégée de chaque groupe de valeurs de colonne.

Automatisations ObjectServer

Vous pouvez créer des déclencheurs et procédures SQL afin d'automatiser les tâches ObjectServer. Un ensemble d'automatisations standard est inclus dans Tivoli Netcool/OMNIbus. Ces automatisations sont créées lors de l'initialisation de la base de données.

Les principaux objets d'automatisation ObjectServer sont les groupes de déclencheurs, les déclencheurs, les signaux et les procédures.

Groupes de déclencheurs
Les groupes de déclencheurs sont des collections de déclencheurs connexes. Tout déclencheur doit appartenir à un groupe de déclencheurs.

Déclencheurs de base de données
Les déclencheurs de base de données sont mis en application en cas de tentative ou d'exécution d'une modification sur une table ou une vue ObjectServer.

Déclencheurs temporels
Les déclencheurs temporels sont mis en application à une fréquence spécifiée.

Déclencheurs de signal
Les déclencheurs de signal sont mis en application en réponse aux incidents survenant dans le serveur ObjectServer ou à des signaux définis par l'utilisateur.

Signaux définis par l'utilisateur
Vous pouvez créer des signaux définis par l'utilisateur afin de les utiliser dans vos propres automatisations.

Signaux système
Les signaux système sont des signaux prédéfinis pouvant être utilisés avec les déclencheurs de signal.

Procédures SQL
Les procédures SQL permettent de manipuler les données dans une base de données ObjectServer.

Procédures externes
Les procédures externes exécutent des fichiers exécutables sur un système distant.

Déclencheurs

Vous pouvez utiliser les déclencheurs pour détecter les modifications du serveur ObjectServer et exécuter des réponses automatiques à ces modifications. Vous permettez ainsi au serveur ObjectServer de traiter des alertes sans qu'un opérateur n'aie besoin d'intervenir. Vous pouvez également utiliser l'automatisation pour gérer le dédoublonnage, qui permet de réduire la quantité de données traitées dans le serveur ObjectServer en éliminant les événements en double.

Vous pouvez utiliser les déclencheurs de base de données, les déclencheurs temporels et les déclencheurs de signal dans les automatisations :

- Les déclencheurs de base de données s'activent si l'une des modifications suivantes est apportée à la base de données :
  - Tentative d'insertion d'une ligne dans une table.
  - Tentative de mise à jour d'une ligne dans une table.

  - Tentative de suppression d'une ligne dans une table.
  - Une tentative d'insertion d'une ligne dans une table a été effectuée, mais une ligne possédant la même valeur pour la clé primaire d'identificateur existe déjà. Vous pouvez utiliser un déclencheur de *réinsertion* pour dupliquer plusieurs fois les lignes dans le serveur ObjectServer.

    Vous pouvez créer votre propre déclencheur de duplication multiple pour générer une action différente.
- Les déclencheurs temporels sont activés de façon répétitive en fonction d'une fréquence spécifiée.

  Par exemple, vous pouvez utiliser un déclencheur temporel pour supprimer toutes les lignes claires (`Severity = 0`) de la table alerts.status qui n'ont pas été modifiées pendant une certaine période temporelle.
- Les déclencheurs de signal sont activés lorsqu'un signal système prédéfini est émis ou lorsqu'un signal défini par l'utilisateur est émis via la commande `RAISE SIGNAL`.

  Par exemple, vous pouvez envoyer un courrier électronique à un opérateur au démarrage ou à l'arrêt du serveur ObjectServer car des signaux système sont générés.

  Vous n'avez rien à faire pour créer ou configurer les signaux de système. Vous devez créer, mettre au premier plan et supprimer de façon explicite les signaux définis par l'utilisateur.

Exécution de commandes dans les actions du déclencheur

La variable *`action_déclencheur`* contient un ensemble de commandes permettant de manipuler les données dans le serveur ObjectServer.

Vous pouvez exécuter les commandes SQL suivantes dans un déclencheur :

```
ALTER FILE
ALTER SYSTEM BACKUP
ALTER SYSTEM DROP CONNECTION
ALTER SYSTEM SET
ALTER TRIGGER
ALTER TRIGGER GROUP
ALTER USER
UPDATE
INSERT
DELETE
WRITE INTO
RAISE SIGNAL
{ EXECUTE | CALL } PROCEDURE
```

L'utilisateur qui crée le déclencheur doit détenir les droits d'accès appropriés pour exécuter les commandes dans le corps du déclencheur.

Avertissement : Les déclencheurs ou procédures ne peuvent pas comporter de dépendances circulaires. Par exemple, vous ne devez pas créer de déclencheur qui appelle une procédure qui, elle-même, met en application le déclencheur d'origine.

Vous pouvez utiliser les constructions de programmation suivantes dans un déclencheur :

- Instruction d'affectation SET
- Instruction IF THEN ELSE
- Instruction CASE WHEN
- Boucle FOR EACH ROW
- Boucle FOR

Référence associée:

«CASE WHEN», à la page 361
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser l'instruction `CASE WHEN` pour effectuer une ou plusieurs actions basées sur une condition. Si la condition n'est pas remplie, vous pouvez, si vous le souhaitez, effectuer une action différente.

«IF THEN ELSE», à la page 395
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser l'instruction `IF THEN ELSE` pour réaliser une ou plusieurs actions en fonction des conditions spécifiées.

«FOR», à la page 390
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser la boucle `FOR` pour réaliser des actions un certain nombre de fois, en fonction de la variable de comptage.

«FOR EACH ROW», à la page 390
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser la boucle `FOR EACH ROW` pour réaliser des actions sur un ensemble de lignes répondant à une certaine condition.

«SET», à la page 406
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser une instruction d'affectation `SET` pour écrire la valeur d'une expression dans une variable ou un paramètre.

«CREATE TRIGGER (déclencheur de base de données)», à la page 373
La commande `CREATE TRIGGER` permet de créer les déclencheurs de base de données qui s'activent lorsqu'une modification ou une tentative de modification dans une table du serveur ObjectServer survient (ou lorsqu'une modification ou une tentative de modification dans une vue affecte une table de base).

Utilisation des variables de déclencheur dans les conditions et actions du déclencheur

Vous pouvez utiliser des variables de déclencheur pour accéder aux informations relatives aux exécutions en cours et précédentes du déclencheur. Utilisez la notation `%trigger` pour spécifier les variables de variables de déclencheur. Le symbole `%` indique que vous faites référence à une variable implicite. Le mot clé `trigger` fait référence au déclencheur en cours.

Par exemple, pour faire référence au nombre de lignes du déclencheur précédent, utilisez la syntaxe suivante :

```
%trigger.previous_rowcount
```

Conseil : Vous pouvez également utiliser le bouton d'aide `%` dans l'éditeur de déclencheur de l'interface graphique Tivoli Netcool/OMNIbus Administrator pour sélectionner les variables `%trigger`.

Le tableau ci-dessous répertorie les attributs en lecture seule disponibles dans la clause WHEN et la section des actions d'un déclencheur.

Tableau 31. Variables de déclencheur implicites

Attribut du déclencheur	Type de données	Description
`%trigger.previous_condition`	BOOLEAN	Valeur de la condition lors de la dernière exécution.

Tableau 31. Variables de déclencheur implicites (suite)

Attribut du déclencheur	Type de données	Description
`%trigger.previous_rowcount`	UNSIGNED	Nombre de lignes renvoyées par la clause EVALUATE lors de la dernière utilisation du déclencheur.
`%trigger.num_positive_rowcount`	UNSIGNED	Nombre de mises en application consécutives avec une ou plusieurs correspondances dans la clause EVALUATE.
`%trigger.num_zero_rowcount`	UNSIGNED	Nombre de mises en application consécutives sans correspondance dans la clause EVALUATE.
`%rowcount`	UNSIGNED	Nombre de lignes correspondant à la clause EVALUATE lors de la mise en application d'un déclencheur temporel. **Remarque :** Cette variable ne requiert pas le préfixe de mot clé `trigger`. Sa valeur est TRUE uniquement si elle est considérée comme la première action réalisée dans le corps du déclencheur. Ensuite, la valeur est indéterminée.

Remarque : Dans un déclencheur de base de données, la seule variable de déclencheur valide est `%trigger.previous_condition`. Toutes les autres variables de déclencheur donne le résultat d'une clause EVALUATE, qui n'est pas prise en charge pour les déclencheurs de base de données.

Exemple

Ce déclencheur de signal système consigne le nom de chaque utilisateur qui connecte le serveur ObjectServer à un fichier.

```
CREATE TRIGGER LogConnections
 GROUP default_triggers
 PRIORITY 1
 ON SIGNAL connect
BEGIN
 WRITE INTO file1 VALUES ('User', %user.user_name, 'has logged on.');
END;
```

Référence associée:

Annexe C, «Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires», à la page 411
Vous pouvez utiliser un certain nombre de commandes SQL, d'expressions variables et de boutons d'aide pour récupérer des informations d'une liste d'événements en cours, de l'événement actuel ou de l'environnement du système d'exploitation. Vous pouvez utiliser ces expressions lors de la création d'un outil, d'un déclencheur ou d'une procédure SQL, ainsi que dans les paramètres transmis à une liste d'événements transitoires.

«Variables utilisateur implicites dans les procédures et les déclencheurs», à la page 195
Vous pouvez utiliser des variables utilisateur pour accéder aux informations relatives aux utilisateurs connectés dans une expression SQL dans le corps d'un déclencheur ou d'une procédure.

Variables implicites NEW et OLD des déclencheurs au niveau de la ligne

Outre les variables locales déclarées dans le déclencheur, les déclencheurs au niveau de la ligne ont accès à des variables implicites dont les valeurs sont automatiquement définies par le système.

La variable `OLD` se rapporte à la valeur d'une colonne avant l'occurrence de l'incident. La variable `NEW` se rapporte à une colonne affectée par l'occurrence de l'incident. Vous pouvez utiliser des expressions pour lire et attribuer des valeurs à des variables de ligne.

Certaines opérations réalisées sur la variable de ligne `NEW` ou `OLD` peuvent ne pas être accessibles ou modifiables selon le type de modification. Par exemple, si le serveur ObjectServer supprime une ligne, il n'existe aucune ligne `NEW` à lire ou modifier.

Le tableau ci-dessous illustre la disponibilité des variables `NEW` et `OLD` en fonction de l'opération de base de données.

Tableau 32. Disponibilités des variables de lignes particulières

Opération	Mode horodatage	La variable NEW est-elle disponible ?	La variable NEW est-elle modifiable ?	La variable OLD est-elle disponible ?	La variable OLD est-elle modifiable ?
INSERT	BEFORE	Y	Y	N	N
INSERT	AFTER	Y	N	N	N
UPDATE	BEFORE	Y	Y	Y	N
UPDATE	AFTER	Y	N	N	N
DELETE	BEFORE	N	N	Y	N
DELETE	AFTER	N	N	Y	N
REINSERT	BEFORE	Y	N	Y	Y
REINSERT	AFTER	Y	N	N	N

Remarque : Dans un déclencheur postérieur à l'insertion, seule la variable NEW est disponible. Elle représente les colonnes issues de l'instruction INSERT ainsi que les colonnes définies par tout déclencheur antérieur à l'insertion. Par exemple, Tally pourra être omis dans l'instruction INSERT mais il sera défini par le déclencheur de dédoublonnage antérieur à l'insertion ; ainsi, new.Tally aura la même nouvelle valeur de colonne dans un déclencheur postérieur à l'insertion.

Exemple

Le déclencheur de base de données suivant utilise la variable `NEW` pour mettre à jour la colonne StateChange en cas de modification d'une ligne de la table alerts.status pour horodater la modification.

```
create trigger SetStateChange
group default_triggers
priority 1
```

```
before update on alerts.status
for each row
begin
  set new.StateChange = getdate;
end;
```

Création de déclencheurs pour la notification des événéments accélérés

Pour prendre en charge une notification d'événement accéléré, créez des déclencheurs postérieurs à l'insertion, à la mise à jour et à la réinsertion qui sont liés à la table alerts.status. Dans les déclencheurs, configurez des conditions permettant de définir ou d'identifier des événements accélérés lorsqu'ils sont insérés ou mis à jour dans la table alerts.status et de transférer ces événements aux clients de notification d'événement accéléré appropriés.

Deux commandes SQL peuvent être utilisées avec vos déclencheurs : une commande de suivi rapide des événements (ou événement accéléré) (IDUC EVTFT) et une commande d'envoi de messages (IDUC SNDMSG).

Conseil : Vous pouvez trouver utile de regrouper des déclencheurs prenant en charge la notification d'événement accéléré au sein de leur propre groupe.

Concepts associés:

Chapitre 6, «Configuration de la notification des événements accélérés», à la page 235
Vous pouvez configurer Tivoli Netcool/OMNIbus pour accélérer la notification des événements pouvant présenter un risque pour le système. Le système Accelerated Event Notification (AEN) permet d'accélérer les événements de priorité élevée afin de garantir une exécution des systèmes sans interruption.

Référence associée:

«IDUC EVTFT», à la page 395
La commande `IDUC EVTFT` permet d'activer les notificateurs contextuels des événements accélérés à envoyer aux clients, puis d'activer la fonction de la liste d'événements de bureau ou de la liste d'événements actifs de l'interface graphique Web.

«IDUC SNDMSG», à la page 397
La commande `IDUC SNDMSG` permet d'envoyer des messages d'informations à un client Notification d'événement accéléré.

Signaux système et leurs attributs

Si un signal système est émis, les attributs qui identifient la cause sont définis. Ces attributs sont transmis sous forme de variables implicites dans le déclencheur de signal associé.

Vous pouvez faire référence aux variables du signal système en utilisant la notation `%signal` dans la section d'action du déclencheur de signal. Le symbole `%` indique que vous faites référence à une variable implicite. Le mot clé `signal` fait référence au signal en cours de transmission au déclencheur. Par exemple, pour faire référence à l'heure à laquelle un signal système a été émis dans un déclencheur de signal, utilisez la syntaxe suivante :

```
%signal.at
```

Conseil : Vous pouvez également utiliser le bouton d'aide `%` pour sélectionner des variables `%signal`.

Les signaux système peuvent être émis par le serveur ObjectServer ou la passerelle, comme suit :

- «Signal de démarrage»
- «Signal d'arrêt», à la page 183
- «Signal de connexion», à la page 183
- «Signal de déconnexion», à la page 183
- «Signal d'échec de sauvegarde», à la page 184
- «Signal de réussite de sauvegarde», à la page 184
- «Signal d'échec de connexion», à la page 184
- «Signal de délai de sécurité», à la page 185
- «Signal create_object», à la page 185
- «Signal alter_object», à la page 186
- «Signal drop_object», à la page 187
- «Signal permission_denied», à la page 188
- «Signal gw_counterpart_down», à la page 188
- «Signal gw_counterpart_up», à la page 188
- «Signal iduc_missed», à la page 189
- «Signal iduc_connect», à la page 189
- «Signal iduc_disconnect», à la page 190
- «Signal iduc_data_fetch», à la page 190
- «Signal resync_lock», à la page 190
- «Signal resync_unlock», à la page 191
- «Signal gw_resync_start», à la page 191
- «Signal gw_resync_finish», à la page 192

Conseil : Vous pouvez analyser les tables catalog.primitive_signals et catalog.primitive_signal_parameters pour afficher des informations relatives aux signaux système. Par exemple, pour afficher les attributs de chaque signal système, utilisez la commande SQL suivante :

```
SELECT * FROM catalog.primitive_signal_parameters ORDER BY SignalName,
OrdinalPosition;
```

Signal de démarrage

Le signal de démarrage est émis au démarrage du serveur ObjectServer. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 33. Attributs du signal de démarrage

Attributs	Type de données	Description
serveur	chaîne	Indique le nom du profil du serveur ObjectServer qui a démarré.
noeud	chaîne	Indique l'ordinateur sur lequel le serveur ObjectServer est démarré.
à	UTC	Indique l'heure à laquelle le serveur ObjectServer a démarré.

Signal d'arrêt

Le signal d'arrêt est émis à l'arrêt du serveur ObjectServer. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 34. Attributs du signal d'arrêt

Attributs	Type de données	Description
serveur	chaîne	Indique le nom du profil du serveur ObjectServer qui est arrêté.
noeud	chaîne	Indique l'ordinateur sur lequel le serveur ObjectServer est arrêté.
à	UTC	Indique l'heure à laquelle le serveur ObjectServer est arrêté.

Signal de connexion

Le signal de connexion est émis lorsqu'un client se connecte au serveur ObjectServer. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 35. Attributs du signal de connexion

Attributs	Type de données	Description
processus	chaîne	Indique le type de processus client qui s'est connecté au serveur ObjectServer.
description	chaîne	Contient des informations complémentaires relatives au client qui s'est connecté, le cas échéant. Par exemple, si le client est une sonde, la description contient le nom de la sonde.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui s'est connecté au serveur ObjectServer.
noeud	chaîne	Indique le nom de l'ordinateur client qui s'est connecté au serveur ObjectServer.
ID_connexion	int	Identifie la connexion de manière unique.
à	UTC	Indique l'heure à laquelle le client s'est connecté.

Signal de déconnexion

Le signal de déconnexion est émis lorsqu'un client se déconnecte du serveur ObjectServer. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 36. Attributs du signal de déconnexion

Attributs	Type de données	Description
processus	chaîne	Indique le type de processus qui s'est déconnecté du serveur ObjectServer.
description	chaîne	Contient des informations complémentaires relatives au client qui s'est déconnecté, le cas échéant. Par exemple, si le client est une sonde, la description contient le nom de la sonde.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui s'est déconnecté du serveur ObjectServer.
noeud	chaîne	Indique le nom de l'ordinateur client qui s'est déconnecté du serveur ObjectServer.

Tableau 36. Attributs du signal de déconnexion (suite)

Attributs	Type de données	Description
ID_connexion	int	Identifie la connexion de manière unique.
à	UTC	Indique l'heure à laquelle le client s'est déconnecté.

Signal d'échec de sauvegarde

Le signal d'échec de sauvegarde est émis lorsqu'une tentative de sauvegarde du serveur ObjectServer n'a pas abouti. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 37. Attributs du signal d'échec de sauvegarde

Attributs	Type de données	Description
erreur	chaîne	Indique la raison de l'échec de la sauvegarde.
à	UTC	Indique l'heure à laquelle la tentative de sauvegarde a eu lieu.
préfixe_chemin	chaîne	Indique le répertoire dans lequel la sauvegarde a tenté d'écrire.
temps_écoulé	réel	Indique le nombre de sauvegardes exécutées avant l'échec.
noeud	chaîne	Indique le nom de l'ordinateur à partir duquel la sauvegarde a été exécutée.

Signal de réussite de sauvegarde

Le signal de réussite de sauvegarde est émis lorsque la sauvegarde du serveur ObjectServer a abouti. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 38. Attributs du signal de réussite de sauvegarde

Attributs	Type de données	Description
à	UTC	Indique l'heure à laquelle la sauvegarde a eu lieu.
préfixe_chemin	chaîne	Indique le répertoire dans lequel la sauvegarde a été écrite.
temps_écoulé	réel	Indique la durée de la sauvegarde.
noeud	chaîne	Indique le nom de l'ordinateur à partir duquel la sauvegarde a été exécutée.

Signal d'échec de connexion

Le signal d'échec de connexion est émis lorsqu'un client n'a pas pu se connecter au serveur ObjectServer. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 39. Attributs du signal d'échec de connexion

Attributs	Type de données	Description
processus	chaîne	Indique le nom du processus qui n'a pas pu se connecter en raison du refus de connexion.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui n'a pas pu se connecter en raison du refus de connexion.

Tableau 39. Attributs du signal d'échec de connexion (suite)

Attributs	Type de données	Description
noeud	chaîne	Indique le nom de l'ordinateur client qui n'a pas pu se connecter en raison du refus de connexion.
à	UTC	Indique l'heure à laquelle le client n'a pas pu se connecter en raison d'un refus de connexion.

Signal de délai de sécurité

Le signal de délai de sécurité est émis lorsque le délai de connexion au serveur ObjectServer a expiré. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 40. Attributs du signal de délai de connexion

Attributs	Type de données	Description
processus	chaîne	Indique le nom du processus qui n'a pas pu se connecter car les données d'identification de connexion n'ont pas pu être validées.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui n'a pas pu se connecter car les données d'identification de connexion n'ont pas pu être validées.
noeud	chaîne	Indique le nom du client qui n'a pas pu se connecter car les données d'identification de connexion n'ont pas pu être validées.
à	UTC	Indique l'heure à laquelle le client n'a pas pu se connecter car les données d'identification de connexion n'ont pas pu être validées.

Signal create_object

Le signal create_object est émis lorsqu'un objet est créé sur le serveur ObjectServer. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 41. Attributs du signal de création d'objets

Attributs	Type de données	Description
type_objet	chaîne	Indique l'un des types d'objet suivants : • CREATE DATABASE • CREATE TABLE • CREATE INDEX • CREATE TRIGGER GROUP • CREATE TRIGGER • CREATE PROCEDURE • CREATE RESTRICTION FILTER • CREATE USER SIGNAL • CREATE FILE • CREATE USER • CREATE GROUP • CREATE ROLE

Tableau 41. Attributs du signal de création d'objets (suite)

Attributs	Type de données	Description
nom_parent	chaîne	Indique le nom de l'objet parent. Pour les déclencheurs, il s'agit du nom du groupe de déclencheurs. Pour les tables, il s'agit du nom de la base de données. Les autres objets ne détiennent pas d'objet parent.
nom	chaîne	Indique le nom de l'objet. Par exemple, la valeur de la table alerts.status est `status`.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui a exécuté la commande.
serveur	chaîne	Indique le nom du serveur ObjectServer auquel a été ajouté l'objet.
noeud	chaîne	Indique le nom de l'ordinateur qui exécute le serveur ObjectServer auquel a été ajouté l'objet.
nom_hôte	chaîne	Indique le nom de l'ordinateur client à l'origine de la demande d'ajout de l'objet.
à	UTC	Indique l'heure à laquelle l'objet a été ajouté.

Signal alter_object

Le signal alter_object est émis lorsqu'un objet du serveur ObjectServer est modifié. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 42. Attributs du signal alter_object

Attributs	Type de données	Description
type_objet	chaîne	Indique l'un des types d'objet suivants : • `ALTER TABLE` • `ALTER TRIGGER GROUP` • `ALTER TRIGGER` • `ALTER PROCEDURE` • `ALTER RESTRICTION FILTER` • `ALTER USER SIGNAL` • `ALTER FILE` • `ALTER USER` • `ALTER GROUP` • `ALTER ROLE` **Remarque :** Les droits ALTER PROCEDURE, ALTER RESTRICTION FILTER et ALTER USER SIGNAL sont obligatoires si une commande CREATE OR REPLACE est exécutée en fonction d'un objet de l'un de ces types et si l'objet existe déjà. Vous devez détenir les droits ALTER appropriés même en l'absence de la commande ALTER pour ces objets.
nom_parent	chaîne	Indique le nom de l'objet parent. Pour les déclencheurs, il s'agit du nom du groupe de déclencheurs. Pour les tables, il s'agit du nom de la base de données. Les autres objets ne détiennent pas d'objet parent.
nom	chaîne	Indique le nom de l'objet. Par exemple, la valeur de la table alerts.status est `status`.

Tableau 42. Attributs du signal alter_object (suite)

Attributs	Type de données	Description
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui a exécuté la commande.
serveur	chaîne	Indique le nom du serveur ObjectServer sur lequel a été modifié l'objet.
noeud	chaîne	Indique le nom de l'ordinateur qui exécute le serveur ObjectServer sur lequel a été modifié l'objet.
nom_hôte	chaîne	Indique le nom de l'ordinateur client à l'origine de la demande de modification de l'objet.
à	UTC	Indique l'heure à laquelle l'objet a été modification.

Signal drop_object

Le signal drop_object est émis lorsqu'un objet du serveur ObjectServer est supprimé. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 43. Attributs du drop_object

Attributs	Type de données	Description
type_objet	chaîne	Indique l'un des types d'objet suivants : • `DROP DATABASE` • `DROP TABLE` • `DROP INDEX` • `DROP TRIGGER GROUP` • `DROP TRIGGER` • `DROP PROCEDURE` • `DROP RESTRICTION FILTER` • `DROP USER SIGNAL` • `DROP FILE` • `DROP USER` • `DROP GROUP` • `DROP ROLE`
nom_parent	chaîne	Indique le nom de l'objet parent. Pour les déclencheurs, il s'agit du nom du groupe de déclencheurs. Pour les tables, il s'agit du nom de la base de données. Les autres objets ne détiennent pas d'objet parent.
nom	chaîne	Indique le nom de l'objet. Par exemple, la valeur de la table alerts.status est `status`.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui a exécuté la commande.
serveur	chaîne	Indique le nom du serveur ObjectServer à partir duquel a été modifié l'objet.
noeud	chaîne	Indique le nom de l'ordinateur qui exécute le serveur ObjectServer à partir duquel a été modifié l'objet.
nom_hôte	chaîne	Indique le nom de l'ordinateur client à l'origine de la demande de suppression de l'objet.
à	UTC	Indique l'heure à laquelle l'objet a été supprimé.

Signal permission_denied

Le signal permission_denied est émis lorsque les droits de réalisation d'une opération sont refusés. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 44. Attributs du signal permission_denied

Attributs	Type de données	Description
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur dont la demande a généré une erreur de droits refusés.
serveur	chaîne	Indique le nom du serveur ObjectServer qui a généré l'erreur de droits refusés.
noeud	chaîne	Indique le nom de l'ordinateur qui exécute le serveur ObjectServer qui a généré l'erreur de droits refusés.
nom_hôte	chaîne	Indique le nom de l'ordinateur client à l'origine de la demande ayant engendré l'erreur de droits refusés.
à	UTC	Indique l'heure à laquelle l'erreur de droits refusés a eu lieu.
cmd_sql	chaîne	Indique la commande SQL qui a généré l'erreur de droits refusés.

Signal gw_counterpart_down

La passerelle émet un signal gw_counterpart_down dans le serveur ObjectServer de sauvegarde lorsqu'elle détecte que le serveur ObjectServer principal est indisponible. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 45. Attributs du signal gw_counterpart_down

Attributs	Type de données	Description
serveur	chaîne	Indique le nom du serveur ObjectServer homologue défaillant, dans une paire reprise en ligne/restauration.
noeud	chaîne	Indique le nom de l'ordinateur à partir duquel le serveur ObjectServer homologue a été exécuté.
à	UTC	Indique l'heure à laquelle le serveur ObjectServer homologue est tombé en panne.
nom_passerelle	chaîne	Indique le nom de la passerelle entre les serveurs ObjectServer principal et de sauvegarde.

Signal gw_counterpart_up

La passerelle émet un signal gw_counterpart_up dans le serveur ObjectServer de sauvegarde lorsqu'elle détecte que le serveur ObjectServer principal est de nouveau disponible. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 46. Attributs du signal gw_counterpart_up

Attributs	Type de données	Description
serveur	chaîne	Indique le nom du serveur ObjectServer homologue de nouveau disponible, dans une paire reprise en ligne/restauration.

Tableau 46. Attributs du signal gw_counterpart_up (suite)

Attributs	Type de données	Description
noeud	chaîne	Indique le nom de l'ordinateur à partir duquel le serveur ObjectServer homologue a été exécuté.
à	UTC	Indique l'heure à laquelle le serveur ObjectServer homologue est tombé en panne.
nom_passerelle	chaîne	Indique le nom de la passerelle entre les serveurs ObjectServer principal et de sauvegarde.

Signal iduc_missed

Le signal iduc_missed est émis à chaque fois qu'un client de bureau ou de passerelle ne répond pas à un invite IDUC provenant du serveur ObjectServer. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 47. Attributs du signal iduc_missed

Attributs	Type de données	Description
processus	chaîne	Indique le type de processus client qui n'a pas répondu à l'invite IDUC provenant du serveur ObjectServer.
description	chaîne	Contient des informations complémentaires relatives au client, le cas échéant.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur connecté au serveur ObjectServer.
noeud	chaîne	Indique le nom de l'ordinateur client connecté au serveur ObjectServer.
ID_connexion	int	Identifie la connexion de manière unique.
à	UTC	Indique l'heure à laquelle le cycle IDUC a été manqué.
cycles_manqués	int	Indique le nombre de cycles IDUC manqués de manière consécutives.

Signal iduc_connect

Le signal iduc_connect est émis lorsqu'un client établit une connexion IDUC. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 48. Attributs du signal iduc_connect

Attributs	Type de données	Description
processus	chaîne	Indique le type de processus client qui s'est connecté au serveur ObjectServer.
description	chaîne	Contient des informations complémentaires relatives au client qui s'est connecté, le cas échéant. Par exemple, si le client est une sonde, la description contient le nom de la sonde.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui s'est connecté au serveur ObjectServer.
noeud	chaîne	Indique le nom de l'ordinateur client qui s'est connecté au serveur ObjectServer.
ID_conn	int	Indique l'ID de la connexion.

Signal iduc_disconnect

Le signal iduc_disconnect est émis lorsqu'un client déconnecte une connexion IDUC établie. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 49. Attribut du signal iduc_disconnect

Attributs	Type de données	Description
processus	chaîne	Indique le type de processus client qui s'est déconnecté du serveur ObjectServer.
description	chaîne	Contient des informations complémentaires relatives au client qui s'est déconnecté, le cas échéant. Par exemple, si le client est une sonde, la description contient le nom de la sonde.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui s'est déconnecté du serveur ObjectServer.
noeud	chaîne	Indique le nom de l'ordinateur client qui s'est déconnecté du serveur ObjectServer.
ID_conn	int	Indique l'ID de la connexion.

Signal iduc_data_fetch

Le signal iduc_data_fetch se déclenche lorsqu'un client IDUC extrait ses modifications IDUC du serveur ObjectServer. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 50. Attributs du signal iduc_data_fetch

Attributs	Type de données	Description
processus	chaîne	Indique le type de processus client qui a nécessité les modifications IDUC en attente à partir du serveur ObjectServer.
description	chaîne	Contient des informations complémentaires relatives au client, le cas échéant. Par exemple, si le client est une sonde, la description contient le nom de la sonde.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur connecté au serveur ObjectServer.
noeud	chaîne	Indique le nom de l'ordinateur client connecté au serveur ObjectServer.
ID_connexion	int	Identifie la connexion de manière unique.
à	UTC	Indique l'heure à laquelle le client a extrait les modifications correspondant à la dernière notification IDUC.

Signal resync_lock

Le signal resync_lock est émis par le serveur ObjectServer lorsque la resynchronisation est verrouillée. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 51. Attributs du signal resync_lock

Attributs	Type de données	Description
processus	chaîne	Indique le type de processus client qui a verrouillé le verrou de resynchronisation.
description	chaîne	Contient des informations complémentaires relatives au client qui a verrouillé la resynchronisation, le cas échéant. Par exemple, si le client est une sonde, la description contient le nom de la sonde.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui exécute le processus.
noeud	chaîne	Indique le nom de l'ordinateur client connecté au serveur ObjectServer.
à	UTC	Indique l'heure à laquelle le signal a eu lieu

Signal resync_unlock

Le signal resync_unlock est émis par le serveur ObjectServer lorsque la resynchronisation est déverrouillée. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 52. Attributs du signal resync_unlock

Attributs	Type de données	Description
processus	chaîne	Indique le type de processus client qui a déverrouillé la resynchronisation.
description	chaîne	Contient des informations complémentaires relatives au client qui a déverrouillé la resynchronisation, le cas échéant. Par exemple, si le client est une sonde, la description contient le nom de la sonde.
nom d'utilisateur	chaîne	Indique le nom de l'utilisateur qui exécute le processus.
noeud	chaîne	Indique le nom de l'ordinateur client connecté au serveur ObjectServer.
à	UTC	Indique l'heure à laquelle le signal a eu lieu

Signal gw_resync_start

Le signal gw_resync_start est émis par la passerelle afin d'indiquer le début d'une opération de resynchronisation. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 53. Attributs du signal gw_resync_start

Attributs	Type de données	Description
nom_passerelle	chaîne	Indique le nom de la passerelle qui a lancé la resynchronisation.
noeud	chaîne	Indique le nom d'hôte de l'ordinateur sur lequel s'exécute la passerelle.
à	UTC	Indique l'heure à laquelle la resynchronisation a démarré.
est_maître	Booléen	Indique si le serveur ObjectServer local est le maître ou l'esclave de la resynchronisation.

Signal gw_resync_finish

Le signal gw_resync_finish est émis par la passerelle afin d'indiquer la fin d'une opération de resynchronisation. Le tableau ci-dessous décrit les attributs de ce signal.

Tableau 54. Attributs du signal gw_resync_finish

Attributs	Type de données	Description
nom_passerelle	chaîne	Indique le nom de la passerelle qui termine la resynchronisation.
noeud	chaîne	Indique le nom d'hôte de l'ordinateur sur lequel s'exécute la passerelle.
à	UTC	Indique l'heure à laquelle la resynchronisation s'est terminée.
est_maître	Booléen	Indique si le serveur ObjectServer local est le maître ou l'esclave de la resynchronisation.

Référence associée:

Annexe C, «Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires», à la page 411
Vous pouvez utiliser un certain nombre de commandes SQL, d'expressions variables et de boutons d'aide pour récupérer des informations d'une liste d'événements en cours, de l'événement actuel ou de l'environnement du système d'exploitation. Vous pouvez utiliser ces expressions lors de la création d'un outil, d'un déclencheur ou d'une procédure SQL, ainsi que dans les paramètres transmis à une liste d'événements transitoires.

Procédures

Une procédure SQL est un ensemble de commandes SQL paramétrées ou de fragments de code doté de constructions de langage de programmation que vous pouvez utiliser pour effectuer des tâches complexes sur des objets de base de données.

Après avoir créé une procédure dans le serveur ObjectServer, vous pouvez l'exécuter à partir de l'interface interactive SQL. Vous pouvez également exécuter la procédure dans un déclencheur à l'aide de la commande `EXECUTE PROCEDURE`.

Vous pouvez créer des procédures externes pour exécuter un programme exécutable sur un système local ou distant.

Vous pouvez créer une procédure contenant un ensemble logique de commandes qui constitue une tâche (par exemple, ensemble de requêtes, de mises à jour ou d'insertions).

Les procédures développent la syntaxe SQL afin d'effectuer les opérations suivantes :

- Transmission des paramètres dans une procédure ou en dehors d'une procédure
- Création de variables locales et affectation de valeurs à des dernières
- Exécution d'un test de condition
- Exécution d'opérations d'analyse sur les tables et les vues

Composants d'une procédure SQL

Les procédures SQL possèdent les composants principaux suivants : paramètres, déclarations de variable locale et corps de procédure.

Les *paramètres* sont des valeurs transmises à/émises d'une procédure. Déclarez les paramètres de la procédure lorsque vous créez la procédure et spécifiez les valeurs transmises comme paramètres lors de l'exécution de la procédure. Le nom de la variable contenant un paramètre est nommé *paramètre formel*, tandis que la valeur du paramètre lors de l'exécution de la procédure est nommée *paramètre réel*.

Les valeurs transmises à la procédure doivent être du même type de données que dans la déclaration de paramètre.

Vous pouvez également créer des *variables locales* à utiliser dans la procédure et devant contenir et modifier les valeurs temporaires du corps de la procédure. Les variables locales et les variables sont toujours supprimées à la sortie de la procédure. Par exemple, vous pouvez créer un compteur d'entier sous forme de variable locale.

Remarque : Les paramètres et les variables locales contenant des données modifiables, ces deux éléments sont nommés 'variables' dans les procédures.

Le *corps* d'une procédure contient un ensemble d'instructions qui teste les conditions et manipule les données dans la base de données.

Référence associée:

«CREATE PROCEDURE (procédures SQL)», à la page 367
La commande `CREATE PROCEDURE` permet de créer des procédures SQL.

Construction d'une instruction de corps de procédure SQL

Le corps d'une procédure SQL contient un ensemble d'instructions SQL et de constructions de programmation qui gèrent les données du serveur ObjectServer.

Syntaxe

```
CREATE [ OR REPLACE ] PROCEDURE nom_procédure ([ paramètre_procédure,... ])
 [ DECLARE déclaration_variable;...[;] ]
 BEGIN
   instruction_corps_procédure;...[;]
END
```

Cette rubrique décrit uniquement les entrées requises pour le corps d'une procédure (*instruction_corps_procédure*), qui est placé entre les mots clés BEGIN et END.

Dans le corps d'une procédure, vous devez séparer chaque instruction par un pont-virgule, sauf la dernière.

Les instructions de la procédure peuvent inclure des commandes SQL et des constructions de programmation supplémentaires.

Vous pouvez exécuter les commandes SQL suivantes dans une procédure :

```
ALTER FILE
ALTER SYSTEM BACKUP
ALTER SYSTEM SET
ALTER SYSTEM DROP CONNECTION
ALTER TRIGGER
ALTER TRIGGER GROUP
ALTER USER
```

```
UPDATE
INSERT
DELETE
WRITE INTO
RAISE SIGNAL
{ EXECUTE | CALL } PROCEDURE
```

L'utilisateur qui crée la procédure doit détenir les droits d'accès appropriés pour exécuter les commandes dans le corps de la procédure.

Avertissement : Les procédures ou les déclencheurs ne peuvent pas comporter de dépendances circulaires. Par exemple, vous ne devez pas créer de procédure qui appelle une autre procédure qui, elle-même, appelle la procédure d'origine.

Vous pouvez utiliser les constructions de programmation suivantes dans le corps de la procédure :

- Instruction d'affectation SET
- Instruction IF THEN ELSE
- Instruction CASE WHEN
- Boucle FOR EACH ROW
- Boucle FOR

Référence associée:

«Variables utilisateur implicites dans les procédures et les déclencheurs», à la page 195
Vous pouvez utiliser des variables utilisateur pour accéder aux informations relatives aux utilisateurs connectés dans une expression SQL dans le corps d'un déclencheur ou d'une procédure.

«SET», à la page 406
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser une instruction d'affectation `SET` pour écrire la valeur d'une expression dans une variable ou un paramètre.

«IF THEN ELSE», à la page 395
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser l'instruction `IF THEN ELSE` pour réaliser une ou plusieurs actions en fonction des conditions spécifiées.

«CASE WHEN», à la page 361
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser l'instruction `CASE WHEN` pour effectuer une ou plusieurs actions basées sur une condition. Si la condition n'est pas remplie, vous pouvez, si vous le souhaitez, effectuer une action différente.

«FOR EACH ROW», à la page 390
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser la boucle `FOR EACH ROW` pour réaliser des actions sur un ensemble de lignes répondant à une certaine condition.

«FOR», à la page 390
Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser la boucle `FOR` pour réaliser des actions un certain nombre de fois, en fonction de la variable de comptage.

«CREATE PROCEDURE (procédures SQL)», à la page 367
La commande `CREATE PROCEDURE` permet de créer des procédures SQL.

Variables utilisateur implicites dans les procédures et les déclencheurs

Vous pouvez utiliser des variables utilisateur pour accéder aux informations relatives aux utilisateurs connectés dans une expression SQL dans le corps d'un déclencheur ou d'une procédure.

Utilisez la notation `%user` pour spécifier les variables utilisateur (`%user.`*`nom_attribut`*, par exemple). Le symbole `%` indique que vous référencez une variable implicite. Le mot clé `user` fait référence à l'utilisateur en cours.

Conseil : Vous pouvez également utiliser le bouton d'aide `%` pour sélectionner des variables `%user`.

Le tableau ci-dessous répertorie les attributs en lecture seule disponibles dans les procédures et déclencheurs.

Tableau 55. Variables utilisateur implicites

Attribut de variable	Type de données	Description
`%user.ID_utilisateur`	INTEGER	Identifiant de l'utilisateur connecté.
`%user.nom_utilisateur`	STRING	Nom de l'utilisateur connecté.
`%user.nom_app`	STRING	Nom de l'application connectée (`nco_sql`, par exemple).
`%user.nom_hôte`	STRING	Nom de l'hôte connecté.
`%user.ID_connexion`	UNSIGNED	ID connexion. Voir «Exemple : utilisation de %user.ID_homologue et %user.ID_connexion», à la page 196.
`%user.ID_homologue`	UNSIGNED	ID connexion de l'homologue. Voir «Exemple : utilisation de %user.ID_homologue et %user.ID_connexion», à la page 196.
`%user.est_auto`	BOOLEAN	Si la valeur attribuée est `TRUE`, l'action en cours a été générée suite à l'exécution d'une automatisation (un déclencheur temporel, par exemple).
`%user.est_passerelle`	BOOLEAN	Si la valeur attribuée est `TRUE`, l'action a été générée par un client de passerelle.
`%user.est_liste_événements`	BOOLEAN	Si la valeur attribuée est `TRUE`, l'action en cours a été générée par un client de liste d'événements.
`%user.description`	STRING	Nom descriptif de l'application. S'applique uniquement aux passerelles ou sondes ObjectServer.

Exemple : utilisation de %user.nom_utilisateur

Pour faire référence au nom de l'utilisateur en cours dans le corps d'une procédure ou d'un déclencheur, utilisez la syntaxe suivante :

```
%user.nom_utilisateur
```

Exemple : utilisation de %user.ID_homologue et %user.ID_connexion

Pour les passerelles, si `%user.ID_connexion` fait référence à un composant de programme d'écriture de passerelle, `%user.ID_homologue` est l'ID de saut du composant de programme de lecture de la passerelle. Cet exemple illustre l'utilisation de ces variables dans une reprise en ligne et une reprise par restauration automatiques, afin de déconnecter tous les clients (sauf la passerelle) d'un serveur ObjectServer de secours au démarrage du serveur ObjectServer principal. Le code suivant peut être ajouté dans le déclencheur `homologue_secours_marche` :

```
For each row connected in catalog.connections where
 (connected.ConnectionID <> %user.connection_id and
 connected.ConnectionID <> %user.counterpart_id and
 connected.AppName == 'GATEWAY') or
 connected.AppName <> 'GATEWAY'
begin
alter system drop connection connected.ConnectionID;
end;
```

Référence associée:

Annexe C, «Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires», à la page 411
Vous pouvez utiliser un certain nombre de commandes SQL, d'expressions variables et de boutons d'aide pour récupérer des informations d'une liste d'événements en cours, de l'événement actuel ou de l'environnement du système d'exploitation. Vous pouvez utiliser ces expressions lors de la création d'un outil, d'un déclencheur ou d'une procédure SQL, ainsi que dans les paramètres transmis à une liste d'événements transitoires.

Automatisations Tivoli Netcool/OMNIbus standard

Un ensemble d'automatisations standard est inclus à Tivoli Netcool/OMNIbus. Ces automatisations sont créées lors de l'initialisation de la base de données.

Les automatisations standard sont stockées dans l'emplacement `$NCHOME/omnibus/etc/automation.sql`. Vous pouvez ouvrir le fichier `automation.sql` dans un éditeur de texte et afficher la syntaxe de chaque automatisation. Les commentaires sont inclus afin de décrire l'objet des automatisations. Certaines automatisations du fichier `automation.sql` ne sont pas activées par défaut.

Vous pouvez également utiliser Netcool/OMNIbus Administrator pour parcourir ces automatisations en sélectionnant le bouton de menu **Automatisation (Automatisation)** de la fenêtre Netcool/OMNIbus Administrator. Vous vous apercevrez probablement que certains groupes de déclencheurs sont désactivés par défaut, alors que les déclencheurs appartenant à ce groupe sont à l'état activé. Vous devez activer ces groupes de déclencheurs afin d'exécuter les déclencheurs. Par exemple, le groupe de déclencheurs audit_config offre la possibilité d'émettre des alertes à chaque modification des objets du serveur ObjectServer. Ce groupe de déclencheurs peut faire office de mécanisme d'audit conjointement avec les fichiers suivi responsable du répertoire `$NCHOME/omnibus/log`.

Les fonctions réalisées par certaines automatisations standard incluent :

- Sauvegarde du serveur ObjectServer
- Ajout d'alertes au serveur ObjectServer
- Insertion d'entrées de journal

- Suppression d'entrées redondantes de différentes tables

Les automatisations standard sont répertoriées dans le tableau ci-dessous.

Tableau 56. Automatisations standard

Nom du déclencheur ou de la procédure	**Description**
audit_config_alter_class	Permet de créer une alerte indiquant qu'une classe a été modifiée.
audit_config_alter_col_visual	Permet de créer une alerte indiquant que le visuel d'une colonne a été modifié.
audit_config_alter_conv	Permet de créer une alerte indiquant qu'une conversion a été modifiée.
audit_config_alter_menu	Permet de créer une alerte indiquant qu'un menu a été modifié.
audit_config_alter_object	Permet de créer une alerte indiquant qu'un objet a été modifié.
audit_config_alter_prompt	Permet de créer une alerte indiquant qu'une invite a été modifiée.
audit_config_alter_property	Permet de créer une alerte indiquant qu'une propriété a été modifiée.
audit_config_alter_tool	Permet de créer une alerte indiquant qu'un outil a été modifié.
audit_config_create_class	Permet de créer une alerte indiquant qu'une classe a été créée.
audit_config_create_col_visual	Permet de créer une alerte indiquant que le visuel d'une colonne a été créé.
audit_config_create_conv	Permet de créer une alerte indiquant qu'une conversion a été créée.
audit_config_create_menu	Permet de créer une alerte indiquant qu'un menu a été créé.
audit_config_create_object	Permet de créer une alerte indiquant qu'un objet a été créé.
audit_config_create_prompt	Permet de créer une alerte indiquant qu'une invite a été créée.
audit_config_create_tool	Permet de créer une alerte indiquant qu'un outil a été créé.
audit_config_drop_class	Permet de créer une alerte indiquant qu'une classe a été supprimée.
audit_config_drop_col_visual	Permet de créer une alerte indiquant que le visuel d'une colonne a été supprimé.
audit_config_drop_conv	Permet de créer une alerte indiquant qu'une conversion a été supprimée.
audit_config_drop_menu	Permet de créer une alerte indiquant qu'un menu a été supprimé.
audit_config_drop_object	Permet de créer une alerte indiquant qu'un objet a été supprimé.
audit_config_drop_prompt	Permet de créer une alerte indiquant qu'une invite a été supprimée.

Tableau 56. Automatisations standard (suite)

Nom du déclencheur ou de la procédure	**Description**
audit_config_drop_tool	Permet de créer une alerte indiquant qu'un outil a été supprimé.
audit_config_permission_denied	Permet de créer une alerte indiquant qu'un droit a été refusé.
automatic_backup	Permet de sauvegarder tous les stockages en mémoire sur le serveur ObjectServer dans une séquence d'emplacements qui dépend de la valeur définie pour une variable *nombre_sauvegardes*.
automation_disable	Permet de désactiver les automatisations qui ne doivent pas fonctionner lorsqu'il s'agit d'un serveur ObjectServer de sauvegarde.
automation_enable	Permet d'activer les automatisations qui doivent fonctionner lorsqu'il s'agit d'un serveur ObjectServer principal.
backup_counterpart_down	Permet d'activer les automatisations qui doivent fonctionner lorsque le serveur ObjectServer principal tombe en panne et que le serveur ObjectServer de sauvegarde le remplace.
backup_counterpart_up	Permet de désactiver les automatisations qui ne doivent pas fonctionner sur le serveur ObjectServer de sauvegarde lorsque le serveur ObjectServer principal redémarre.
backup_failed	Permet de spécifier une action à réaliser sur une opération de sauvegarde qui n'a pas abouti.
backup_startup	Permet de désactiver les automatisations qui ne doivent pas fonctionner lorsqu'un serveur ObjectServer désigné comme sauvegarde est démarré.
backup_state_integrity	Permet de garantir la présence d'un seul enregistrement dans la table de séquence de saisie de sauvegarde en annulant toutes les autres insertions.
backup_succeeded	Permet de spécifier une action à réaliser sur une opération de sauvegarde réussie.
clean_details_table	Permet de nettoyer la table alerts.details. Permet de supprimer toutes les entrées introuvables dans la table alerts.status.
clean_journal_table	Permet de nettoyer la table alerts.journal. Permet de supprimer toutes les entrées introuvables dans la table alerts.status.
connection_watch_connect	Permet de créer une alerte lorsqu'un nouveau client se connecte. Le nom du processus ou de l'application identifié par le signal est mis en correspondance avec la table alerts.application_types afin d'identifier la gravité appropriée et le type d'événement pour la connexion. Une connexion de passerelle, par exemple, est traitée comme une résolution (suppression d'une déconnexion), alors qu'une connexion de liste d'événements est un événement de Type 1 résolu par une déconnexion.

Tableau 56. Automatisations standard (suite)

Nom du déclencheur ou de la procédure	Description
connection_watch_disconnect	Permet de créer une alerte lorsqu'un nouveau client se déconnecte. Le nom du processus ou de l'application identifié par le signal est mis en correspondance avec la table alerts.application_types afin d'identifier la gravité appropriée et le type d'événement pour la déconnexion. Une déconnexion de passerelle, par exemple, est traitée comme un problème, alors qu'une déconnexion de liste d'événements est une résolution.
dedup_status_inserts	Permet de compter les insertions de table d'état dédoublonnées.
deduplicate_details	Permet de dédoublonner les lignes de la table alerts.details.
deduplicate_iduc_stats	Permet de dédoublonner les lignes de la table iduc_system.iduc_stats.
deduplication	Processus de dédoublonnage de la table alerts.status. Permet de conserver le compte de dédoublonnage et de régénérer les détails de l'alerte.
delete_clears	Toutes les 60 secondes, permet de supprimer les alertes de suppression de plus de deux minutes dans la table alerts.status.
details_inserts	Permet de compter les insertions de la table des détails.
disable_inactive_users	S'exécute une fois par jour pour désactiver les utilisateurs qui ne se sont pas connectés au serveur ObjectServer dans une période définie.
disable_user	Permet de désactiver les utilisateurs après *n* tentatives infructueuses de connexion.
disconnect_iduc_missed	Déconnecte les clients en temps réel qui n'ont pas communiqué avec le serveur ObjectServer pendant 100 périodes de granularité.
escalate_off	Permet d'attribuer la valeur 0 (pas de clignotement) aux zones Flash et SuppressEscl (pas d'escalade dans cet exemple) lorsqu'un événement dont la zone Flash était définie sur 1 est reconnu ou si l'événement est effacé (Gravité = 0).
expire	Permet de gérer l'expiration des alertes. Permet de définir la gravité d'une alerte sur `0` si la valeur d'ExpireTime (temps durant lequel l'alerte est non valide) est dépassée.
flash_not_ack	Permet d'activer le clignotement (Flash = 1) pour les événements de plus de 10 minutes et critiques (Severity = 5), mais qui n'ont pas encore été reconnus par l'utilisateur (Acknowledge = 0). Permet d'attribuer la valeur 1 à SuppressEscl comme indication supplémentaire de l'état d'escalade de l'événement.
generic_clear	Supprime (Gravité = 0) toutes les lignes de la table alerts.status indiquant un dispositif en panne (Type = 1), dans laquelle a été insérée une ligne indiquant que le dispositif a été rétabli (Type = 2).

Tableau 56. Automatisations standard (suite)

Nom du déclencheur ou de la procédure	Description
iduc_messages_tblclean	Permet de nettoyer la table alerts.iduc_messages. S'exécute toutes les 60 secondes et supprime les messages de plus de deux minutes.
iduc_stats_insert	Permet d'insérer une entrée client dans la table iduc_system.iduc_stats lorsque le signal iduc_connect est déclenché.
iduc_stats_update	Permet de mettre à jour la zone LastIducTime de la table iduc_system.iduc_stats lorsque le signal iduc_data_fetch est déclenché.
jinsert	Permet d'insérer un enregistrement dans la table alerts.journal. Les automatisation qui nécessitent des entrées de journal doivent exécuter cette procédure.
journal_inserts	Permet de compter les insertions de table de journal.
mail_on_critical	Permet d'envoyer un message électronique concernant des alertes critiques dont la réception n'a pas été accusée après 30 minutes. **Remarque :** Cet outil est spécifique à UNIX, sauf si un logiciel de courrier NT équivalent est disponible.
new_row	Permet de définir les valeurs par défaut de nouvelles alertes dans la table alerts.status.
new_status_inserts	Permet de compter les nouvelles insertions de la table d'état.
pass_deletes	Permet de supprimer du serveur ObjectServer de destination les lignes qui n'existent pas sur le serveur ObjectServer source après la resynchronisation.
profiler_group_report	Permet d'écrire une ligne dans le fichier `profiler_report` indiquant la somme des temps passés par chaque type d'application distincte lors de la dernière période de profilage.
profiler_report	Permet d'écrire dans le fichier `profiler_report` une ligne par client connecté, avec le temps utilisé par ce dernier lors de la dernière période de profilage.
profiler_toggle	Permet d'enregistrer que le profileur a basculé.
registry_new_probe	Déclencheur de pré-insertion qui lie une entrée de sonde de la table registry.probes à l'entrée correspondante dans la table catalog.connections.
registry_probe_disconnect	Rétablit la valeur 0 de la zone ConnectionID dans la table registry.probes lorsqu'une sonde ou un serveur proxy se déconnecte du serveur ObjectServer.
registry_reinsert_probe	Met à jour les entrées de la table registry.probes en réponse aux instructions de réinsertion qui proviennent d'une sonde. Toutes les colonnes de l'entrée sont mises à jour. La colonne LastUpdate enregistre l'horodatage de la mise à jour. **Remarque :** Ce déclencheur accepte uniquement les réinsertions provenant de sondes. Les autres applications doivent utiliser une instruction UPDATE pour modifier les entrées existantes de la table registry.probes.

Tableau 56. Automatisations standard (suite)

Nom du déclencheur ou de la procédure	Description
registry_update_probe	Ce déclencheur évite que les passerelles ne remplacent une entrée de registre de sonde par des informations obsolètes. Cela peut se produire durant la reprise en ligne ou la reprise par restauration d'une sonde, lorsqu'un cycle IDUC de passerelle se produit une fois que l'entrée de sonde la plus récente a été écrite sur le serveur ObjectServer.
reset_user	Permet de réinitialiser le nombre d'incidents d'un utilisateur qui se connecte successivement.
resync_finished	Permet d'identifier la fin de la resynchronisation et de définir la propriété **ActingPrimary** du serveur ObjectServer de sauvegarde sur FALSE pour en faire le serveur de sauvegarde.
security_watch_security_failure	Permet de créer une alerte lorsqu'un client ne parvient pas à s'authentifier.
service_insert	Traitement de service de la table service.status.
service_reinsert	Traitement de service de la table service.status.
service_update	Traitement de service de la table service.status.
state_change	Traitement de la modification d'état de la table alerts.status. Permet de conserver l'horodatage du serveur Objectserver pour les dernières insertions et mises à jour d'une alerte provenant d'une source.
statistics_cleanup	Permet de supprimer les statistiques datées de plus d'une heure.
statistics_gather	Permet de collecter des métriques, notamment le nombre total de clients connectés au serveur ObjectServer, le nombre de clients en temps réel et le nombre de nouvelles insertions dans la table alerts.status ainsi que les métriques de la table master.stats. Ces données peuvent ensuite être consultées à l'aide de Netcool/OMNIbus Administrator ou de **nco_sql**. Sinon, elles peuvent être écrites dans un fichier ou traitées par d'autres automatisations.
stats_reset	Permet de réinitialiser les données statistiques.
system_watch_shutdown	Permet de créer une alerte indiquant que le serveur ObjectServer est en cours d'arrêt.
system_watch_startup	Permet de créer une alerte indiquant que le serveur ObjectServer a démarré.
trigger_stats_report	Permet d'écrire dans le fichier `trigger_stats.log` la durée utilisée par chaque déclencheur dans la dernière période de profilage.

Tableau 56. Automatisations standard (suite)

Nom du déclencheur ou de la procédure	Description
update_service_affecting_events	S'exécute à une fréquence spécifiée pour activer les événements de service affecté dans Network Manager IP Edition afin de s'effacer automatiquement lorsque leurs événements associés sont supprimés. Un événement affecté par un service est une alerte qui avertit les opérateurs qu'un service client critique a été affecté par un ou plusieurs événements réseau. Cette automatisation fonctionne uniquement avec Tivoli Netcool/OMNIbus version 7.0 ou une version ultérieure. **Conseil :** L'automatisation est requise uniquement si Network Manager IP Edition est utilisé. Elle est utilisée avec les tables precision.entity_service, precision.service_details et precision.service_affecting_event.
webtop_compatibility	Permet de remplir la table master.profiles avec des utilisateurs du serveur ObjectServer afin qu'elle puisse être lue par l'interface graphique Web (ou par Netcool/Webtop). Permet en outre de définir la zone AllowISQL pour chaque utilisateur qui s'est vu accorder des droits d'utilisation de l'outil SQL interactif dans l'interface graphique Web (ou dans Netcool/Webtop).

Automatisation des événements affectés par le service

Un événement affecté par un service (SAE - service-affected event) est une alerte qui avertit les opérateurs qu'un service client critique a été affecté par un ou plusieurs événements réseau. Les événements affectés par un service sont générés dans IBM Tivoli Network Manager IP Edition.

Vous pouvez configurer Tivoli Netcool/OMNIbus pour exécuter une automatisation à une fréquence spécifiée et permettre d'effacer les événements affectés au service dans Network Manager IP Edition lorsque les événements qui leur sont liés sont effacés.

Pour rendre cette fonction opérationnelle au niveau d'une installation de Network Manager IP Edition et de Tivoli Netcool/OMNIbus, vous devez configurer Network Manager IP Edition, tel que cela est décrit dans *IBM Tivoli Network Manager IP Edition Installation and Configuration Guide*, SC27-2760-00. Lorsque vous installez Tivoli Netcool/OMNIbus, les objets du serveur ObjectServer suivants sont ajoutés pour prendre en charge l'opération SAE :

- Tables de base de données de l'utilisation de l'application SAE : precision.entity_service, precision.service_details, precision.service_affecting_events
- Zone NmosEntityId dans la table alerts.status
- Groupe de déclencheurs sae et déclencheur update_service_affecting_events

  L'automatisation du déclencheur update_service_affecting_events ne fonctionne qu'avec Tivoli Netcool/OMNIbus version 7.0 ou ultérieure.
- Outils de liste d'événements permettant de générer les événements affectés au service sous UNIX et Windows

A partir de la liste d'événements de Tivoli Netcool/OMNIbus, vous pouvez surveiller les événements affectés par un service de la manière suivante :

- Pour afficher les événements sous-jacents (par exemple, linkDowns) associés à un événement affecté par un service, sélectionnez ce dernier et cliquez dessus avec le bouton droit de la souris, puis cliquez sur **Show SAE Related Events** (Afficher les événements associés à l'événement affecté par un service) à partir du menu en incrustation. Tous les événements associés à l'événement sélectionné sont affichés dans la nouvelle fenêtre.
- Pour afficher les événements affectés par un service auxquels un événement est associé (par exemple, linkDown), sélectionnez l'événement, cliquez dessus avec le bouton droit de la souris, puis cliquez sur **Show SAE Related Services** (Afficher les services associés à l'événement affecté par un service) à partir du menu contextuel. Une liste de tous les événements affectés par un service associés s'affiche dans une nouvelle fenêtre. Par exemple, si vous avez sélectionné un événement linkDown dans la liste des événements, cette fenêtre affiche tous les services affectés par cet événement linkDown.

Exemples d'automatisation

Les exemples suivants indiquent comment créer des déclencheurs et des procédures pour certaines automatisations fréquemment exécutées.

- «Exemple : déclencheur de dédoublonnage de la table d'état»
- «Exemple : déclencheur de dédoublonnage de la table des détails», à la page 204
- «Exemple : déclencheur de nettoyage de la table des détails», à la page 204
- «Exemple : déclencheur de définition de la colonne StateChange de la table d'alertes», à la page 204
- «Exemple : déclencheur de suppression des lignes en clair», à la page 204
- «Exemple : déclencheur d'envoi de notifications par courrier électronique pour les alertes critiques», à la page 205
- «Exemple : déclencheur pour tronquer un fichier», à la page 205
- «Exemple : procédure d'insertion d'une entrée de journal pour les déclencheurs» , à la page 206

Exemple : déclencheur de dédoublonnage de la table d'état

Ce déclencheur de base de données intercepte une tentative de réinsertion sur la table alerts.status et incrémente le compteur pour indiquer qu'une nouvelle ligne est arrivée dans le serveur ObjectServer. Il définit également la zone LastOccurrence.

```
create or replace trigger deduplication
group default_triggers
priority 1
comment 'Deduplication processing for ALERTS.STATUS'
before reinsert on alerts.status
for each row
begin
set old.Tally = old.Tally + 1;
set old.LastOccurrence = new.LastOccurrence;
set old.StateChange = getdate();
set old.InternalLast = getdate();
set old.Summary = new.Summary;
set old.AlertKey = new.AlertKey;
if ((old.Severity = 0) and (new.Severity > 0))
  then set old.Severity = new.Severity;
end if;
end;
```

Exemple : déclencheur de dédoublonnage de la table des détails

Ce déclencheur de base de données intercepte une tentative de réinsertion sur la table alerts.details.

```
create or replace trigger
deduplicate_details
group default_triggers
priority 1
comment 'Deduplicate rows on alerts.details'
before reinsert on alerts.details
for each row
begin
cancel; -- Do nothing. Allow the row to be discarded
end;
```

Exemple : déclencheur de nettoyage de la table des détails

Ce déclencheur temporel supprime régulièrement les entrées de la table alerts.details lorsqu'il n'existe aucune entrée correspondante dans la table alerts.status.

```
create or replace trigger
clean_details_table
group default_triggers
priority 1
comment 'Housekeeping cleanup of ALERTS.DETAILS' every 60 seconds
begin
delete from alerts.details
  where Identifier not in (select Identifier from alerts.status);
end;
```

Exemple : déclencheur de définition de la colonne StateChange de la table d'alertes

En cas de modification d'une ligne de la table alerts.status, ce déclencheur de base de données met à jour la colonne StateChange afin d'horodater la modification.

```
create or replace trigger state_change
group default_triggers
priority 1
comment 'State change processing for ALERTS.STATUS'
before update on alerts.status
for each row
begin
 set new.StateChange = getdate;
end;
```

Exemple : déclencheur de suppression des lignes en clair

Ce déclencheur temporel permet de supprimer toutes les lignes en clair (`Severity = 0`) de la table alerts.status qui n'ont pas été modifiées dans les deux dernières minutes.

```
create or replace trigger delete_clears
group default_triggers
priority 1
comment 'Delete cleared alerts over 2 minutes old every 60 seconds'
every 60 seconds
begin
delete from alerts.status where Severity = 0 and StateChange < (getdate() - 120);
end;
```

Exemple : déclencheur d'envoi de notifications par courrier électronique pour les alertes critiques

Ce déclencheur temporel envoie un message électronique en appelant une procédure externe, si des alertes critiques ne sont pas reconnues dans les 30 minutes.

```
create or replace trigger mail_on_critical
group default_triggers
enabled false
priority 1
comment 'Courrier électronique envoyé à propos d'alertes critiques dont la réception
est accusée après 30 minutes. REMARQUE : cet outil est spécifique à
UNIX sauf si un logiciel de courrier NT équivalent est disponible.'
every 10 seconds
begin
    for each row critical in alerts.status where critical.Severity = 5 and
      critical.Grade < 2 and critical.Acknowledged = 0 and
      critical.LastOccurrence <= ( getdate() - (60*30) )
    begin
      execute send_email( critical.Node, critical.Severity, 'Netcool Email',
      'root@localhost', critical.Summary, 'localhost');
      update alerts.status via critical.Identifier set Grade=2;
    end;
end;
```

La procédure externe `send_email` est déclarée comme suit et appelle l'utilitaire **`nco_mail`** :

```
create or replace procedure send_email
(in node character(255), in severity integer, in subject character(255),
 in email character(255), in summary character(255), in hostname character(255))
executable '$NCHOME/omnibus/utils/nco_mail' host 'hostname' user 0 group 0
arguments '\'' + node + '\'', severity, '\'' + subject + '\'',
'\'' + email + '\'', '\'' + summary + '\'';
```

Cet exemple affiche également la manière de transmettre des chaînes de texte à un exécutable. Les chaînes doivent être placées entre guillemets, lesquels doivent être échappés avec des barres obliques inversées. Tous les guillemets de cet exemple sont des guillemets simples.

Exemple : déclencheur pour tronquer un fichier

Lorsqu'un signal système ou défini par l'utilisateur est émis, ce déclencheur de signal tronque un fichier. Ce déclencheur est utile si vous voulez que des fichiers soient tronqués lorsque le serveur ObjectServer démarre, par exemple si des fichiers ont été archivés après le dernier arrêt anormal et ne sont plus nécessaires.

L'exemple de déclencheur suivant tronque un fichier lorsqu'un déclencheur de signal est émis :

```
create trigger exemple_déclencleur
group default_triggers
priority 1
on signal exemple_signal
begin
    alter file exemple_fichier truncate;
end;
```

Où *exemple_déclencheur* est le nom du déclencheur, *exemple_signal* est le nom du signal et *exemple_fichier* est le nom du fichier.

Exemple : procédure d'insertion d'une entrée de journal pour les déclencheurs

La procédure `jinsert` permet d'insérer des lignes dans la table alerts.journal. Les automatisations qui demandent les entrées de journalisation appellent la procédure et transmettent le numéro de série de la ligne, l'ID utilisateur de la personne qui procède à la modification (le cas échéant), l'heure à laquelle l'action a eu lieu et le texte descriptif de l'action en cours de journalisation.

```
create or replace trigger
nom_déclencheur
group default_triggers
priority 10
before delete on alerts.status
for each row
begin
execute jinsert( old.Serial, %user.user_id, getdate(), 'chaîne');
end;
```

Dans cette automation, *nom_déclencheur* et *chaîne* sont des variables représentant respectivement le nom du déclencheur et le texte de l'entrée de journal.

Référence associée:

«ALTER FILE», à la page 354
La commande `ALTER FILE` permet de modifier la configuration d'un fichier ObjectServer existant.

Administration des utilisateurs, des groupes et des rôles

Vous pouvez utiliser les commandes SQL pour organiser les collectes d'utilisateurs en groupes, puis affecter les rôles à chaque groupe pour contrôler l'accès aux objets du serveur ObjectServer. Vous pouvez créer, modifier et supprimer des utilisateurs, groupes et rôles.

Les autorisations contrôlent l'accès aux objets et aux données du serveur ObjectServer. La combinaison d'une ou de plusieurs autorisations dans des rôles permet de gérer l'accès rapidement et efficacement.

Chaque utilisateur est affecté à un ou plusieurs groupes. Vous pouvez ensuite affecter aux groupes l'autorisation d'effectuer des actions sur les objets de base de données en accordant un ou plusieurs rôles au groupe. Vous pouvez créer des regroupements logiques, tels que des utilisateurs root ou des administrateurs système, des regroupements physiques, tels que le centre d'opérations réseau de Londres ou de New York ou d'autres regroupements visant à simplifier la configuration de la sécurité.

Par exemple, la création d'automatisations nécessite de connaître les opérations Tivoli Netcool/OMNIbus et le mode de configuration d'un serveur ObjectServer particulier. Généralement, vous ne souhaitez pas autoriser tous vos utilisateurs à créer ou modifier des automatisations. Une solution consiste à créer un rôle nommé AutoAdmin avec des autorisations permettant de créer et de modifier des déclencheurs, de déclencher des groupes, des fichiers, des procédures SQL, des procédures externes et des signaux. Vous pouvez ensuite accorder ce rôle à un groupe d'administrateurs qui créera et mettra à jour les déclencheurs.

Une fois un rôle créé, vous devez affecter des autorisations au rôle à l'aide de la commande `GRANT`. Vous pouvez ensuite utiliser la commande `GRANT ROLE` pour affecter le rôle à un ou plusieurs groupes. Tous les utilisateurs qui sont membres du groupe sont automatiquement affectés aux autorisations définies pour ce rôle.

Les groupes et les rôles par défaut des opérateurs et administrateurs de gestion de réseau sont définis dans le script SQL `security.sql`. Vous pouvez également utiliser ce script en tant que modèles pour créer vos propres groupes et rôles.

Tâches associées:

«Gestion des autorisations avec des utilisateurs, des groupes, des rôles et des filtres de restriction», à la page 83
Une *autorisation* est la vérification des droits d'affichage et de modification des informations.

Référence associée:

«Commandes SQL ObjectServer»
SQL ObjectServer fournit des commandes DDL (langage de définition de données) et DML (langage de manipulation des données) destinées à la création et à la modification des objets et données ObjectServer.

Commandes SQL ObjectServer

SQL ObjectServer fournit des commandes DDL (langage de définition de données) et DML (langage de manipulation des données) destinées à la création et à la modification des objets et données ObjectServer.

Vous pouvez créer de nouveaux objets ou modifier des objets existants à l'aide des fichiers de définition SQL ou de l'interface interactive SQL.

Vous pouvez modifier les données ObjectServer en utilisant l'interface interactive SQL ou l'Netcool/OMNIbus Administrator (**`nco_config`**).

Les commandes disponibles sont regroupées dans les sections suivantes :

- «Commandes d'objet et de données»
- «Commandes d'automatisation», à la page 208
- «Commandes d'utilisateur, de groupe et de rôle», à la page 209
- «Autres commandes», à la page 210

Pour plus d'informations sur les commandes répertoriées ici, voir *Annexe B : Informations de référence sur les commandes SQL ObjectServer* dans le manuel *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Commandes d'objet et de données

Le tableau suivant répertorie les commandes DDL et DML disponibles dans le langage SQL ObjectServer.

Tableau 57. Commandes d'objet SQL ObjectServer

Objet ObjectServer	Commandes SQL
serveur ObjectServer	`ALTER SYSTEM`

Database	`CREATE DATABASE`
	`DROP DATABASE`
	`SET DATABASE`
	`USE DATABASE`

Tableau 57. Commandes d'objet SQL ObjectServer (suite)

Objet ObjectServer	Commandes SQL
Table	ADD COLUMN ALTER COLUMN ALTER TABLE CREATE TABLE DELETE DESCRIBE DROP COLUMN DROP TABLE INSERT SELECT (scalaire) SELECT (agrégation) SELECT (GROUP BY) SVC UPDATE
Index	CREATE INDEX DROP INDEX
Vue	CREATE VIEW DESCRIBE DROP VIEW SELECT (scalaire) SELECT (agrégation) SELECT (GROUP BY) SVC
Filtre de restriction	CREATE RESTRICTION FILTER DROP RESTRICTION FILTER
File	ALTER FILE CREATE FILE DROP FILE WRITE INTO

Commandes d'automatisation

Le tableau suivant répertorie les commandes d'automatisation disponibles dans le langage SQL ObjectServer.

Tableau 58. Commandes d'automatisation SQL ObjectServer

Objet d'automatisation	Commandes SQL

Groupe de déclencheurs	`ALTER TRIGGER GROUP`
	`CREATE TRIGGER GROUP`
	`DROP TRIGGER GROUP`

Déclencheur de base de données	`ALTER TRIGGER`
	`CREATE TRIGGER` (déclencheur de base de données)
	`DROP TRIGGER`

Déclencheur temporel	`ALTER TRIGGER`
	`CREATE TRIGGER` (déclencheur temporel)
	`DROP TRIGGER`

Déclencheur de signal	`ALTER TRIGGER`
	`CREATE TRIGGER` (déclencheur de signal)
	`DROP TRIGGER`

Signal défini par l'utilisateur	`CREATE SIGNAL`
	`DROP SIGNAL`
	`RAISE SIGNAL`

Signal système	Les signaux système ne sont pas destinés à être modifiés.

Procédure SQL	`CREATE PROCEDURE` (procédures SQL)
	`DROP PROCEDURE`
	`EXECUTE PROCEDURE`

Procédure externe	`CREATE PROCEDURE` (procédures externes)
	`DROP PROCEDURE`
	`EXECUTE PROCEDURE`

Commandes d'utilisateur, de groupe et de rôle

Le tableau suivant répertorie les commandes d'utilisateur, de groupe et de rôle disponibles dans le langage SQL ObjectServer.

Tableau 59. Commandes SQL d'utilisateur, de groupe et de rôle

Objet	Commande SQL

User (Utilisateur)	`ALTER USER`
	`CREATE USER`
	`DROP USER`

Tableau 59. Commandes SQL d'utilisateur, de groupe et de rôle (suite)

Objet	Commande SQL

Groupe	`ALTER GROUP`
	`CREATE GROUP`
	`DROP GROUP`

Rôle	`ALTER ROLE`
	`CREATE ROLE`
	`DROP ROLE`
	`GRANT` (Affecte des droits d'accès à un rôle)
	`GRANT ROLE` (Affecte des rôles à des groupes)
	`REVOKE` (Annule les droits d'accès associés à un rôle)
	`REVOKE ROLE` (Annule les rôles d'un groupe)

Autres commandes

Le tableau suivant répertorie les commandes restantes disponibles en langage SQL ObjectServer.

Tableau 60. Autres commandes SQL

Objet	Commandes SQL
Client IDUC	`IDUC EVTFT` (Active la notification d'événement accéléré) `IDUC FLUSH` (Envoie des notifications aux clients ObjectServer) `IDUC SNDMSG` (Envoie des messages aux clients AEN (notification d'événement accéléré))

Clause SQL	`CASE WHEN`
	`FOR`
	`FOR EACH ROW`
	`IF THEN ELSE`
	`SET`

Syntaxe SQL	`CHECK STATEMENT` (Vérifie la syntaxe SQL)

Concepts associés:

Annexe B, «Informations de référence sur les commandes SQL ObjectServer», à la page 353
Informations de référence sur les commandes SQL ObjectServer.

Mots réservés

Dans le serveur ObjectServer, certains mots sont réservés en tant que mots clés SQL ou ObjectServer.

Vous n'êtes pas autorisé à utiliser ces mots réservés comme noms d'objet dans SQL ObjectServer. Les mots réservés sont répertoriés par ordre alphabétique dans les tables suivantes.

A à E

Tableau 61. A à E

A	B	C	D	E
ACTCMD	BACKUP	CACHE	DATABASE	EACH
ADD	BEFORE	CALL	DATE[TIME]	EDGE
AFTER	BEGIN	CANCEL	DEBUG	ELSE
ALL	BETWEEN	CASE	DEC[EMBER]	ELSEIF
ALTER	BI DIRECTIONAL	CHAR[ACTER]	DECLARE	EMPTY
AND	BINARY	CHECK	DEFERRED	END
ANY	BIND	CHECK POINTING	DELAYED	ENCRYPTED
APR[IL]	BOOL[EAN]	COLUMN	DELETE	EVALUATE
ARGUMENTS	BREAK	COMMENT	DESC	EVENT
ARRAY	BY	COMMIT	DESCENT	EVERY
AS		CONN[ECTION]	DESCRIBE	EVTFT
ASC		COUNT	DETACHED	EXECUTABLE
ASSIGN		CREATE	DISABLE	EXEC[UTE]
AUG[UST]		CURRENT	DIST	EXTENSION
AUTHORIZE | AUTHORISE			DISTINCT	EXTERNAL
AVERAGE | AVG			DO	
			DOUBLE	
			DROP	

F à J

Tableau 62. F à J

F	G	H	I	J
FALSE	GET	HARD	ID	JAN[UARY]
FANP	GETIDUC	HAVING	IDUC	JOIN
FEB[RUARY]	GRANT	HIDDEN	IF	JUL[Y]
FILE	GROUP	HOST	IMMEDIATE	JUN[E]
FILTER		HOURS	IN	
FLUSH			INCLUDING	
FOR			INCR	

Tableau 62. F à J (suite)

F	G	H	I	J
FORMAT			INCREMENT	
FRI[DAY]			INITIAL	
FROM			INSERT	
FULL			INT[EGER]	
			INT[EGER]64	
			INTO	
			ISQL	

L à P (aucun mot réservé ne commence par K)

Tableau 63. L à P (aucun mot réservé ne commence par K)

L	M	N	O	P
LEAVE	MAR[CH]	NAMING	OCT[OBER]	PAM
LIKE	MAX	NEXT	OF	PASSWORD
LIMIT	MAXFILES	NO	ON	PERSISTENT
LINK	MAXSIZE	NODEFAULT	ONCE	PRIMARY
LOAD	MAY	NOMODIFY	ONLY	PRIORITY
LOCK[PH]	MEMSTORE	NOT	OPTION	PRIVILEGE
LOGIN	MESSAGE	NOTIFY	OR	PROCEDURE
	METRIC	NOV[EMBER]	ORDER	PROP[S]
	MIN		OUT	PROTECT
	MINUTES			PUBLISH
	MON[DAY]			

Q à U

Tableau 64. Q à U

Q	R	S	T	U
QUERY	RAISE	SAT[URDAY]	TABLE	UNIDIRECT IONAL
	REAL	SAVE	TEMPORAL	UNION
	REGISTER	SECONDS	TEMP[ORARY]	UNIQUE
	REINSERT	SELECT	THEN	UNLOAD
	REMOVE	SELF	THU[RSDAY]	UNREGISTER
	REPLACE	SEND	TIME	UNSIGNED
	RESTRICTION	SEP[TEMBER]	TO	UNSIGNED64
	RESYNC	SESSION	TOKEN	UNSUBSCRIBE
	RETRY	SET	TOP	UNTIL
	REVOKE	SHORT	TRANS [ACTION]	UPDATE
	ROLE	SHOW	TRANSIENT	UPDATING
	ROW	SIGNAL	TRIGGER	USE

Tableau 64. Q à U (suite)

Q	R	S	T	U
	ROWOF	SKIP	TRUE	USER
		SNDMSG	TRUNCATE	UTC
		SOFT	TUE[SDAY]	
		SQL	TYPEOF	
		STATEMENT		
		STORE		
		SUBSCRIBE		
		SUM		
		SUN[DAY]		
		SVC		
		SYNC		
		SYSTEM		

V à Y (aucun mot réservé ne commence par Z)

Tableau 65. V à Y (aucun mot réservé ne commence par Z)

V	W	X	Y
VALUES	WAIT	XST	YES
VARCHAR[ACTER]	WED[NESDAY]		
VERBOSE	WHEN		
VIA	WHERE		
VIEW	WIDTH		
VIRTUAL	WITH		
	WORK		
	WRITE		

Conventions de dénomination SQL ObjectServer

Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

Les noms des objets et identificateurs ObjectServer sont sensibles à la casse.

- Serveurs ObjectServer :

  Le nom d'un serveur ObjectServer doit se composer d'un maximum de 29 lettres en majuscules. Il ne doit pas commencer par un entier.
- Objets ObjectServer :

  Les noms des objets ObjectServer doivent commencer par une lettre en majuscule ou minuscule, suivie de lettres en majuscules ou minuscules ou de caractères de soulignement (_). La longueur maximale d'un nom est de 40 caractères. Les noms d'objet doivent être uniques dans le jeu d'objets de leur type.
- Utilisateurs, groupes et rôles :

  Les noms d'utilisateur, de groupe et de rôle consistent en une chaîne de texte incluse entre guillemets (maximum : 64 caractères).

Syntaxe SQL

Vous devez vous conformer à la syntaxe SQL ObjectServer lorsque vous exécutez des commandes SQL à partir de l'interface interactive SQL ou dans des fichiers SQL.

La table suivante décrit la syntaxe utilisée dans les commandes SQL ObjectServer.

Tableau 66. Syntaxe SQL

Syntaxe	Description
`{ a | b }`	Dans la syntaxe SQL, les accolades entourent deux choix alternatifs ou plus, séparés par des barres verticales.
`[ ]`	Dans la syntaxe SQL, les crochets indiquent un élément ou une clause en option. Plusieurs éléments ou clauses sont séparés par des barres verticales.
`|`	Dans la syntaxe SQL, les barres verticales séparent deux ou plusieurs éléments de syntaxe.
`...`	Dans la syntaxe SQL, les ellipses indiquent que l'élément précédent peut être répété. Sauf mention contraire, la répétition est illimitée.
`,...`	Dans la syntaxe SQL, les ellipses précédées d'une virgule indiquent que l'élément précédent peut être répété. Chaque élément répété est séparé du précédent par une virgule. Sauf mention contraire, la répétition est illimitée.
a	Dans la syntaxe SQL, un élément souligné indique une option par défaut.
`( )`	Dans la syntaxe SQL, les parenthèses qui apparaissent dans la syntaxe d'instruction font partie de la syntaxe et doivent (sauf mention contraire) être saisies comme suit.

L'exemple suivant de commande `CREATE TABLE` montre comment est utilisée la syntaxe. Les valeurs variables apparaissent en italique.

```
CREATE TABLE [nom_base_de_données.]nom_table
PERSISTENT | VIRTUAL
(nom_colonne type_données [ PRIMARY KEY | NODEFAULT | NOMODIFY | HIDDEN ],...
[, PRIMARY KEY(nom_colonne,...) ] );
```

Remarque : Les mots clés SQL ne sont pas sensibles à la casse et peuvent être en majuscules et/ou en minuscules.

Opérateurs

Vous pouvez utiliser les *opérateurs* pour calculer les valeurs à partir d'éléments de données.

Un opérateur traite (ajoute, soustrait, etc) un élément ou des éléments de données. Le calcul se base sur les *opérandes*. Ensemble, les opérateurs et les opérandes forment des *expressions*. Dans l'expression 7 + 3, le symbole plus (+) est l'opérateur, tandis que 7 et 3 sont les opérandes.

Les opérateurs peuvent être monadiques ou binaires. Les opérateurs monadiques n'agissent que sur une seule opérande. Par exemple, le signe moins (-) peut être utilisé pour indiquer la négation. Les opérateurs binaires agissent sur deux opérandes. Par exemple, le même opérateur moins (-) peut être utilisé pour soustraire une opérande d'une autre opérande.

Certains opérateurs, tels que l'opérateur plus (+), sont polymorphes et peuvent avoir différentes significations selon le contexte. Par exemple, vous pouvez utiliser l'opérateur plus (+) pour additionner deux nombres (7+3) ou pour concaténer deux chaînes ('The ObjectServer ' + 'started').

Les opérateurs utilisés dans le serveur ObjectServer SQL sont divisés en deux catégories :

- Opérateurs mathématiques et de chaîne
- Opérateurs de comparaison binaires
- Opérateurs de comparaison de liste
- Opérateurs logiques

Concepts associés:

«Expressions», à la page 227
Une expression est une combinaison syntaxique de valeurs et d'opérations associées pour calculer de nouvelles valeurs. Les expressions peuvent être simples ou complexes.

Opérateurs mathématiques et de chaîne

Utilisez les opérateurs mathématiques pour ajouter, soustraire, multiplier et diviser les opérandes numériques dans les expressions. Utilisez les opérateurs de chaîne pour manipuler les chaînes de caractères (types de données VARCHAR et CHAR).

La table suivante décrit les opérateurs mathématiques pris en charge par le serveur ObjectServer.

Tableau 67. Opérateurs mathématiques

Opérateur	Description	Exemple
+ -	Opérateurs monadiques indiquant une opérande positive ou négative.	`SELECT * FROM london.status WHERE Severity = -1;`
* /	Opérateurs binaires utilisés pour multiplier (*) ou diviser (/) deux opérandes.	`SELECT * FROM london.status WHERE Tally * Severity > 10;`
+ -	Opérateurs binaires utilisés pour ajouter (+) ou soustraire (-) deux opérandes.	`SELECT * FROM london.status WHERE Severity = Old_Severity - 1;`

La table suivante décrit l'opérateur de chaîne pris en charge par le serveur ObjectServer.

Tableau 68. Opérateur de chaîne

Opérateur	Description	Exemple
+	Opérateur binaire utilisé pour concaténer deux chaînes.	`UPDATE mydb.mystatus SET Location = Node + NodeAlias;`

Opérateurs de comparaison binaires

Utilisez ces opérateurs pour comparer l'égalité et l'inégalité des valeurs numériques et de chaîne.

La table suivante décrit les opérateurs de comparaison pris en charge par le serveur ObjectServer.

Tableau 69. Opérateurs de comparaison

Opérateur	Description	Exemple
=	Tests d'égalité.	`SELECT * FROM london.status WHERE Severity = 3;`
!= <>	Tests d'inégalité.	`SELECT * FROM london.status WHERE Severity <> 1;`
< > <= >=	Tests de supériorité à (>), d'infériorité à (<), de supériorité ou d'égalité à (>=) ou d'infériorité ou d'égalité à (<=). Ces opérateurs effectuent des comparaisons de chaîne sensibles à la casse. Dans les comparaisons standard sensibles à la casse ASCII, les majuscules viennent avant les minuscules.	`SELECT * FROM london.status WHERE Severity > 5;`
%= %!= %<>	Tests d'égalité (%=) ou d'inégalité (%!=, %<>) entre les chaînes, en ignorant la casse. Pour être égales, les chaînes doivent contenir les mêmes caractères dans le même ordre. Toutefois, la capitalisation des lettres n'est pas prise en compte.	`SELECT * FROM london.status WHERE Location %= 'New York';`
%< %> %<= %>=	Compare la relation lexicographique entre deux chaînes, en ignorant la casse. Cette comparaison détermine si les chaînes se situent alphabétiquement avant (%<) ou après (%>) d'autres chaînes. Vous pouvez également rechercher des chaînes inférieures ou égales à (%<=)/supérieures ou égales à (%>=) d'autres chaînes. Par exemple, `aaa` précède `AAB` car `aaa` est alphabétiquement inférieur à (situé avant) `AAB` lorsque la casse n'est pas prise en compte.	`SELECT * FROM london.status WHERE site_code %< 'UK3';`

Tableau 69. Opérateurs de comparaison (suite)

Opérateur	Description	Exemple
[NOT] LIKE	L'opérateur LIKE effectue des comparaisons de chaîne. La chaîne suivant l'opérateur LIKE, qui peut être le résultat d'une expression régulière, est le modèle par rapport auquel l'expression de colonne est comparée. Une expression régulière peut inclure la syntaxe de correspondance de modèle décrite dans *IBM Tivoli Netcool/OMNIbus User's Guide*. Le mot clé NOT inverse le résultat de la comparaison.	`SELECT * FROM london.status WHERE Summary LIKE 'down';` Le résultat contient toutes les lignes dans lesquelles `Summary` contient la sous-chaîne `down`.

Les opérateurs de comparaison LIKE et NOT LIKE permettent de comparer la correspondance de modèle d'expression régulière de la chaîne à l'expression de la colonne. Les expressions régulières sont des séquences d'*atomes* composés de caractères et métacaractères normaux. Un atome est un caractère ou un modèle d'un ou plusieurs caractères entre parenthèses. Les caractères normaux comprennent les lettres majuscules et minuscules, ainsi que des chiffres. Les métacaractères sont des caractères non alphabétiques possédant des significations spéciales dans des expressions régulières. Le serveur ObjectServer prend en charge deux types de bibliothèque d'expressions régulières :

- NETCOOL : utilisez cette bibliothèque par défaut pour le traitement des caractères à octet unique.
- TRE : cette bibliothèque permet d'utiliser la syntaxe d'expression régulière étendue POSIX 1003.2 et prend en charge les langues de caractères à octet unique et à octets multiples.

Pour plus d'informations sur ces bibliothèques, ainsi que des descriptions des formats de syntaxe d'expression régulière et des exemples d'utilisation, voir *IBM Tivoli Netcool/OMNIbus User's Guide*.

Opérateurs de comparaison de liste

Utilisez les opérateurs de comparaison de liste pour comparer une valeur et une liste de valeurs.

Les conditions utilisant les opérateurs de comparaison de liste utilisent les opérateurs de comparaison binaires avec les opérateurs logiques (ANY, ALL, IN ou NOT IN).

La syntaxe d'une expression de comparaison de liste est soit :

```
expression opérateur_comparaison { ANY | ALL } ( expression,... )
```

soit :

```
expression [ NOT ] IN ( expression,... )
```

Si vous utilisez le mot clé ANY, la condition de comparaison de liste est définie sur `TRUE` si la comparaison de l'expression de gauche par rapport à l'expression de

droite retourne TRUE pour l'une des valeurs. Si vous utilisez le mot clé ALL, la condition de comparaison de liste est définie sur TRUE si la comparaison de l'expression de gauche par rapport à l'expression de droite retourne TRUE pour toutes les valeurs.

Une comparaison IN retourne les mêmes résultats que la comparaison =ANY. Une comparaison NOT IN retourne les mêmes résultats que la comparaison <>ALL.

Restriction : Les opérateurs ANY et ALL ne sont pas pris en charge dans les sous-requêtes.

Exemple

La requête suivante retourne les lignes dans lesquelles Severity - 1 est égal à la valeur de Old_Severity ou le numéro 5.

```
select * from mystatus where Severity - 1 IN (Old_Severity, 5)
```

Concepts associés:

«Opérateurs de comparaison binaires», à la page 216
Utilisez ces opérateurs pour comparer l'égalité et l'inégalité des valeurs numériques et de chaîne.

«Opérateurs logiques»
Vous pouvez utiliser des opérateurs logiques sur des valeurs booléennes pour former les expressions qui aboutissent au résultat TRUE ou FALSE.

Opérateurs logiques

Vous pouvez utiliser des opérateurs logiques sur des valeurs booléennes pour former les expressions qui aboutissent au résultat TRUE ou FALSE.

Le serveur ObjectServer prend en charge les opérateurs suivants :

- NOT
- AND
- OR

Vous pouvez combiner des comparaisons à l'aide d'opérateurs logiques.

Les tables de vérité suivantes affichent les résultats des opérations logiques sur les valeurs booléennes. Dans les tables de vérité suivantes, A et B représentent une valeur ou une expression.

Une expression NOT retourne TRUE uniquement si son entrée est FALSE, comme illustré dans la table suivante.

Tableau 70. Table de vérité pour opérateur NOT

A	NOT A
FALSE	TRUE
TRUE	FALSE

Une expression AND est vraie uniquement si toutes ses entrées sont TRUE, comme illustré dans la table suivante.

Tableau 71. Table de vérité pour opérateur AND

A	B	A ET B
FALSE	FALSE	FALSE
FALSE	TRUE	FALSE
TRUE	FALSE	FALSE
TRUE	TRUE	TRUE

Une expression OR est `TRUE` si l'une de ses entrées est `TRUE`, comme illustré dans la table suivante.

Tableau 72. Table de vérité pour opérateur OR

A	B	A OU B
FALSE	FALSE	FALSE
FALSE	TRUE	TRUE
TRUE	FALSE	TRUE
TRUE	TRUE	TRUE

Exemple

La requête suivante combine des comparaisons à l'aide d'opérateurs logiques :

```
SELECT * from alerts.status where Node = 'node1' and Severity > 4 and
Summary like 'alert on .*'
```

Exemple

La requête suivante retourne toutes les lignes de la table `t1` où la valeur `col1` n'est pas égale à 0 :

```
select * from t1 where NOT(col1 = 0);
```

Opérateurs bit à bit

Utilisez les opérateurs bit à bit pour former des expressions testant ou manipulant les opérandes au niveau binaire.

Le serveur ObjectServer prend en charge les opérateurs bit à bit sur les types de données suivants : entier, non signé, booléen, entier 64 bits et non signé 64 bits. Si vous tentez d'effectuer des opérations au niveau du bit sur des types de données non pris en charge, une erreur «type non valide» est générée.

Le tableau suivant décrit les opérateurs bit à bit pris en charge par le serveur ObjectServer :

Tableau 73. Opérateurs bit à bit

Opérateur	Description	Exemple
&	AND au niveau du bit	Dans l'exemple suivant, le résultat est 0 : `SET ValeurA = 1; SET ValeurB = 0;  SET result = ValeurA & ValeurB;`

Tableau 73. Opérateurs bit à bit (suite)

Opérateur	Description	Exemple
|	OR au niveau du bit	Dans l'exemple suivant, le résultat est 1 : `SET ValeurA = 1; SET ValeurB = 0; SET result = ValeurA | ValeurB;`
~	NOT au niveau du bit	Dans l'exemple suivant, le résultat est -1 : `SET ValeurA = 0; SET result = ~ValeurA;`

Priorité de l'opérateur

Si une expression contient plusieurs opérateurs, le serveur ObjectServer utilise la priorité de l'opérateur pour déterminer l'ordre dans lequel évaluer l'expression.

Les opérateurs sont déterminés des opérateurs possédant la priorité la plus haute à ceux possédant la priorité la plus basse. Par exemple, l'opérateur binaire plus (+) a une priorité inférieure à l'opérateur de multiplication (*). Dans l'expression `3 + 5 * 2`, le résultat est 13 car 5 est multiplié par 2 avant que le résultat (10) ne soit additionné à 3.

Utilisez les parenthèses dans une expression pour modifier l'ordre dans lequel les éléments sont évalués. Le contenu des parenthèses est toujours évalué avant tout élément situé en dehors de celles-ci. Dans l'expression `(3 + 5) * 2`, le résultat est 16 car 3 est additionné à 5 avant que le résultat (8) ne soit multiplié par 2.

Si des opérateurs possèdent des priorités égales, ils sont évalués dans l'ordre de gauche à droite. La table suivante indique l'ordre de priorité de tous les opérateurs du serveur ObjectServer.

Tableau 74. Priorité de l'opérateur

Priorité la plus haute
Monadique `+ - ~`
Mathématique `* /`
Binaire `+ - & |`
Opérateurs de comparaison (y compris comparaisons de liste)
NOT
AND
OR
Priorité la plus basse

Fonctions

Une fonction traite un élément de données ou des éléments dans une commande SQL, puis retourne une valeur.

La syntaxe d'une fonction se présente comme suit :

```
function(
opérande,...)
```

Les parenthèses sont facultatives si la fonction ne possède pas d'opérandes.

La table suivante décrit les fonctions SQL prises en charge par le serveur ObjectServer.

Tableau 75. Fonctions SQL ObjectServer

Fonction	Description	Exemple
`array_len(matrice)`	Retourne le nombre d'éléments dans une matrice. Cette fonction ne peut être utilisée que dans les procédures ou les déclencheurs.	Si la matrice `myarray` possède dix éléments, `array_len(myarray)` retourne `10`.
`ceil(réel)`	Prend un argument réel et retourne la plus petite valeur intégrale qui n'est pas inférieure à l'argument.	`ceil(2.01)` retourne `3.000000`
`current_user()`	Ne prend aucun argument et retourne le nom de l'utilisateur qui exécute la commande SQL.	`update alerts.status set Summary = Summary + ' modified by ' + current_user;`
`current_userid()`	Ne prend aucun argument et retourne l'ID de l'utilisateur qui exécute la commande SQL.	`select Node, Summary from alerts.status where OwnerUID = current_userid()`
`dayasnum(heure)`	Prend un argument temporel et extrait le jour de la semaine sous forme d'entier. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	`select dayasnum(LastOccurrence) from mytab;` Dimanche correspond à `0`, lundi correspond à `1`, etc.
`dayname(heure)`	Prend un argument temporel et retourne le nom du jour. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	`select dayname(LastOccurrence) from mytab;` La sortie est `Monday`, `Tuesday`, etc.
`dayofmonth(heure)`	Prend un argument temporel et extrait le jour du mois sous forme d'entier. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	`select dayofmonth(LastOccurrence) from mytab;`
`dayofweek(heure)`	Prend un argument temporel et extrait le jour de la semaine sous forme d'entier. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	`select dayofweek(LastOccurrence) from mytab;` A la différence de `dayasnum`, dimanche correspond à `1`, lundi correspond à `2`, etc.
`getdate()`	Ne prend aucun argument et retourne la date et l'heure actuelles sous la forme d'une valeur de temps universel coordonné (nombre de secondes depuis le 1er janvier 1970).	Pour retourner toutes les lignes de la table `alerts.status` dont l'ancienneté est supérieure à dix minutes : `select Summary, Severity from alerts.status where LastOccurrence < getdate - 600;`
`getenv(chaîne)`	Retourne la valeur de la variable d'environnement spécifiée sous forme de chaîne.	`getenv('NCHOME')` retourne un nom de répertoire, (par exemple, `/opt/netcool`).
`get_prop_value(chaîne)`	Retourne la valeur de la propriété d'environnement spécifiée sous forme de chaîne.	`get_prop_value('Name')` retourne le nom du serveur ObjectServer (par exemple, `NCOMS`).

Tableau 75. Fonctions SQL ObjectServer (suite)

Fonction	Description	Exemple
`getservername()`	Ne prend aucun argument et retourne le nom du serveur ObjectServer sous forme de chaîne.	`select LastOccurrence from alerts.status where ServerName = getservername();`
`hourofday(`*heure*`)`	Prend un argument temporel et extrait l'heure du jour sous forme d'entier. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	`select hourofday(LastOccurrence) from mytab;`
`instance_of(`*classe*`, `*classe_parent*`)`	Retourne `TRUE` si *classe* est une sous-classe de *classe_parent* ou si ces valeurs sont égales, à l'aide de la hiérarchie définie dans la table master.class_membership. Sinon, la valeur retournée est `FALSE`. Les variables *classe* et *classe_parent* peuvent être des chaînes ou des entiers.	`select Node, Summary, AlertGroup, Server from alerts.status where instance_of(Class, 'DB2') = true;`
`is_env_set(`*chaîne*`)`	Retourne 1 si la variable d'environnement spécifiée est définie ; sinon, la valeur `0` est retournée.	Lorsque la variable d'environnement `NCHOME` est définie, `is_env_set('NCHOME')` retourne `1`.
`log_2(`*réel*`)`	Prend un argument réel positif et retourne le logarithme à la base 2.	`log_2(4.0)` retourne `2.000000`
`lower(`*chaîne*`)`	Convertit un argument de chaîne de caractères en caractères minuscules.	`lower('LIMA')` retourne `lima`
`ltrim(`*chaîne*`)`	Supprime un blanc à gauche d'une chaîne.	`ltrim(' tree')` retourne `tree`.
`minuteofhour(`*heure*`)`	Prend un argument temporel et extrait la minute de l'heure sous forme d'entier. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	`select minuteofhour(LastOccurrence) from mytab;`
`mod(`*ent1*`,`*ent2*`)`	Retourne le reste de l'entier de *ent1* divisé par *ent2*.	`mod(12,5)` retourne `2`
`monthasnum(`*heure*`)`	Prend un argument temporel et extrait le mois de l'année sous forme d'entier. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	`select monthasnum(LastOccurrence) from mytab;` Janvier correspond à `0`, février à `1`, etc..
`monthname(`*heure*`)`	Prend un argument temporel et retourne le nom du mois. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	`select monthname(LastOccurrence) from mytab;` La sortie est `January`, `February`, etc..
`monthofyear(`*heure*`)`	Prend un argument temporel et extrait le mois de l'année sous forme d'entier. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	`select monthofyear(LastOccurrence) from mytab;` A la différence de `monthasnum`, January correspond à `1`, February à `2`, etc..
`nvp_exists(` *paires_valeur_nom_chaîne*`,` *nom_chaîne*`)`	Vérifie si une paire nom_valeur existe. S'utilise avec les attributs étendus.	`nvp_exists(ExtendedAttr, 'Region')` Retourne `TRUE` si Region existe dans les attributs étendus sous la forme d'un nom de paire nom-valeur. Si le nom n'existe pas, la fonction retourne `FALSE`.

Tableau 75. Fonctions SQL ObjectServer (suite)

Fonction	Description	Exemple
chaîne nvp_get(*chaîne* *paires_nom_valeur*, *clé_chaînes*)	Extrait la valeur de *nom* dans une paire nom_valeur. Si le nom existe dans l'attribut, la fonction retourne la valeur. Si *paires_nom_valeur* est invalide, la chaîne vide est retournée et une erreur est consignée. Si la *clé* est absente, le chaîne vide est retournée ('''). Si le nom possède plusieurs entrées, la première entrée est retournée.	nvp_get(ExtendedAttr, 'Region') Retourne l'attribut Region.
chaîne nvp_set(*chaîne* *paires_nom_valeur*, *clé_chaînes*, *toute_valeur*)	Remplace ou ajoute une paire nom_valeur dans une chaîne de paire nom_valeur. Retourne une chaîne de paire nom-valeur avec la nouvelle paire nom-valeur ajoutée ou remplacée. Ajoute ou remplace des clés d'une chaîne de paire nom-valeur et retourne la nouvelle chaîne de paire nom-valeur. Une date est stockée en secondes depuis 1970. Il s'agit du format d'époque UNIX. Si *paires_nom_valeur* n'est pas une chaîne valide (c'est-à-dire, s'il s'agit d'une chaîne vide ou si elle contient des entrées non conformes au format correct), la valeur $key="$value" est retournée et une erreur est consignée. Si la clé possède plusieurs entrées, seule la première est remplacée.	ExtendedAttr = nvp_set(ExtendedAttr, 'Region', 'EMEA'); Définit l'attribut Region dans les attributs étendus.
power(*réel1*, *réel2*)	Prend deux arguments réels et retourne *réel1* à la puissance *réel2*.	power(2.0, 3.0) retourne 8.000000
rtrim(*chaîne*)	Supprime un blanc à droite d'une chaîne.	rtrim('tree ') retourne tree.
secondofminute(*heure*)	Prend un argument temporel et extrait la seconde de la minute sous forme d'entier. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	select secondofminute(LastOccurrence) from mytab;

Tableau 75. Fonctions SQL ObjectServer (suite)

Fonction	Description	Exemple

`split_multibyte(`
`message_chaîne,`
`num_bloc_ent,`
`lon_bloc_ent)`
	Retourne la chaîne multi-octets complète *support* d'un maximum de *lon_bloc_ent* octets, en commençant par l'octet *(num_bloc_ent -1) * lon_chaîne_bloc_ent*.
	
	Si la division génère un caractère multi-octets incomplet dans la chaîne cible, la fonction retourne la plus longue chaîne possible. L'appel suivant de la fonction (*lon_bloc_ent* étant identique à l'appel précédent) commence à partir du caractère qui n'a pas pu être extrait complètement.
	
	La fonction divise un caractère multi-octets en chaînes plus petites contenant uniquement les caractères multi-octets. Il s'agit principalement de stocker les chaînes importantes en plusieurs zones de base de données plus petites.
	
	La variable *lon_bloc_ent* de tous les appels à la fonction `split` de la même chaîne doit être identique.
		Voir l'exemple suivant cette table.

`substr(message_chaîne,` `posdépart_ent, lon_ent)`	Extrait une sous-chaîne, en commençant à la position spécifiée dans le deuxième paramètre pour le nombre de caractères spécifiés dans le troisième paramètre. La chaîne et indexée à partir de 1.	`substr('abcdefg', 2, 3)` retourne `bcd`, en commençant au deuxième caractère et en retournant les trois caractères suivants.

`to_char(argument [,`
`'spécification_conversion'])`
	Convertit l'argument en chaîne. L'argument peut être tout type de données, à l'exception d'une chaîne.
	
	La virgule (,) n'est obligatoire que si une chaîne de conversion est spécifiée.
	
	Si l'argument est de type temporel, vous pouvez spécifier un deuxième argument consistant en une spécification de conversion visant à mettre en forme la sortie. Ce format est défini dans la section Format date/heure de référence. Le format par défaut est `EEE MMM dd HH:mm:ss yyyy` ou, en format POSIX, `%a %b %d %T %Y`. Notez que le format POSIX avec `%` est obsolète.
		`to_char(73)` retourne `73`
		
		`to_char(FirstOccurrence)` retourne une chaîne telle que `Thu Dec 11 16:02:05 2003`
		
		`to_char(LastOccurrence, '%Y')` renvoie une chaîne telle que `2010`

Tableau 75. Fonctions SQL ObjectServer (suite)

Fonction	Description	Exemple

`to_int('argument')`	Convertit l'argument en un nombre entier. L'argument peut être tout type de données, à l'exception d'un entier.
	
	Cette fonction enlève tout blanc en début de ligne de l'argument, puis analyse la chaîne restante. L'analyse s'arrête lorsqu'elle rencontre un caractère ne pouvant être converti en caractère décimal ou lorsqu'elle atteint la fin de la chaîne, quel que soit le premier événement survenant. Lorsque l'analyse s'arrête, la fonction convertit les caractères en valeur décimale ou retourne 0 si elle n'a pas réussi à trouver des caractères convertibles en caractère décimal.
		`to_int('73')` retourne `73`
		
		`to_int('3F')` retourne `3`
		
		`to_int('UK')` retourne `0`
		
		`to_int('F3')` retourne `0`

`to_real('argument')`	Convertit l'argument en nombre réel de 64 bits. L'argument peut être tout type de données, à l'exception d'un réel.
	
	Cette fonction enlève tout blanc en début de ligne de l'argument, puis analyse la chaîne restante. L'analyse s'arrête lorsqu'elle rencontre un caractère ne pouvant être converti en caractère décimal ou lorsqu'elle atteint la fin de la chaîne, quel que soit le premier événement survenant. Lorsque l'analyse s'arrête, la fonction convertit les caractères en valeur décimale ou retourne 0 si elle n'a pas réussi à trouver des caractères convertibles en caractère décimal.
		`to_real('7.3')` retourne `7.300000`
		
		`to_real('3F')` retourne `3.000000`
		
		`to_real('UK')` retourne `0`
		
		`to_real('F3')` retourne `0`

`to_time(argument [,'spécification_conversion'])`
 	
`to_date(argument [,'spécification_conversion'])`
	Convertit l'argument en type temporel. L'argument peut être tout type de données, à l'exception d'un type temporel.
	
	La virgule (,) n'est obligatoire que si une chaîne de conversion est spécifiée.
	
	Si l'argument est de type chaîne, vous pouvez spécifier un deuxième argument consistant en une spécification de conversion visant à mettre en forme la sortie. Le format par défaut est `EEE MMM dd HH:mm:ss yyyy` ou, en format POSIX, `%a %b %d %T %Y`. Notez que le format POSIX avec `%` est obsolète.
		`update mytab set my_utc_col = to_time('Thu Dec 11 16:00:00 2003')`

Tableau 75. Fonctions SQL ObjectServer (suite)

Fonction	Description	Exemple
to_unsigned(*argument*)	Convertit l'argument en un entier non signé de 64 bits. L'argument peut être tout type de données, à l'exception d'un entier non signé de 64 bits. Cette fonction enlève tout blanc en début de ligne de l'argument, puis analyse la chaîne restante. L'analyse s'arrête lorsqu'elle rencontre un caractère ne pouvant être converti en caractère décimal ou lorsqu'elle atteint la fin de la chaîne, quel que soit le premier événement survenant. Lorsque l'analyse s'arrête, la fonction convertit les caractères en valeur décimale ou retourne 0 si elle n'a pas réussi à trouver des caractères convertibles en caractère décimal.	to_unsigned('73') retourne 73 to_unsigned(73) retourne 73 to_unsigned('UK') retourne 0 to_unsigned('F3') retourne 0
upper(*chaîne*)	Convertit un argument de chaîne de caractères en caractères majuscules.	upper('Vancouver') retourne VANCOUVER
user_in_group(UID, GID) user_in_group(UID, *chaîne_nom_groupe*) user_in_group(GID) user_in_group(*chaîne nom_groupe*)	Retourne TRUE si l'utilisateur est membre du groupe indiqué ; sinon, retourne FALSE. Cette fonction peut avoir un ou deux arguments. La forme à un seul argument utilise l'ID (UID) de l'utilisateur qui exécute la commande SQL. La forme à deux arguments requiert un ID utilisateur et un nom ou un ID de groupe (GID).	select user_in_group('Normal') from alerts.status retourne TRUE si l'utilisateur en cours est membre du groupe Normal. select Summary from alerts.status where user_in_group(OwnerGID) = true retourne toutes les alarmes qui ont été affectées à un groupe dont l'utilisateur actuel est membre.
year(*heure*)	Prend un argument temporel et extrait l'année sous forme d'entier. Si aucun argument n'est spécifié, l'argument est supposé être l'heure actuelle.	select year(LastOccurrence) from mytab;

Exemple : utilisation de split_multi-byte

```
for each row res_filter in catalog.restrictions where
  res_filter.RestrictionName = rf_users.RestrictionName
 begin
   -- Populate master.profile with the new row.
   -- Cut up the filter text into 255 byte chunks
   update master.profiles set HasRestriction = 1,
    Restrict1 = split_multibyte( res_filter.ConditionText, 1, 255),
    Restrict2 = split_multibyte( res_filter.ConditionText, 2, 255),
    Restrict3 = split_multibyte( res_filter.ConditionText, 3, 255),
    Restrict4 = split_multibyte( res_filter.ConditionText, 4, 255)
    Where UID = rf_users.GranteeID;
 end;
```

Dans cet exemple :

Restrict1 est affecté au maximum à 255 octets de res.filter.ConditionText de l'octet 1

Restrict2 est affecté au maximum à 255 octets de res.filter.ConditionText de l'octet (2-1) * 255

Restrict1 est affecté au maximum à 255 octets de res.filter.ConditionText de l'octet (3-1) * 255

Restrict1 est affecté au maximum à 255 octets de res.filter.ConditionText de l'octet (4-1) * 255

Référence associée:

«Table alerts.status», à la page 415
La table alerts.status contient des informations de statut sur les problèmes détectés par les sondes.

«Table master.class_membership», à la page 427
La table master.class_membership prend en charge le mappage des classes Tivoli Enterprise Console aux classes Tivoli Netcool/OMNIbus, et stocke les informations d'appartenance de classe. Cette table est utilisée avec la fonction SQL instance_of().

Expressions

Une expression est une combinaison syntaxique de valeurs et d'opérations associées pour calculer de nouvelles valeurs. Les expressions peuvent être simples ou complexes.

Expressions simples

Une expression simple est une valeur variable ou constante simple, un nom de colonne ou une référence variable. Il peut s'agir de l'une des entités suivantes :

- Une chaîne entre guillemets (`'Node XB1'`)
- Un numéro (`9`)
- Un nom de colonne (`Severity`)
- Une propriété du serveur ObjectServer (`ServerName`)
- Une variable d'environnement (`NCHOME`)
- Une variable contenant une valeur temporaire dans une procédure ou dans un déclencheur

Expressions complexes

Une expression complexe est créée à partir d'expressions simples combinées à l'aide d'opérateurs (`Severity - 1`) et de fonctions SQL (`get_prop_value(ServerName)`). Vous pouvez combiner des expressions simples ou complexes avec d'autres expressions simples ou complexes pour créer des expressions de plus en plus complexes, telles que `-(Severity + Tally)`.

Remarque : Les expressions complexes sont sujettes à des contraintes de type de données. Par exemple, l'expression `5 * 'Node XB1'` n'est pas valide car vous ne pouvez pas multiplier un entier et une chaîne.

Référence associée:

«Spécification des types de données des colonnes», à la page 171
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.

Conditions

Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

Vous pouvez utiliser les conditions pour rechercher, filtrer et tester les lignes dans :

- Les filtres de restriction
- La clause WHERE des commandes SELECT, UPDATE et DELETE
- La clause HAVING de la commande SELECT GROUP BY
- La clause WHEN des déclencheurs
- Instructions IF THEN ELSE, CASE WHEN et FOR EACH ROW dans les procédures et les déclencheurs

Les conditions peuvent contenir des opérateurs de comparaison (`Severity < 3`), des opérateurs logiques (`NOT(Is_Enabled)`) et des opérateurs de comparaison de liste (`Severity IN ANY(0,5)`).

Les conditions valides sont présentées dans la liste suivante :

```
TRUE | FALSE
( condition )
NOT condition
condition AND condition
condition OR condition
expression opérateur expression
expression opérateur ANY ( expression, ... )
expression opérateur ALL ( expression, ... )
expression[ NOT ] IN( sous-requête )
expression opérateur ANY ( sous-requête )
expression [ NOT ] IN ( expression, ... )
expression [ NOT ] LIKE modèle_expression_régulière
expression [ NOT ] LIKE ANY ( modèle_expression_régulière, ... )
expression [ NOT ] LIKE ALL ( modèle_expression_régulière, ... )
```

Remarque : Les opérateurs ANY et ALL ne sont pas pris en charge dans les sous-requêtes.

Vous pouvez combiner des conditions en conditions de plus en plus complexes.

Exemple

```
(Severity > 4) AND (Node = 'node%')
```

L'exemple suivant présente l'utilisation d'une condition dans une sous-requête :

```
select * from alerts.status where Serial in (select Serial from alerts.journal);
```

Concepts associés:

«Opérateurs de comparaison binaires», à la page 216
Utilisez ces opérateurs pour comparer l'égalité et l'inégalité des valeurs numériques et de chaîne.

«Opérateurs de comparaison de liste», à la page 217
Utilisez les opérateurs de comparaison de liste pour comparer une valeur et une liste de valeurs.

«Opérateurs logiques», à la page 218
Vous pouvez utiliser des opérateurs logiques sur des valeurs booléennes pour former les expressions qui aboutissent au résultat TRUE ou FALSE.

«Expressions», à la page 227
Une expression est une combinaison syntaxique de valeurs et d'opérations associées pour calculer de nouvelles valeurs. Les expressions peuvent être simples

ou complexes.

Référence associée:

«CREATE TRIGGER (déclencheur de base de données)», à la page 373
La commande `CREATE TRIGGER` permet de créer les déclencheurs de base de données qui s'activent lorsqu'une modification ou une tentative de modification dans une table du serveur ObjectServer survient (ou lorsqu'une modification ou une tentative de modification dans une vue affecte une table de base).

interface interactive SQL

Vous pouvez utiliser l'utilitaire de ligne de commande de l'interface interactive SQL fourni avec Tivoli Netcool/OMNIbus pour émettre des commandes SQL en direction d'un serveur ObjectServer.

Pour qu'il soit possible de se connecter à un serveur ObjectServer avec cette interface, le compte utilisateur doit appartenir à un groupe d'utilisateurs ayant un rôle ISQL prédéfini. Pour qu'il soit possible de modifier des données ObjectServer avec cette interface, le compte utilisateur doit appartenir à un groupe d'utilisateurs ayant un rôle ISQLWrite prédéfini.

Mode sécurisé

Lorsqu'un serveur ObjectServer s'exécute en mode sécurisé, il faut disposer de clients tels que l'interface interactive SQL pour s'y connecter à l'aide de noms d'utilisateur et de mots de passe valides. Les informations de connexion sont chiffrées lorsqu'elles sont transmises entre le client et le serveur ObjectServer. Les mots de passe sont déchiffrés par le serveur ObjectServer.

L'interface interactive SQL s'exécute en mode sécurisé par défaut. Si vous n'avez pas besoin d'une connexion sécurisée, vous pouvez utiliser l'option de ligne de commande `-nosecure` au démarrage de l'interface.

Concepts associés:

«Configuration des rôles», à la page 84
Les rôles sont des collectes de droits d'accès que vous pouvez affecter aux utilisateurs et aux groupes.

«Configuration des groupes», à la page 89
Utilisez les groupes pour classer les utilisateurs Tivoli Netcool/OMNIbus en unités disposant d'objectifs fonctionnels communs. Tous les membres d'un groupe disposent des droits d'accès affectés aux rôles du groupe.

Exécution des commandes SQL

Exécution des commandes SQL à l'aide de l'interface interactive SQL.

Procédure

Démarrage de l'interface.

- Utilisez la commande suivante pour démarrer l'interface :

  UNIX Linux `$NCHOME/omnibus/bin/`**`nco_sql`** `-server` *`nom_serveur_objets`* `-user` *`nom_utilisateur`* `-password` *`mot_de_passe`*

  Si vous ne spécifiez pas le nom du serveur ObjectServer, le nom par défaut NCOMS est utilisé. Si vous n'indiquez pas de nom d'utilisateur, la valeur par défaut correspond à l'utilisateur qui exécute la commande. Si vous n'utilisez pas l'option `-password`, vous êtes invité à entrer un mot de passe.

Windows `%NCHOME%\omnibus\bin\isql -S nom_serveur_objets -U nom_utilisateur -P mot_de_passe`

Vous devez préciser le nom du serveur ObjectServer et le nom d'utilisateur. Vous devez également inclure l'option `-P`, même si vous utilisez une chaîne de mot de passe vide (comme lors de la connexion au serveur ObjectServer par défaut NCOMS).

Remarque : Le mot de passe est visible dans la ligne de commande.

Au démarrage de l'interface, une invite numérotée s'affiche. L'invite se présente comme suit :

```
1>
```

Emission de commandes

- Pour émettre une commande à l'aide de l'interface :
  1. Entrez la commande à l'invite et appuyez sur Entrée.
  2. Entrez le mot clé `go` (en minuscules) et appuyez sur Entrée.

  L'interface possède les caractéristiques suivantes :
  - Les commandes sont limitées à 4094 caractères par ligne. Si vous avez besoin d'utiliser une commande plus longue, vous pouvez la fractionner sur plusieurs lignes ou utiliser l'éditeur par défaut pour lancer la commande.
  - Vous pouvez entrer plusieurs commandes séparées par un point-virgule (;) et toutes les exécuter via une commande `go` unique.
  - Vous pouvez fractionner une commande unique sur plusieurs lignes.
  - Pour annuler une commande, entrez `reset` au début d'une nouvelle ligne ou appuyez sur Ctrl+C n'importe où sur une ligne.

    Toutes les commandes qui n'ont pas été exécutées sont supprimées.
  - Windows Lors de la spécification des chemins des répertoires (comme cela est requis pour certaines commandes SQL), utilisez une barre oblique inversée (\) supplémentaire pour chaque séparateur de répertoire. Exemple :

    ```
    create file nom_fichier 'c:\\temp\\nom_fichier.txt';
    ```

    L'interface accepte aussi les chemins de style UNIX sur les systèmes Windows. Exemple :

    ```
    create file nom_fichier 'c:/temp/nom_fichier';
    ```
  - Pour exécuter une commande de système d'exploitation, entrez `!!`, suivi de la commande, au début d'une nouvelle ligne .

    Exemple : `!!ls`
  - UNIX Linux Pour exécuter l'éditeur par défaut (tel que défini par la variable d'environnement `EDITOR`), entrez `vi` au début d'une nouvelle ligne.
  - UNIX Linux **`nco_sql`** n'autorise pas les espaces avant le mot clé `go`.

Fermeture de l'interface.

- Pour fermer l'interface, entrez `quit` ou `exit` au début d'une nouvelle ligne.

  UNIX Linux Vous pouvez également appuyer sur Ctrl+D pour fermer l'interface.

Référence associée:

«Options de ligne de commande de l'interface interactive SQL», à la page 232
Vous pouvez spécifier un certain nombre d'options de ligne de commande lorsque vous lancez l'interface interactive SQL.

Exécution des commandes SQL à partir de fichiers

L'interface interactive SQL accepte les fichiers SQL qui peuvent contenir plusieurs commandes SQL. Vous pouvez également utiliser l'interface pour rediriger les fichiers texte. Ces deux fonctions sont utiles lorsque vous exécutez des tâches d'administration répétitives.

Avant de commencer

Les fichiers SQL utilisés avec l'interface interactive SQL doivent contenir uniquement des commandes SQL et doivent se terminer par la commande `go`.

`nco_sql` n'autorise pas les espaces avant le mot clé `go`. Par exemple, si vous utilisez un espace pour mettre les commandes en retrait dans un fichier, les commandes échouent.

Procédure

Exécution de commandes à partir d'un fichier.

- Pour lire un fichier au démarrage de l'interface, utilisez la commande suivante :

  UNIX Linux `$NCHOME/omnibus/bin/`**`nco_sql`** `-server` *`nom_serveur_objets`* `-username` *`nom_utilisateur`* `-password` *`mot_de_passe`* `<` *`nom_fichier`*`.txt`

  Windows `%NCHOME%\omnibus\bin\`**`isql`** `-S` *`nom_serveur_objets`* `-U` *`nom_utilisateur`* `-P` *`mot_de_passe`* `-i` *`nom_fichier`*`.txt`

Redirection de la sortie vers un fichier.

- Pour rediriger la sortie vers un fichier au démarrage de l'interface, utilisez la commande suivante :

  UNIX Linux `$NCHOME/omnibus/bin/`**`nco_sql`** `-server` *`nom_serveur_objets`* `-username` *`nom_utilisateur`* `-password` *`mot_de_passe`* `<` *`fichier_entrée`*`.txt >` *`fichier_sortie`*`.txt`

  Windows `%NCHOME%\omnibus\bin\`**`isql`** `-S` *`nom_serveur_objets`* `-U` *`nom_utilisateur`* `-P` *`mot_de_passe`* `-i` *`fichier_entrée`*`.txt -o` *`fichier_sortie`*`.txt`

  Remarque : N'utilisez pas les caractères standard de redirection < et > sur les systèmes d'exploitation Windows.

Lecture d'un fichier lorsque l'interface est en cours d'exécution.

- Lorsque l'interface est en cours d'exécution, entrez la commande suivante au début d'une nouvelle ligne :

  `:r` *`nom_fichier`*

  N'incluez pas la commande `go` dans le fichier. Entrez plutôt la commande `go` au début d'une nouvelle ligne.

Résultats

Si les commandes SQL du fichier sont valides, les modifications sont implémentées sur le serveur ObjectServer.

Chiffrement des mots de passe dans les scripts

UNIX Linux

Vous pouvez appeler l'interface interactive SQL (**nco_sql**) à partir des scripts. Dans ce cas, vous devez avoir recours à l'utilitaire **nco_sql_crypt** pour chiffrer votre mot de passe ObjectServer. Les mots de passe sont déchiffrés par le serveur ObjectServer.

Pourquoi et quand exécuter cette tâche

Lorsque vous exécutez Tivoli Netcool/OMNIbus en mode FIPS 140–2, utilisez des mots de passe en texte normal ou utilisez l'utilitaire **nco_aes_crypt** pour chiffrer les mots de passe dans les scripts.

Procédure

1. Utilisez la commande suivante pour chiffrer le mot de passe :

   `$NCHOME/omnibus/bin/nco_sql_crypt` *`mot_de_passe`*

   L'utilitaire affiche en sortie une version chiffrée du mot de passe.
2. Copiez le mot de passe chiffré dans votre script.

Options de ligne de commande de l'interface interactive SQL

Vous pouvez spécifier un certain nombre d'options de ligne de commande lorsque vous lancez l'interface interactive SQL.

Le tableau suivant répertorie les options de ligne de commande disponibles.

Tableau 76. Options de ligne de commande de l'interface interactive SQL

Option de ligne de commande	Description
`-help`	Utilisez cette option pour afficher des informations d'aide sur les options de ligne de commande.
UNIX Linux `-input` *chaîne* Windows `-i` *chaîne*	Cette option permet de spécifier l'emplacement d'un fichier d'entrée SQL. Les commandes SQL du fichier sont exécutées sur le serveur ObjectServer auquel se connecte l'interface interactive SQL. **Remarque :** Lorsque vous vous connectez à l'aide de l'option `-nosecure`, vous devez utiliser l'option `-input` pour lire le fichier au lieu de la syntaxe d'entrée de fichier standard `<` *`nom_fichier`*.

Tableau 76. Options de ligne de commande de l'interface interactive SQL (suite)

Option de ligne de commande	Description
UNIX Linux `-networktimeout` *entier* Windows `-l` *entier* Windows `-t` *entier*	UNIX Linux Cette option permet de spécifier un délai d'attente (en secondes) pour les tentatives de connexion et pour la connexion au serveur ObjectServer lorsqu'il se produit une panne réseau. A l'issue du délai d'attente spécifié, l'interface tente de se reconnecter au serveur ObjectServer. Si la connexion n'aboutit pas après un second délai d'attente, l'interface tente de se connecter à un serveur ObjectServer de sauvegarde (le cas échéant). **Remarque :** La commande **nco_sql** exécute l'utilitaire **nco_get_login_token** pour obtenir un jeton de connexion et démarre l'interface avec le jeton. Le délai d'attente spécifié est transmis à **nco_get_login_token** et à l'interface au moment de leur démarrage. Les jetons de connexion ne sont valides que pendant 15 secondes. Le délai d'attente maximum autorisé en mode sécurisé par défaut est donc de 14 secondes. Vous pouvez indiquer un délai d'attente plus long si vous utilisez l'option `-nosecure`. Windows L'option `-l` permet de définir le délai d'attente de connexion. La valeur par défaut est `60` secondes. L'option `-t` permet de définir un délai d'attente pour les commandes. Si vous ne définissez pas de délai d'attente, les commandes peuvent s'exécuter sans fin.
`-nosecure`	Cette option permet de spécifier que les informations de connexion ne sont pas chiffrées lorsqu'elles sont transmises entre l'interface et le serveur ObjectServer.
UNIX Linux `-password` *chaîne* Windows `-P` *chaîne*	Cette option permet de spécifier le mot de passe utilisateur. Windows L'option `-P` est obligatoire, même si vous utilisez une chaîne de mot de passe vide (comme lorsque vous vous connectez au serveur par défaut ObjectServer NCOMS).
`-secure`	Cette option permet de spécifier que les informations de connexion sont chiffrées lorsqu'elles sont transmises entre l'interface et le serveur ObjectServer. La connexion sécurisée est activée par défaut.
UNIX Linux `-server` *chaîne* Windows `-S` *chaîne*	Cette option permet d'indiquer le nom du serveur ObjectServer auquel se connecter. La valeur par défaut est NCOMS.

Tableau 76. Options de ligne de commande de l'interface interactive SQL (suite)

Option de ligne de commande	Description
UNIX Linux `-user` *chaîne* Windows `-U` *chaîne*	Cette option permet de spécifier un nom d'utilisateur Tivoli Netcool/OMNIbus. La valeur par défaut est l'utilisateur qui exécute la commande. Si le nom d'utilisateur contient des espaces, placez-le entre guillemets (" "). Exemple : `$NCHOME/omnibus/bin/nco_sql -server NCOMS -user "NCOMS Admin"`

Tâches associées:

«Exécution des commandes SQL», à la page 229
Exécution des commandes SQL à l'aide de l'interface interactive SQL.

Chapitre 6. Configuration de la notification des événements accélérés

Vous pouvez configurer Tivoli Netcool/OMNIbus pour accélérer la notification des événements pouvant présenter un risque pour le système. Le système Accelerated Event Notification (AEN) permet d'accélérer les événements de priorité élevée afin de garantir une exécution des systèmes sans interruption.

Lorsque vous configurez la notification des événements accélérés, procédez comme suit :

- Déterminez les conditions pour lesquelles vous souhaitez accélérer les événements. Déterminez si vous souhaitez utiliser le fichier de règles de sonde pour indiquer les événements à accélérer ou pour configurer les conditions d'un déclencheur de post-insertion, de post-mise à jour ou de post-réinsertion dans le serveur ObjectServer.

  Si les conditions pour lesquelles les événements doivent être accélérés sont configurées dans le fichier de règles et sont complexes, vous aurez peut-être besoin d'insérer une colonne pour recevoir un indicateur signalant un événement accéléré dans la table alerts.status.

  Généralement, vous pouvez exécuter un déclencheur de post-insertion dans les événements insérés pour déterminer s'ils remplissent les conditions d'accélération, exécuter un déclencheur de post-mise à jour sur les événements mis à jour qui ont été enrichis avec les informations d'autres sources ou exécuter un déclencheur de post-réinsertion pour intensifier les événements répétitifs.
- Déterminez si vous avez besoin d'une passerelle dédiée pour transmettre les événements accélérés. Pensez à utiliser une passerelle dédiée si vous souhaitez éviter tout délai imprévu causé par de grands volumes d'événements, qui peuvent être non critiques, ou si vous souhaitez envoyer des événements accélérés dans un serveur ObjectServer différent utilisé pour les mises à jour IDUC normales. Si vous utilisez une passerelle dédiée, vous devez configurer cette passerelle pour répliquer uniquement les insertions et les mises à jour accélérées.

Vous devez également utiliser Netcool/OMNIbus Administrator pour configurer une notification d'événement accéléré, comme suit :

- Configurez une colonne d'événements dédiée pour indiquer un événement à accélérer, si nécessaire.
- Configurez les canaux pour définir le type de données d'événement à inclure dans les notifications d'événement accéléré et les destinataires de ces données d'événement.
- Configurez les déclencheurs de post-insertion, post-mise à jour ou post-réinsertion devant agir sur les événements accélérés insérés ou mis à jour dans la table alerts.status.

Pour plus d'informations sur la surveillance et la gestion des événements accélérés, voir *IBM Tivoli Netcool/OMNIbus User's Guide*.

© Copyright IBM Corp. 1994, 2014

Configuration d'une sonde pour indiquer des événements pour l'accélération

Si les conditions de notification d'événement accéléré sont complexes, déterminez si vous devez les configurer dans le fichier de règles de la sonde.

Pourquoi et quand exécuter cette tâche

Vous pouvez également ajouter une colonne d'événement dédié à la table alerts.status pour indiquer les événements d'accélération et utiliser cette zone dans le fichier de règles de la sonde.

L'exemple de fichier de règles de sonde ci-dessous explique comment un flot d'événements peut être analysé afin de déterminer les événements dont la priorité est élevée. Au début du fichier de règles de sonde, les éléments (indiqués par le symbole $) sont attribués aux zones ObjectServer (indiquées par le symbole @). L'instruction conditionnelle utilise l'élément $Summary pour définir les valeurs AlertKey et FastTrack dans la table alerts.status.

L'instruction se traduit de la manière suivante : si la valeur Summary commence par 'Port failure', insérez le numéro de port et la valeur 1 respectivement dans les zones AlertKey et FastTrack de la table alerts.status. Sinon, si la valeur Summary commence par la chaîne 'Diskspace', insérez le pourcentage concaténé et la chaîne complète % dans la zone AlertKey de la table alerts.status.

```
@Manager = "Simnet Probe"
@Class  = 3300
@Node   = $Node
@Agent   = $Agent
@AlertGroup = $Group
@Summary  = $Summary
@Severity  = $Severity
@Identifier  = $Node + $Agent + $Severity + $Group

if (nmatch($Summary, "Port failure"))
{
 @AlertKey = $PortNumber
 @FastTrack = 1
}
else if (nmatch($Summary, "Diskspace"))
{
@AlertKey = $PercentFull + "% full"
}
```

Configuration d'une passerelle de notification d'événement accéléré

Vous pouvez choisir d'utiliser une passerelle dédiée pour transférer les événements accélérés pour des raisons de performances principalement ou pour envoyer les événements à un serveur ObjectServer particulier. Si vous utilisez une passerelle dédiée, vous devez mettre à jour son fichier de définition de réplication de table de manière à ne répliquer que les insertions et mises à jour accélérées.

Pourquoi et quand exécuter cette tâche

Pour configurer une passerelle ObjectServer unidirectionnelle de notification d'événement accéléré, procédez comme suit :

Procédure

1. Ouvrez le fichier de définition de réplication de table de passerelle suivant :
   `$OMNIHOME/gates/objserv_uni/objserv_uni.reader.tblrep.def`
2. Recherchez les lignes ci-après :
   ```
   REPLICATE ALL FROM TABLE 'alerts.status' USING MAP 'StatusMap';
   REPLICATE ALL FROM TABLE 'alerts.journal' USING MAP 'JournalMap';
   REPLICATE ALL FROM TABLE 'alerts.details' USING MAP 'DetailsMap';
   ```
3. Remplacez ces trois lignes par la ligne suivante :
   ```
   REPLICATE FT_INSERT,FT_UPDATE FROM TABLE 'alerts.status' USING MAP 'StatusMap';
   ```
4. Sauvegardez le fichier.

Que faire ensuite

Dans le cas d'une passerelle bidirectionnelle, vous devez modifier les fichiers correspondants de la même manière. Le fichier de définition de réplication de table de passerelle d'une passerelle bidirectionnelle est `$OMNIHOME/gates/objserv_bi/objserv_bi.reader.tblrep.def`.

Configuration de la table alerts.status pour recevoir l'indicateur AEN

Si vous avez configuré votre fichier de règles de sonde avec un indicateur de notification d'événement accéléré, vous pouvez ajouter une colonne permettant de prendre en charge l'accélération des événements dans la table alerts.status.

Pourquoi et quand exécuter cette tâche

Par exemple, dans Netcool/OMNIbus Administrator, ajoutez une colonne dont le Nom de colonne (**Column Name**) est `FastTrack` et le type de données (**Data Type**) Entier(`Integer`).

Tâches associées:

«Ajout et édition de colonnes d'une table», à la page 151
Vous pouvez ajouter de nouvelles colonnes aux tables ObjectServer ou modifier des colonnes existantes. Une table peut contenir au maximum 512 colonnes, à l'exclusion des colonnes gérées par le système. La taille de ligne maximale d'une table, qui est la somme des longueurs de colonne de la ligne, est de 64 Ko. Vous ne pouvez pas ajouter de colonnes de clé primaire aux tables existantes.

Référence associée:

«ALTER TABLE», à la page 357
La commande `ALTER TABLE` permet de modifier les caractéristiques d'une table existante et de ses colonnes. Vous pouvez ajouter, supprimer et modifier des colonnes ; vous pouvez également définir manuellement la valeur des zones de type `INCR`.

Configuration des canaux afin de diffuser les données d'événement

Lors de la configuration des notifications d'événement accéléré, vous devez utiliser les canaux pour définir le type de données d'événement à diffuser dans les notifications d'événement accéléré et les destinataires de ces données. Vous pouvez configurer plusieurs canaux possédant des données d'événement et des destinataires variés.

L'administration des canaux est uniquement autorisée aux utilisateurs possédant le rôle ChannelAdmin.

Les données d'événement des canaux proviennent des colonnes des tables du serveur ObjectServer.

Conseil : Les colonnes que vous choisissez pour un canal doivent contenir suffisamment de données récapitulatives pour aider les opérateurs à interpréter rapidement les problèmes critiques lorsque ces problèmes sont transmis à l'écran sous la forme de notificateurs en incrustation. Les opérateurs peuvent ensuite cliquer dans la liste d'événements du bureau ou dans la liste d'événements actifs de l'interface graphique Web pour obtenir des informations complètes sur l'événement et gérer l'événement.

Notez que la fonction par clic d'un notificateur contextuel vers la liste d'événements ou la liste d'événements actifs n'est disponible que pour les événements de la table alerts.status.

Création et édition de canaux

Vous devez créer des canaux sur un serveur ObjectServer à partir duquel les événements accélérés vont être transférés.

Avant de commencer

Vous devez être connecté au serveur ObjectServer sur lequel vous souhaitez créer un canal.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un canal, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Channels** (Canaux). Le panneau Channels (Canaux) s'ouvre.
3. Pour ajouter un canal, cliquez sur **Add Channel** (Ajouter un canal) dans la barre d'outils. La fenêtre Channel Details (Détails du canal) s'ouvre.
4. Pour éditer un canal, sélectionnez-le, puis cliquez sur **Edit Channel** (Editer le canal) dans la barre d'outils. La fenêtre Channel Details (Détails du canal) s'ouvre.
5. Définissez le canal comme suit :

   Nom Entrez un nom unique pour le canal. Si vous éditez un canal, vous ne pouvez pas modifier son nom.

   Description
   Entrez un texte pertinent résumant la fonction du canal.

6. Dans l'onglet **Columns** (Colonnes), indiquez les colonnes que vous souhaitez inclure dans la définition du canal. Renseignez l'onglet comme suit :

   Add new Channel Columns (Ajouter de nouvelles colonnes de canal)
   Cliquez sur ce bouton pour ajouter des colonnes au canal. La fenêtre Channel Column Details (Détails des colonnes de canal) s'ouvre alors. Complétez cette fenêtre comme suit et enregistrez vos modifications :

   Table Dans la liste de gauche, sélectionnez une base de données ObjectServer. Dans la liste de droite, sélectionnez une table dans cette base de données.

   Si vous éditez des colonnes de canal, vous ne pouvez pas modifier le nom de la base de données ou de la table.

   Restriction : Actuellement, la prise en charge n'est disponible que pour les événements de la table alerts.status.

   Columns: Available (Colonnes : Disponibles)
   Cette liste est remplie à l'aide du nom des colonnes définies dans la table de base de données sélectionnée et que vous pouvez affecter au canal. Pour affecter une ou plusieurs de ces colonnes, utilisez les touches de déplacement pour insérer les colonnes dans la liste **Sélectionnés**.

   Pour insérer toutes les colonnes dans la liste **Sélectionnés**, cliquez sur **>>**. Pour insérer une seule ou plusieurs colonnes dans la liste **Sélectionnés**, sélectionnez chacune des colonnes et cliquez sur **>**. Vous pouvez utiliser la touche majuscule pour effectuer des sélections consécutives, ou la touche CTRL pour effectuer des sélections non consécutives.

   Vous pouvez également cliquer deux fois sur une colonne pour la faire passer de la liste **Disponibles** à la liste **Sélectionnés**. Les colonnes sont alors ajoutées à la fin de la liste **Sélectionnés**.

   Columns: Selected (Colonnes : Sélectionnés)
   Cette liste contient les colonnes incluses dans la définition de canal. Pour supprimer des colonnes de la définition de canal, utilisez les touches de déplacement pour insérer les colonnes dans la liste **Disponibles**.

   Pour insérer toutes les colonnes dans la liste **Disponibles**, cliquez sur **<<**. Pour insérer une seule ou plusieurs colonnes dans la liste **Disponibles**, sélectionnez chacune des colonnes et cliquez sur **<**. Vous pouvez utiliser la touche majuscule pour effectuer des sélections consécutives, ou la touche CTRL pour effectuer des sélections non consécutives.

   Vous pouvez également cliquer deux fois sur une colonne pour la faire passer de la liste **Sélectionnés** à la liste **Disponibles**.

   Utilisez les flèches à droite de la liste **Sélectionnés** pour indiquer la position des données de colonnes, au sein du notificateur contextuel dans le client de notification d'événement accéléré. La liste **Sélectionnés** affiche les marqueurs par défaut suivants pour indiquer la position des données de colonnes :

   - H : En-tête
   - F : Première ligne

- S : Deuxième ligne
- M : Message principal
- N : Remarque

Pour modifier la position d'une colonne, sélectionnez-la et cliquez sur la flèche appropriée. Utilisez les flèches comme suit :

Tableau 77. Flèches

Bouton	Description
	Déplace la colonne sélectionnée tout en haut de la liste **Sélectionnés**.
	Déplace la colonne sélectionnée vers le niveau supérieur dans la liste **Sélectionnés**.
	Déplace la colonne sélectionnée vers le niveau inférieur dans la liste **Sélectionnés**.
	Déplace la colonne sélectionnée tout en bas de la liste **Sélectionnés**.

Lorsque vous revenez à l'onglet **Colonnes** de la fenêtre Channel Details (Détails du canal), vos sélections de colonnes s'affichent au sein d'une même ligne. (Vous ne pouvez pas en ajouter plusieurs.)

Edit selected Channel Columns (Editer les colonnes de canal sélectionnées)
Cliquez sur ce bouton pour éditer la définition des colonnes du canal. Sélectionnez la ligne dans l'onglet **Colonnes** et cliquez sur le bouton. La fenêtre Channel Column Details (Détails des colonnes de canal) s'ouvre alors. Modifiez vos sélections de colonnes de cette fenêtre et enregistrez vos modifications avant de revenir à l'onglet **Colonnes**.

Delete selected Channel Columns (Supprimer les colonnes de canal sélectionnées)
Utilisez ce bouton pour supprimer une définition de colonne du canal. Sélectionnez la ligne dans la table **Colonnes** et cliquez sur **Supprimer**. A l'invite, confirmez la suppression. Vos modifications sont ainsi prises en compte dans l'onglet **Colonnes**.

7. Dans l'onglet **Recipients** (Destinataires), indiquez l'utilisateur ou le groupe d'utilisateurs auquel les données du canal vont être envoyées. Renseignez l'onglet comme suit :

   Add new Channel Recipient (Ajouter un nouveau destinataire de canal)
   Cliquez sur ce bouton pour indiquer des destinataires pour les informations de canal. La fenêtre Channel Recipient Details (Détails de destinataires de canal) s'ouvre alors. Complétez cette fenêtre comme suit et enregistrez vos modifications :

   isGroup (estGroupe)
   Cochez cette case pour indiquer que les destinataires de canal forment un groupe d'utilisateurs. Décochez cette case si le destinataire de canal est un utilisateur unique.

   Nom Cette liste fonctionne conjointement avec la case à cocher **isGroup** (estGroupe) et est remplie à l'aide d'une liste de tous les utilisateurs ou d'une liste de tous les groupes du serveur ObjectServer. Sélectionnez le nom de l'utilisateur ou du groupe devant recevoir les données de canal.

Hostname (Nom d'hôte)
Entrez le nom de l'hôte connecté. Vous pouvez utiliser des expressions régulières pour filtrer les connexions correspondant à cette valeur. Laissez la zone vide pour qu'elle affiche toute correspondance sur un nom d'hôte. Par exemple, entrez l'expression régulière `*test*` pour faire correspondre un hôte et la chaîne `test` au sein de son nom.

Application Name (Nom d'application)
Entrez le nom d'une application connectée au serveur ObjectServer. Vous pouvez utiliser des expressions régulières pour filtrer les connexions correspondant à votre entrée. Par exemple, `*event*` indique une correspondance entre un hôte et la chaîne `event` au sein de son nom. Laissez cette zone vide pour qu'elle affiche toute correspondance sur un nom d'application.

Description d'application
Entrez une description d'application. Vous pouvez utiliser des expressions régulières pour filtrer les connexions correspondant à votre entrée. Par exemple, `*real time*` établit une correspondance entre un hôte et la chaîne `real time` au sein de sa description d'application. Laissez cette zone vide pour qu'elle affiche toute correspondance sur une description d'application.

Lorsque vous revenez à l'onglet **Recipients** (Destinataires) de la fenêtre Channel Details (Détails du canal), les détails des destinataires s'affichent au sein d'une même ligne. (Vous pouvez ajouter d'autres lignes de destinataires.)

Edit the selected Channel Recipient (Editer le destinataire de canal sélectionné)
Cliquez sur ce bouton pour éditer les détails du destinataire pour une ligne sélectionnée dans l'onglet **Recipients** (Destinataires). La fenêtre Channel Recipient Details (Détails de destinataires de canal) s'ouvre alors. Modifiez les détails de destinataires et enregistrez vos modifications avant de revenir à l'onglet **Recipients** (Destinataires).

Delete the selected Channel Recipient (Supprimer le destinataire de canal sélectionné)
Utilisez ce bouton pour supprimer des destinataires de la définition de canal. Sélectionnez la ligne de destinataires à supprimer et cliquez sur ce bouton. A l'invite, confirmez la suppression. Vos modifications sont ainsi prises en compte dans l'onglet **Recipients** (Destinataires).

8. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails du canal et fermer la fenêtre. Les nouveaux canaux sont ajoutés dans le panneau Channels (Canaux).

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Copie et collage de canaux

Vous pouvez utiliser un canal comme modèle pour un autre en copiant et collant sa définition. Cela est très utile si vous souhaitez créer des canaux avec de légères variations de leurs définitions.

Pourquoi et quand exécuter cette tâche

Restriction : Vous ne pouvez pas procéder à un copier-coller entre des serveurs ObjectServer.

Pour copier et coller un canal, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Channels** (Canaux). Le panneau Channels (Canaux) s'ouvre.
3. Pour copier un canal, sélectionnez-le dans le panneau Channels (Canaux), puis cliquez sur **Copy** (Copier) dans la barre d'outils.
4. Pour coller le canal, cliquez sur **Paste** (Coller) dans la barre d'outils. La fenêtre New Channel (Nouveau canal) s'ouvre.
5. Entrez un nom unique pour le canal.
6. Confirmez ou annulez vos actions comme suit :
   - Cliquez sur **OK** pour créer le canal avec une définition identique à celle du canal sélectionné. La fenêtre Channel Details (Détails du canal) s'ouvre. Vous pouvez procéder aux modifications requises de ce nouveau canal.
   - Cliquez sur **Cancel** (Annuler) pour annuler l'action de copier-coller.

Suppression d'un canal

Pourquoi et quand exécuter cette tâche

Pour supprimer un canal, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur l'icône **Channels** (Canaux). La fenêtre Channels (Canaux) s'ouvre.
3. Sélectionnez le canal à supprimer et cliquez sur **Delete** (Supprimer) dans la barre d'outils. Le canal est supprimé.

Envoi de messages aux destinataires de canal

Vous pouvez envoyer des messages aux destinataires du canal qui exécutent le client de notification d'événement accéléré.

Pourquoi et quand exécuter cette tâche

Vous pouvez envoyer des messages aux destinataires qui écoutent sur un ou plusieurs canaux.

Pour envoyer des messages, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Channels** (Canaux). Le panneau Channels (Canaux) s'ouvre.
3. Dans le panneau Channels (Canaux), sélectionnez un ou plusieurs canaux associés au message à envoyer. Vous pouvez utiliser la touche majuscule pour les sélections consécutives ou la touche Ctrl pour les sélections non consécutives.
4. Dans la barre d'outils, cliquez sur **Send Message** (Envoyer le message). La fenêtre Send Message (Envoyer le message) s'ouvre.
5. Entrez le message dans la zone **Message Text** (Texte du message). Cette zone défile horizontalement pour permettre d'entrer le texte.
6. Confirmez ou annulez vos actions comme suit :
   - Cliquez sur **OK** pour initier l'action d'envoi, puis confirmez l'envoi du message en cliquant sur **Yes** (Oui).

     Remarque : Si la zone **Message Text** (Texte du message) est vide, aucun message n'est envoyé.
   - Cliquez sur **Cancel** (Annuler) pour annuler l'action d'envoi.

Résultats

Le message s'affiche dans une boîte de message sur tous les écrans du client de notification d'événement accéléré correspondants. Le texte du message est limité à 120 caractères. Si vous envoyez le message à plusieurs canaux, un client écoutant un ou plusieurs de ces canaux ne reçoit le message qu'une seule fois.

Déconnexion des clients de notification d'événement accéléré

Si vous souhaitez procéder à des opérations de maintenance mineures sur le serveur ObjectServer (une resynchronisation, par exemple), vous pouvez déconnecter (ou fermer la session) à distance les clients de notification d'événement accéléré en cours d'exécution. Lors de la déconnexion, vous pouvez entrer un bref message contenant les informations pertinentes à l'intention des utilisateurs.

Pourquoi et quand exécuter cette tâche

Pour déconnecter les clients de notification d'événement accéléré à l'écoute sur un ou plusieurs canaux, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Channels** (Canaux). Le panneau Channels (Canaux) s'ouvre.
3. Dans le panneau Channels (Canaux), sélectionnez un ou plusieurs canaux associés à l'action de déconnexion. Vous pouvez utiliser la touche majuscule pour les sélections consécutives ou la touche Ctrl pour les sélections non consécutives.
4. Dans la barre d'outils, cliquez sur **Disconnect Clients** (Déconnecter les clients). La fenêtre Send Disconnect Command (Envoyer la commande de déconnexion) s'ouvre.

5. Entrez un message dans la zone **Reason for Disconnect** (Raison de la déconnexion). Cette zone défile horizontalement pour permettre d'entrer le texte.
6. Confirmez ou annulez vos actions comme suit :
   - Cliquez sur **OK** pour initier l'action de déconnexion, puis confirmez la déconnexion en cliquant sur **Yes** (Oui).

     Remarque : Si la zone **Reason for Disconnect** (Raison de la déconnexion) est vierge, les clients se déconnectent, mais aucun explication n'est donnée dans la boîte de message.
   - Cliquez sur **Cancel** (Annuler) pour annuler l'action de déconnexion.

Résultats

Après avoir confirmé l'action de déconnexion, tous les messages que vous avez entrés s'affichent dans une boîte de message sur tous les écrans du client de notification d'événement accéléré correspondants. Le texte du message est limité à 120 caractères. Une fermeture automatique de session se produit. L'indicateur d'état des clients de notification d'événement accéléré reflète l'état de fermeture de session, même si les clients continuent de s'exécuter en arrière-plan.

Arrêt des clients de notification d'événement accéléré

Si vous souhaitez arrêter le serveur ObjectServer, vous pouvez arrêter à distance les clients de notification d'événement accéléré en cours d'exécution. Lors de la procédure d'arrêt, vous pouvez entrer un bref message contenant les informations pertinentes à l'intention des utilisateurs.

Pourquoi et quand exécuter cette tâche

Pour arrêter les clients de notification d'événement accéléré à l'écoute sur un ou plusieurs canaux, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **System** (Système).
2. Cliquez sur **Channels** (Canaux). Le panneau Channels (Canaux) s'ouvre.
3. Dans le panneau Channels (Canaux), sélectionnez un ou plusieurs canaux associés à l'action d'arrêt. Vous pouvez utiliser la touche majuscule pour les sélections consécutives ou la touche Ctrl pour les sélections non consécutives.
4. Dans la barre d'outils, cliquez sur **Shutdown Clients** (Arrêter les clients). La fenêtre Send Shutdown Command (Envoyer la commande d'arrêt) s'ouvre.
5. Entrez un message dans la zone **Reason for Shutdown** (Raison de l'arrêt). Cette zone défile horizontalement pour permettre d'entrer le texte.
6. Confirmez ou annulez vos actions comme suit :
   - Cliquez sur **OK** pour initier l'action d'arrêt, puis confirmez l'arrêt en cliquant sur **Yes** (Oui).

     Remarque : Si la zone **Reason for Shutdown** (Raison de l'arrêt) est vierge, les clients procèdent à l'arrêt, mais aucun explication n'est donnée dans la boîte de message.
   - Cliquez sur **Cancel** (Annuler) pour annuler l'arrêt.

Résultats

Après avoir confirmé l'action d'arrêt, tous les messages que vous avez entrés s'affichent dans une boîte de message sur tous les écrans du client de notification d'événement accéléré correspondants. Le texte du message est limité à 120 caractères. Au bout de cinq secondes, la session est automatiquement fermée.

Configuration des déclencheurs pour la prise en charge de la notification d'événement accéléré

Pour prendre en charge une notification d'événement accéléré, créez des déclencheurs postérieurs à l'insertion, à la mise à jour et à la réinsertion qui sont liés à la table alerts.status. Dans les déclencheurs, configurez des conditions permettant de définir ou d'identifier des événements accélérés lorsqu'ils sont insérés ou mis à jour dans la table alerts.status et de transférer ces événements aux clients de notification d'événement accéléré appropriés.

Pourquoi et quand exécuter cette tâche

Deux commandes SQL peuvent être utilisées avec vos déclencheurs : une commande de suivi rapide des événements (ou événement accéléré) (IDUC EVTFT) et une commande d'envoi de messages (IDUC SNDMSG).

Conseil : Vous pouvez trouver utile de regrouper des déclencheurs prenant en charge la notification d'événement accéléré au sein de leur propre groupe.

Concepts associés:

«Déclencheurs», à la page 176
Vous pouvez utiliser les déclencheurs pour détecter les modifications du serveur ObjectServer et exécuter des réponses automatiques à ces modifications. Vous permettez ainsi au serveur ObjectServer de traiter des alertes sans qu'un opérateur n'aie besoin d'intervenir. Vous pouvez également utiliser l'automatisation pour gérer le dédoublonnage, qui permet de réduire la quantité de données traitées dans le serveur ObjectServer en éliminant les événements en double.

«Configuration des déclencheurs», à la page 116
Vous pouvez créer et modifier des déclencheurs à partir des fenêtres Netcool/OMNIbus Administrator par défaut ou en utilisant un script SQL. Pour créer le script SQL, vous utilisez un éditeur de texte externe. Il existe trois types de déclencheur : déclencheurs de base de données, déclencheurs de signal et déclencheurs temporels.

Référence associée:

«IDUC EVTFT», à la page 395
La commande `IDUC EVTFT` permet d'activer les notificateurs contextuels des événements accélérés à envoyer aux clients, puis d'activer la fonction de la liste d'événements de bureau ou de la liste d'événements actifs de l'interface graphique Web.

«IDUC SNDMSG», à la page 397
La commande `IDUC SNDMSG` permet d'envoyer des messages d'informations à un client Notification d'événement accéléré.

Chapitre 7. Utilisation du contrôle de processus pour gérer des processus et des procédures externes

Le système de contrôle Tivoli Netcool/OMNIbus effectue deux tâches principales. Il gère les processus locaux et distants, puis exécute les processus externes spécifiés dans les automatisations.

Vous pouvez utiliser le contrôle de processus pour simplifier la gestion des composants Tivoli Netcool/OMNIbus, tels que les serveurs ObjectServer, les sondes et les passerelles. Vous pouvez installer les agents de processus sur chaque hôte et les configurer afin de gérer les processus. Les agents de processus configurés coopèrent automatiquement et comprennent leur propre configuration. Ils démarrent les processus et maintiennent l'exécution de ces processus. Vous pouvez définir des processus dépendant d'autres processus et des processus possédant des dépendances de seuil temporisées. Si un hôte géré redémarre, l'agent de processus peut être configuré pour redémarrer automatiquement les composants locaux.

Le système de contrôle de processus comprend un ensemble d'utilitaires de ligne de commande possédant une interface de gestion. Vous pouvez configurer et gérer un contrôle de processus à partir de la ligne de commande ou à l'aide de Netcool/OMNIbus Administrator.

Connexion des agents de processus

Pour configurer un système réseau de contrôle de processus, configurez le contrôle de processus sur plusieurs hôtes Tivoli Netcool/OMNIbus.

Les agents de processus peuvent ensuite communiquer entre eux et exécuter des programmes à la demande. Les agents de processus exécutés sous Windows peuvent communiquer avec les agents de processus exécutés sous UNIX, et inversement.

Exécution des agents de processus dans une configuration de routage

Lorsque plusieurs agents de processus sont connectés par des instructions de routage, chaque agent de processus du réseau de contrôle de processus peut être informé des processus des autres agents de processus. Le fichier de configuration d'un agent de processus permet de définir des processus, des services et des hôtes dans le réseau du contrôle de processus. Il permet également de définir les instructions de routage.

Le système de contrôle de processus prend intégralement en charge la gestion à distance des agents de processus depuis une console unique. Vous pouvez ajouter, modifier, supprimer, démarrer et arrêter les services et processus à distance. Vous pouvez également afficher le statut des processus locaux et distants. Si vous utilisez Netcool/OMNIbus Administrator pour gérer le contrôle de processus, vous pouvez, si vous le souhaitez, enregistrer vos modifications dans les fichiers de configuration des agents de processus.

© Copyright IBM Corp. 1994, 2014

Les agents de processus prennent en charge l'ajout dynamique des instructions de routage depuis Netcool/OMNIbus Administrator. Vous pouvez également ajouter un agent de processus à un groupe de routage comme suit :

1. Copiez et modifiez la configuration en cours à partir d'un agent de processus existant pour visualiser les processus en cours.
2. Mettez à jour les fichiers de configuration de chaque agent de processus. Pour ce faire, ajoutez la nouvelle entrée de routage à la zone de définition de routage des fichiers.
3. Arrêtez chaque agent de processus existant et leurs processus enfants. Pour ce faire, exécutez l'utilitaire **nco_pa_shutdown** à l'aide du paramètre `-option STOP`. Redémarrez ensuite chaque agent de processus afin de sélectionner le nouveau routage.

Remarque : Les noms de service et de processus doivent être uniques dans le réseau de contrôle de processus.

Tâches associées:

«Création et démarrage d'un système réseau de contrôle de processus», à la page 252
Pour gérer le contrôle de processus, vous devez en premier lieu déterminer vos exigences en matière de configuration du contrôle de processus, puis réaliser un certain nombre de tâches de configuration.

«Définition de processus, de services et d'hôtes pour le contrôle de processus», à la page 269
Pour exécuter le contrôle de processus, les processus, services et hôtes doivent être définis dans un fichier de configuration d'agent de processus. Au démarrage de l'agent de processus, ce dernier lit ce fichier afin d'établir les paramètres de configuration.

«Affichage et configuration des informations d'état d'un agent de processus», à la page 291
Vous pouvez afficher les détails de version d'un agent de processus auquel vous êtes connecté et modifier le niveau de consignation des messages que génère l'agent de processus. Vous pouvez également configurer le routage de l'hôte en ajoutant les agents de processus à un groupe de routage.

Référence associée:

«Affichage du statut des services et des processus (nco_pa_status)», à la page 281
Vous pouvez exécuter l'utilitaire **nco_pa_status** pour extraire le statut des services dans la configuration du système de contrôle de processus. Pour chaque service, l'utilitaire **nco_pa_status** retourne une liste de processus définis, le statut de chaque processus, et l'identificateur de processus.

«Ajout de services ou de processus (nco_pa_addentry)», à la page 285
Vous pouvez exécuter l'utilitaire **nco_pa_addentry** pour ajouter des services ou des processus lors de l'exécution de l'agent de processus.

Résolution du nom d'hôte au démarrage

Un hôte doit être spécifié pour chaque automatisation externe ou processus sous le contrôle d'un agent de processus. L'hôte est défini dans le fichier de configuration de l'agent de processus, dans le cadre de la définition d'un processus, ou bien dans l'automatisation externe du serveur ObjectServer.

Le fichier de configuration de l'agent de processus est `$NCHOME/omnibus/etc/nco_pa.conf`.

L'agent de processus peut gérer des noms et des adresses d'hôte spécifiés dans différents formats. Ceci est nécessaire car les détails des hôtes sont souvent extraits des éléments de données d'un événement ou d'une alerte. En fonction de la source de l'alerte, un hôte peut être spécifié sous forme d'adresse IP ou d'un nom d'hôte, ou encore sous une autre forme.

Au démarrage, l'agent de processus crée une liste interne de toutes les interfaces réseau de l'hôte local, en résolvant le nom de chaque interface réseau qu'il trouve. Cette liste perme tà l'agent de processus de mapper un hôte donné à un nombre quelconque de noms d'hôte d'interface locaux valides. Par exemple, le nom d'hôte par défaut sur une machine peut être «testbox», mais une requête peut être envoyée à l'agent de processus, demandant d'un processus soit exécuté sur «testbox.company.com», ou bien une requête peut spécifier l'adresse IPv4 ou IPv6 de l'hôte. Dans chaque cas, l'agent de processus doit reconnaître que l'hôte référencé est l'hôte sur lequel le processus s'exécute.

Si la résolution du nom d'hôte (par exemple à l'aide du DNS ou d'Active Directory) est lente ou n'est pas disponible, l'agent de processus peut être lent à démarrer car il doit attendre qu'un dépassement de délai d'attente se produise sur tous les noms d'hôte qui ne peuvent pas être résolus. De même, toute requête ultérieure utilisant un nom d'hôte inconnu ne sera pas exécutée. Dans le cas d'un processus configuré pour démarrer en tant que service Windows et si des problèmes de résolution DNS font qu'il démarre lentement, le service peut dépasser le délai d'attente au démarrage. Tous ces incidents sont consignés. Les administrateurs système peuvent utiliser ces informations pour résoudre les problèmes de configuration réseau sur les hôtes affectés.

Composants du contrôle de processus

Le contrôle de processus se compose d'agents de processus et des fichiers de configuration, des processus, des services et des utilitaires de contrôle de processus associés. Les processus sont organisés et exécutés dans des services. Les utilitaires de contrôle de processus permettent de gérer les agents de processus, les processus et les services.

Agents de processus

Les agents de processus sont des programmes qui sont installés sur chaque hôte pour gérer les processus d'un système de contrôle de processus. Tout hôte participant doit posséder un agent de processus et un fichier de configuration associé.

Le nombre d'agents de processus et le nombre d'hôte sont illimités. Les agents de processus peuvent gérer autant de processus que vous souhaitez.

Processus

Les processus sont des programmes exécutés par un agent de processus sur le même poste de travail dans un système de contrôle de processus. Les processus doivent être définis dans un *service* afin de faciliter la gestion.

Découverte du contrôle de processus

Un processus *PA aware* fait partie de la configuration du contrôle de processus et est au courant du contrôle de processus. Toutes les fonctions de contrôle de processus, telles que les dépendances de processus, peuvent être utilisées. Par exemple, les serveurs ObjectServer, les serveurs proxy et les passerelles du serveur ObjectServer tolèrent le processus PA aware. Un processus non PA aware peut être géré par le contrôle de processus, mais ne peut pas utiliser toutes les fonctions de contrôle de processus. Par exemple, le bureau Tivoli Netcool/OMNIbus n'est pas un processus PA aware.

Processus dépendants

L'ordre dans lequel les applications sont démarrées peut être important. Vous pouvez utiliser le contrôle de processus pour configurer les processus afin qu'ils soient dépendants entre eux *s'ils se trouvent dans le même service*. Par exemple, un processus peut être configuré pour démarrer uniquement après le démarrage d'un autre processus et la finalisation de plusieurs tâches de démarrage.

Un processus PA aware communique avec l'agent de processus. Lorsque le processus atteint le point de démarrage pendant lequel il reconnaît être exécuté, il envoie un message à l'agent de processus. Lorsque l'agent de processus reçoit ce message, il démarre les processus dépendants.

Tâches associées:

«Définition de processus, de services et d'hôtes pour le contrôle de processus», à la page 269

Pour exécuter le contrôle de processus, les processus, services et hôtes doivent être définis dans un fichier de configuration d'agent de processus. Au démarrage de l'agent de processus, ce dernier lit ce fichier afin d'établir les paramètres de configuration.

Services

Les processus doit être groupés en services dans un système de contrôle de processus. Vous pouvez grouper les processus associés dans un service afin de faciliter leur gestion.

Une fois un service correctement configuré, il peut être géré par un contrôle de processus.

Vous pouvez configurer un service afin qu'il démarre automatiquement lors du démarrage de l'agent de processus. Vous pouvez sinon démarrer un service manuellement.

Un service peut être configuré comme service maître duquel dépendent les autres services ou comme service non maître. Lorsque les services maîtres démarrent automatiquement via le contrôle de processus, ils doivent démarrer avant les services non maîtres.

Tâches associées:

«Définition de processus, de services et d'hôtes pour le contrôle de processus», à la page 269
Pour exécuter le contrôle de processus, les processus, services et hôtes doivent être définis dans un fichier de configuration d'agent de processus. Au démarrage de l'agent de processus, ce dernier lit ce fichier afin d'établir les paramètres de configuration.

Utilitaires de contrôle de processus

Les utilitaires de ligne de commande permettent de gérer les agents de processus, les processus et les services dans un système de contrôle de processus.

La table suivante répertorie ces utilitaires de ligne de commande.

Tableau 78. Utilitaires de ligne de commande de contrôle de processus

Utilitaire	Description
nco_pa_status	Cet utilitaire extrait et affiche le statut des services et processus contrôlés par l'agent de processus. Si un processus est défini comme non géré (`Managed = False`) dans le fichier de configuration d'agent de processus (`nco_pa.conf`), **nco_pa_status** peut renvoyer un état aléatoire lorsqu'il vérifie le processus. Par exemple, le statut renvoyé peut être `DEAD` lorsque le processus s'exécute correctement. Cela n'est pas une erreur et se produit car l'agent de processus démarre les processus non gérés mais ne les surveille pas.
nco_pa_start	Cet utilitaire démarre un service ou un processus situé n'importe où dans la configuration.
nco_pa_stop	Cet utilitaire arrête un service ou un processus situé n'importe où dans la configuration.
nco_pa_shutdown	Cet utilitaire interrompt un agent de processus.
nco_pa_addentry	Cet utilitaire ajoute une entrée de service ou de processus pendant l'exécution d'un agent de processus.

Référence associée:

«Affichage du statut des services et des processus (nco_pa_status)», à la page 281
Vous pouvez exécuter l'utilitaire **nco_pa_status** pour extraire le statut des services dans la configuration du système de contrôle de processus. Pour chaque service, l'utilitaire **nco_pa_status** retourne une liste de processus définis, le statut de chaque processus, et l'identificateur de processus.

«Démarrage d'un service ou d'un processus (nco_pa_start)», à la page 282
Vous pouvez exécuter l'utilitaire **nco_pa_start** pour démarrer un service ou un processus à n'importe quel emplacement de la configuration du système de contrôle de processus.

«Arrêt d'un service ou d'un processus (nco_pa_stop)», à la page 283
Vous pouvez exécuter l'utilitaire **nco_pa_stop** pour arrêter un service ou un processus à n'importe quel emplacement de la configuration du système de contrôle de processus.

«Fermeture d'un agent de processus (nco_pa_shutdown)», à la page 284
Vous pouvez exécuter l'utilitaire **nco_pa_shutdown** pour arrêter un agent de processus et éventuellement arrêter les services et processus associés.

«Ajout de services ou de processus (nco_pa_addentry)», à la page 285
Vous pouvez exécuter l'utilitaire **nco_pa_addentry** pour ajouter des services ou des processus lors de l'exécution de l'agent de processus.

Création et démarrage d'un système réseau de contrôle de processus

Pour gérer le contrôle de processus, vous devez en premier lieu déterminer vos exigences en matière de configuration du contrôle de processus, puis réaliser un certain nombre de tâches de configuration.

Avant de commencer

Déterminez les composants Tivoli Netcool/OMNIbus installés et leur emplacement. Vérifiez d'avoir pris en compte tous les composants et systèmes de reprise en ligne et de secours. Les bureaux Tivoli Netcool/OMNIbus ne sont pas gérés par le contrôle de processus.

Pourquoi et quand exécuter cette tâche

Voici un récapitulatif des tâches de configuration :

Procédure

1. Si vous utilisez le mécanisme d'authentification par défaut sous UNIX, configurez un groupe d'utilisateurs dédié que vous pouvez utiliser pour contrôler l'accès au système de contrôle de processus. Attribuez des utilisateurs à ce groupe.

   Sous Windows, l'accès au système de contrôle de processus est régi par un compte local ou un compte d'utilisateur de domaine valide. Tous les comptes utilisés pour vous connecter à l'ordinateur sur lequel un agent de processus s'exécute peuvent également être utilisés pour vous connecter à l'agent de processus, à condition que vous disposiez du droit `Access this computer from the network`.
2. Pour chaque agent de processus, configurez les informations de communication du serveur (c'est-à-dire un nom d'hôte et un numéro de port) sur l'ordinateur hôte et sur chaque ordinateur à connecter au réseau de contrôle de processus. Vous devez configurer les informations de communication du serveur dans l'éditeur de serveur (ou dans **nco_xigen**) avant de démarrer un agent de processus.
3. Mettez à jour le fichier de configuration d'agent de processus par défaut de chaque agent de processus en définissant des processus, des services et des hôtes.

Que faire ensuite

A l'issue de la configuration, démarrez les agents de processus. Vous pouvez démarrer un agent de processus manuellement à partir de la ligne de commande ou automatiquement au démarrage du système. Pour démarrer automatiquement un agent de processus, vous pouvez utiliser les scripts de démarrage (sous UNIX) ou installer et configurer l'agent de processus pour l'exécuter en tant que service Windows. Les processus s'exécutent automatiquement tels qu'ils sont définis dans le fichier de configuration de chaque agent de processus, tous les agents de processus communiquant tels qu'ils ont été configurés.

Tâches associées:

«Configuration et gestion du contrôle de processus à partir de la ligne de commande», à la page 269
Vous pouvez définir des processus, services et hôtes dans le fichier de configuration du contrôle de processus. Vous pouvez également utiliser les utilitaires de ligne de commande pour démarrer, arrêter et ajouter un service ou processus, afficher l'état des services et processus, et arrêter un agent de processus.

Création de groupes d'utilisateurs UNIX pour le système de contrôle de processus

Le démon de contrôle de processus permet de contrôler les personnes qui s'y connectent. Sous UNIX, tous les utilisateurs qui souhaitent accéder au système de contrôle de processus doivent appartenir à groupe d'utilisateurs UNIX que vous identifiez comme un groupe d'administration à cette fin.

Pourquoi et quand exécuter cette tâche

Par défaut, le système de contrôle de processus utilise des noms d'utilisateur et des mots de passe UNIX pour autoriser l'accès. Lors de l'exécution du démon d'agent de processus (**nco_pad**), vous pouvez indiquer d'autres modes d'autorisation pris en charge à l'aide de l'option de ligne de commande `-authenticate`.

Vous pouvez utiliser un groupe d'utilisateurs UNIX existant ou en créer un, et ajouter des utilisateurs de contrôle de processus à ce groupe. Si vous exécutez NIS, NIS+ ou un autre service d'informations global, cette configuration doit être réalisée par l'administrateur de ce service. Consultez la documentation accompagnant votre système d'exploitation pour obtenir des informations relatives à ces groupes d'utilisateurs.

Lors de l'exécution d'un démon de contrôle de processus, identifiez le groupe d'administration avec l'option de ligne de commande -admingroup. Si vous ne précisez aucun nom de groupe, le contrôle de processus vérifie si l'utilisateur est un membre du groupe par défaut ncoadmin.

Avertissement : Si vous utilisez les modules PAM pour procéder à l'authentification, il n'est pas indispensable que les utilisateurs appartiennent à un groupe d'utilisateurs UNIX (ncoadmin, par exemple) pour pouvoir accéder au système de contrôle de processus. Avec les clients PAM, le système de contrôle de processus ne valide pas les utilisateurs en fonction d'un groupe d'utilisateurs UNIX, l'accès n'étant donc pas restreint.

Concepts associés:

«Services», à la page 250
Les processus doit être groupés en services dans un système de contrôle de processus. Vous pouvez grouper les processus associés dans un service afin de faciliter leur gestion.

Référence associée:

«Propriétés des agents de processus et options de ligne de commande», à la page 257
Lors de l'exécution de l'agent de processus à l'aide de la commande **nco_pad**, vous pouvez spécifier des options de ligne de commande supplémentaires pour affiner la configuration.

Exigences des comptes Windows pour le système de contrôle de processus

Sous Windows, l'agent de processus doit être exécuté en tant qu'administrateur sur l'ordinateur local.

Pour se connecter à un agent de processus depuis Netcool/OMNIbus Administrator, d'un autre agent de processus, du serveur ObjectServer ou d'un utilitaire de contrôle de processus, vous avez besoin des types de compte suivants :

- Un compte utilisateur Windows local
- Un compte de domaine Windows local
- Un compte au format UPN (User Principal Name) (c'est-à-dire *nom_utilisateur@nom_domaine_DNS*

Pour des connexions d'égal à égal entre les agents de processus, utilisez des règles sur les mots de passe qui verrouillent les comptes Windows après un nombre défini de tentatives. Les agents de processus essaient de se connecter plusieurs fois s'ils n'y parviennent pas ; par conséquent, l'utilisation d'un mot de passe erroné peut provoquer le verrouillage de vos comptes Windows.

Configuration des informations de communication du serveur pour les agents de processus

Vous devez utiliser l'éditeur de serveur pour attribuer un nom de serveur unique à chaque agent de processus et indiquer d'autres informations de communication, puis mettre ces informations à disposition de chaque ordinateur hôte du système de réseau de contrôle de processus. Cela permet à tous les agents de processus installés sur des ordinateurs hôte de communiquer entre eux.

Pourquoi et quand exécuter cette tâche

Pour configurer les informations de communication du serveur pour les agents de processus, procédez comme suit :

Procédure

1. Démarrez l'éditeur de serveur sur un ordinateur hôte et ajoutez une entrée de serveur à chaque agent de processus à inclure dans le système de réseau de contrôle de processus. Sauvegardez ces informations.

   Conseil : Le nom d'une entrée serveur doit être composé de 29 lettres en majuscules au maximum et ne peut pas commencer par un entier. La convention de dénomination implique également d'ajouter _PA au nom, de manière à pouvoir aisément identifier le serveur comme étant un agent de processus dans l'éditeur de serveur. Par exemple, si vous configurez l'agent de processus sur un hôte nommé sfosys1, il peut s'appeler SFOSYS1_PA. Par défaut, le premier agent de processus installé dans une configuration s'appelle NCO_PA.
2. Sous UNIX, générez un fichier d'interfaces contenant les informations de communication du serveur. En règle générale, le fichier d'interfaces s'appelle `$NCHOME/etc/interfaces.`*`arch`*, où *arch* et le nom du système d'exploitation UNIX.

   Sous Windows, configurez les communication du serveur sur chaque ordinateur Windows.

3. Sous UNIX, distribuez le fichier d'interfaces mis à jour à tous les postes de travail hôtes de la configuration.

Que faire ensuite

Voir le *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus* pour plus d'informations sur la configuration des définitions de serveur dans l'éditeur de serveur et sur la génération des fichiers d'interfaces.

Mise à niveau du fichier de configuration de contrôle de processus par défaut

Un fichier de configuration de contrôle de processus est installé pour chaque agent de processus. Ce fichier contient les définitions de chaque processus, service et hôte dans la configuration du système de contrôle de processus.

Avant de commencer

Le nombre de processus par service pouvant être édités via l'interface graphique d'administration est limité à 84 dans le fichier `nco_pa.conf`. Si vous voulez utiliser cette interface pour éditer plus de 84 processus, vous devez exécuter plusieurs processus nco_pad avec chacun son propre fichier `nco_pa.conf`.

Pourquoi et quand exécuter cette tâche

Le fichier de configuration de contrôle de processus `nco_pa.conf` se trouve dans le répertoire `$NCHOME/omnibus/etc`.

Déterminez le processus à exécuter dans le contrôle de processus et identifiez les dépendances de processus. Editez manuellement le fichier de configuration de contrôle de processus pour configurer les définitions de contrôle de processus :

Procédure

- Créez des définitions de service afin de grouper les processus connexes ou dépendants. Cela permet de déterminer l'ordre d'exécution des processus.
- Créez des définitions de routage pour spécifier chaque agent de processus et son ordinateur hôte associé à inclure dans la configuration.

Résultats

Au démarrage, l'agent de processus lit ce fichier pour établir les paramètres de configuration.

Tâches associées:

«Définition de processus, de services et d'hôtes pour le contrôle de processus», à la page 269
Pour exécuter le contrôle de processus, les processus, services et hôtes doivent être définis dans un fichier de configuration d'agent de processus. Au démarrage de l'agent de processus, ce dernier lit ce fichier afin d'établir les paramètres de configuration.

Démarrage manuel des agents de processus

Vous pouvez démarrer manuellement les agents de processus à partir de la ligne de commande.

Pourquoi et quand exécuter cette tâche

Pour démarrer manuellement un agent de processus, entrez la commande suivante sur la ligne de commande de l'hôte :

```
$NCHOME/omnibus/bin/nco_pad -name agent_processus
```

Dans cette commande, *agent_processus* est le nom de l'agent de processus, tel que défini dans le fichier `omni.dat` (UNIX) ou `sql.ini` (Windows). Vous pouvez spécifier des options de ligne de commande supplémentaires avec cette commande.

Le démon d'agent de processus (**nco_pad**) s'exécute en fonction de l'emplacement `$NCHOME/omnibus`.

UNIX Le démon d'agent de processus (**nco_pad**) suit les règles de citation UNIX concernant le développement des variables d'environnement dans les arguments pour les actions externes. Les règles sont les suivantes, par ordre de priorité :

- Des guillemets doubles (" ") dans une sous-chaîne délimitée par des guillemets simples n'ont aucun effet.
- Des guillemets simples (' ') dans une sous-chaîne délimitée par des guillemets doubles n'ont aucun effet.
- Des variables d'environnement dans des guillemets simples ne doivent pas être développées.

Voir la section «Exemple» pour plus d'informations.

Remarque : Une nouvelle instance de l'agent de processus ne peut pas gérer les processus qui ont été démarrés par une autre instance et qui sont toujours en cours d'exécution. Lorsque l'agent de processus est arrêté et redémarré, il n'a pas connaissance de ces processus et lance donc de nouvelles instances. Les instances précédentes sont encore en cours d'exécution.

Remarque : Si vous exécutez la commande **nco_pad** sur un ordinateur qui contient déjà un agent de processus installé en tant que service Windows, toutes les commandes spécifiées pour le service Windows sont fusionnées avec les options de ligne de commande pour exécuter **nco_pad** dans l'invite de commande. En cas de conflit entre les options spécifiées pour le service et les options que vous entrez à l'invite de commande, ces dernières l'emportent. La sortie à l'écran affiche les options fusionnées.

Exemple

UNIX Dans l'exemple suivant, seule `$B` est développée, car `$B` se situe à l'extérieur d'une paire de guillemets simples :

```
'$A'$B'$C'
```

L'exemple suivant indique comment éviter le développement et conserver les guillemets dans la sortie. Pour obtenir des guillemets simples dans la sortie, vous devez soustraire la chaîne à l'intérieur des guillemets doubles comme suit :

```
"'"'$A'"'"'$B'"'"'$C'"'"
```

Cette chaîne génère la sortie suivante :

```
'$A'$B'$C'
```

Pour obtenir la sortie `'$A'$B'$C'` sans développement, il ne suffit pas de placer la chaîne entre guillemets doubles car, selon les règles de citation UNIX, la chaîne serait développée. Pour qu'aucun développement ne se produise, la chaîne entière doit être placée dans des guillemets simples.

Propriétés des agents de processus et options de ligne de commande

Lors de l'exécution de l'agent de processus à l'aide de la commande **nco_pad**, vous pouvez spécifier des options de ligne de commande supplémentaires pour affiner la configuration.

Fix Pack 1 Une autre solution consiste à entrer les définitions dans un fichier de propriétés et à exécuter la commande **nco_pad** avec l'option `-propsfile`.

Les options de ligne de commande de la commande `$NCHOME/omnibus/bin/nco_pad` sont décrites dans le tableau ci-après. Les options de ligne de commande spécifiques à UNIX et Linux, non prises en charge sous Windows, sont signalées dans le tableau. Les propriétés sont applicables uniquement si vous avez installé le groupe de correctifs. 1. Placez les valeurs de chaîne entre guillemets simples - exemple : `AdminGroupName: 'ncoadmin'`.

Tableau 79. Options de ligne de commande nco_pad du démon d'agent de processus

Propriété	Option de ligne de commande	Description
UNIX Linux **AdminGroupName** *chaîne*	`-admingroup` *chaîne*	Spécifie le nom du groupe d'utilisateurs UNIX disposant des droits d'administrateur. Les membres de ce groupe peuvent accéder au système de contrôle de processus. Default `ncoadmin` **Remarque :** L'option `-admingroup` ne s'applique qu'au mode d'authentification UNIX.
ApiCheck `TRUE|FALSE`	`-apicheck`	Permet de vérifier l'API Sybase. Default `FALSE`

Tableau 79. Options de ligne de commande nco_pad du démon d'agent de processus (suite)

Propriété	Option de ligne de commande	Description
Authenticate *chaîne*	`-authenticate` *chaîne*	Indique le mode d'authentification à utiliser pour vérifier les données d'identification d'un utilisateur ou d'un démon d'agent de processus distant. Les valeurs possibles dépendent du système d'exploitation. Seule l'option `PAM` peut être indiquée pour l'authentification. UNIX Linux Les valeurs admises sont les suivantes : • Default `UNIX` : La fonction Posix getpwnam ou getspnam est utilisée pour vérifier les données d'identification de l'utilisateur. Selon la configuration du système, les mots de passe sont vérifiés à l'aide du fichier `/etc/password`, le fichier de mots de passe shadow `/etc/shadow`, NIS ou NIS+. Si l'agent de processus s'exécute en tant qu'utilisateur non-root, assurez-vous que cet utilisateur dispose d'un accès en lecture au répertoire `/etc/shadow`. • Linux Solaris `PAM` : Si le module PAM est spécifié en tant que mode d'authentification, les modules PAM sont utilisés pour vérifier les données d'identification de l'utilisateur. Le nom de service utilisé par la passerelle lors de l'initialisation de l'interface PAM est netcool. • Solaris `KERBEROS` : Si le module KERBEROS est spécifié en tant que mode d'authentification, l'authentification KERBEROS IV est utilisée pour vérifier les données d'identification de l'utilisateur. Un serveur d'authentification Kerberos IV est nécessaire pour cette option. • `none` : Aucune authentification n'est tentée. L'invite du démon de l'agent de processus accepte toute donnée d'identification de connexion. Vous ne devez utiliser cette option qu'à des fins de test. Ne l'utilisez pas dans un environnement de production. Windows Les valeurs admises sont les suivantes : • `WINDOWS` : Il s'agit du mode d'authentification par défaut, dans le cadre duquel l'agent de processus s'authentifie par rapport au compte Windows. • `none` : Aucune authentification n'est tentée. L'invite du démon de l'agent de processus accepte toute donnée d'identification de connexion. Vous ne devez utiliser cette option qu'à des fins de test. Ne l'utilisez pas dans un environnement de production.
ConfigFile *chaîne*	`-configfile` *chaîne*	Chemin d'accès au fichier de configuration de l'agent de processus. Default `$NCHOME/omnibus/etc/nco_pa.conf`
Connections *chaîne*	`-connections` *entier*	Définit le nombre maximal de connexions disponibles pour exécuter les actions externes. Default `30`

Tableau 79. Options de ligne de commande nco_pad du démon d'agent de processus (suite)

Propriété	Option de ligne de commande	Description
N/D	`-cryptalgorithm` *chaîne*	Indique l'algorithme de cryptographie à utiliser pour le déchiffrement des valeurs de chaîne chiffrées à l'aide de l'utilitaire **nco_aes_crypt** et stockées ensuite dans le fichier de configuration de l'agent de processus. Définissez la valeur *chaîne* comme suit : • 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 codés à l'aide des outils fournis dans des versions antérieures à Tivoli Netcool/OMNIbus version 7.2.1. La valeur que vous spécifiez doit être identique à celle utilisée lors de l'exécution de la commande **nco_aes_crypt** avec le paramètre `-c` pour coder les mots de passe dans la section de définition de routage du fichier. Utilisez l'option de ligne de commande `-cryptalgorithm` en association avec l'option `-keyfile`.
N/D	`-debug` *entier*	Fix Pack 1 Cette option est obsolète. Utilisez la propriété **messagelevel** en lieu et place. Active le débogage. La valeur *entier* indique la quantité d'informations de débogage entrées dans le fichier journal. Les niveaux disponibles sont `1` (Debug), `2` (Information), `3` (Warning), `4` (Error), et `5` (Fatal). La valeur par défaut est `3`. Lors d'une exécution au niveau de débogage 1, l'agent de processus consigne des informations sur les processus qu'il est sur le point démarrer. Ces informations comprennent le chemin d'accès au programme, chaque argument de ligne de commande et l'ID utilisateur effectif du processus. Le cas échéant, le journal comprend également le répertoire de travail en cours et, pour UNIX, l'ID groupe effectif et le umask (en octal) du processus. Lors d'une exécution au niveau de débogage 2, le journal contient un message indiquant le compteur utilisateur sous lequel l'agent de processus est exécuté.
DnsHostname *chaîne*	`-DNS` *chaîne*	Indique une valeur remplaçant le nom d'hôte dans les environnements DNS. Assurez-vous que la même valeur est indiquée dans le fichier de configuration de l'agent de processus.
N/D	`-help`	Affiche les informations d'aide sur l'agent de processus et quitte.
N/D	`-keyfile` *chaîne*	Indique le chemin et le nom du fichier qui contient la clé à utiliser pour décoder les mots de passe codés qui sont stockés dans le fichier de configuration d'agent de processus. Le fichier clé que vous spécifiez doit être identique à celui utilisé lorsque vous avez exécuté l'utilitaire **nco_aes_crypt** avec le paramètre `-k`, pour coder les mots de passe dans la section de définition de routage du fichier. Utilisez l'option de ligne de commande `-keyfile` en association avec l'option `-cryptalgorithm` pour décoder les mots de passe.

Tableau 79. Options de ligne de commande nco_pad du démon d'agent de processus (suite)

Propriété	Option de ligne de commande	Description
KillProcessGroup	UNIX Linux `-killprocessgroup`	Si cette option est spécifiée lorsque le démon d'agent de processus arrête un processus, il envoie également un signal pour interrompre tout processus dans le même groupe de processus du système d'exploitation.
N/D	`-logfile` *chaîne*	Fix Pack 1 Cette option est obsolète. Utilisez la propriété **messagelog** en lieu et place. Spécifie un autre fichier journal. Sous UNIX, le journal peut être redirigé vers stderr et stdout. Sous Windows, le journal est toujours inscrit dans un fichier. Default `$NCHOME/omnibus/log/`*nom_ap*`.log`, où *nom_ap* représente le nom de l'agent de processus indiqué avec l'option `-name`.
N/D	`-logsize` *entier*	Fix Pack 1 Cette option est obsolète. Utilisez la propriété **maxlogfilesize** en lieu et place. Spécifie la taille maximale du fichier journal en ko. La taille par défaut est 1024 ko et la taille minimale est 16 ko.
MaxLogFileSize *entier*	Fix Pack 1 `-maxlogfilesize` *entier*	Taille maximale du fichier journal en ko. Default 1024
MessageLevel *entier*	Fix Pack 1 `-messagelevel` *entier*	Indique le niveau de consignation du message. Valeurs possibles : `debug`, `info`, `warn`, `error` et `fatal`. Les messages qui sont consignés à chaque niveau sont les suivants : • `fatal` : messages de type `fatal` uniquement • `error` : messages de types `fatal` et `error` • Default `warn` : messages de types `fatal`, `error` et `warn` • `info` : messages de types `fatal`, `error`, `warn` et `info` • `debug` : messages de types `fatal`, `error`, `warn`, `info` et `debug` **Conseil :** La valeur de *chaîne* peut être être composée de majuscules et/ou de minuscules.
MessageLog *entier*	Fix Pack 1 `-messagelog` *entier*	Chemin du fichier journal des messages. `$OMNIHOME/log/NCO_PA.log`
MessagePoolSize *entier*	`-msgpoolsize` *entier*	Spécifie le nombre de messages disponibles dans l'agent de contrôle de processus.
Name *chaîne*	`-name` *chaîne*	Spécifie le nom du serveur pour l'agent de processus. Si cette option n'est pas spécifiée, le nom d'agent de processus par défaut est `NCO_PA`.
NewLog `TRUE|FALSE`	`-newlog`	Cette option est obsolète. L'agent de processus remplace toujours le fichier journal précédent.
NoAutoStart `TRUE|FALSE`	`-noautostart`	Si cette option n'est pas spécifiée, l'agent de processus ne provoque le démarrage automatique d'aucun service, même si ce démarrage automatique est défini dans le fichier `nco_pa.conf`. Default `TRUE`

Tableau 79. Options de ligne de commande nco_pad du démon d'agent de processus (suite)

Propriété	Option de ligne de commande	Description
NoConfig TRUE|FALSE	-noconfig	Si cette option n'est pas spécifiée, l'agent de processus ne lit pas le fichier de configuration nco_pa.conf. Le contrôle de processus démarre sans informations de configuration.
UNIX Linux **NoDaemon** TRUE|FALSE	-nodaemon	Par défaut, le contrôle de processus passe en arrière-plan pour une exécution en tant que processus de démon. Lorsque -nodaemon est spécifié, le processus s'exécute au premier plan.
OldTimeStamp TRUE|FALSE	Fix Pack 1 -oldtimestamp	Le journal des messages **nco_pad** utilise un horodatage formaté ISO8601 - exemple : *2014-06-30 17:20:07*. Cette option repasse l'horodatage du journal des messages à l'ancien format - exemple *Jun 30 17:20:07 2014*. Cet ancien format n'est pas cohérent par rapport aux horodatages générés par les applications utilisant NLog(), dont le format est ISO8601.
Password *chaîne*	-password *chaîne*	Spécifie le mot de passe à utiliser pour la connexion à d'autres agents de processus.
UNIX Linux **PidFile** *chaîne*	-pidfile *chaîne*	Spécifie le chemin relatif à $NCHOME/omnibus pour l'accès au fichier dans lequel le PID du démon de contrôle de processus est stocké. Un fichier PID est nécessaire pour chaque démon d'agent de processus. Default $NCHOME/omnibus/var/*nom_ap*.pid, où *nom_ap* représente le nom de l'agent de processus. Sous réserve que tous les agents de processus possèdent des noms uniques, il n'est pas nécessaire de modifier ce paramètre. Cela permet d'exécuter plus d'un démon d'agent de processus sur le même ordinateur. **Conseil :** Windows Sous Windows, il n'existe aucune restriction relative aux nombre d'agents de processus exécutables sous les services Windows du même hôte.
UNIX Linux **PidMessagePoolSize** *entier*	-pidmsgpool *entier*	Spécifie la taille du pool de messages de gestion des signaux.
Props.CheckNames *chaîne*	Fix Pack 1 -propsfile *chaîne*	Chemin du fichier de propriétés. Default $OMNIHOME/etc/ncp_pa.props
UNIX Linux **RedirectFile** *chaîne*	-redirectfile *chaîne*	Spécifie un fichier vers lequel doivent être dirigés les messages stderr et stdout des processus démarrés par l'agent de processus. Cette option est utile à des fins de dépannage. Windows Vous pouvez obtenir le même résultat en exécutant le processus à partir de l'interface de ligne de commande. À chaque processus est associée une fenêtre de console dans laquelle la sortie du traitement est affichée. Si l'agent de processus est exécuté en tant que service Windows, vous pouvez également afficher les propriétés de ce service, sélectionner **Local System account** dans l'onglet **Log On**, puis sélectionner **Allow service to interact with desktop**.

Tableau 79. Options de ligne de commande nco_pad du démon d'agent de processus (suite)

Propriété	Option de ligne de commande	Description
RetryTime *entier*	`-retrytime` *entier*	Spécifie le nombre de secondes d'exécution d'un processus démarré par contrôle de processus pour que ce processus soit considéré comme ayant abouti. Default `5` L'agent de processus tente de redémarrer un processus si le processus est interrompu. Si le processus est interrompu après *durée_nouvelle_tentative* secondes, l'agent de processus tente de redémarrer le processus immédiatement. Si le processus est interrompu avant *durée_nouvelle_tentative* secondes, l'agent de processus tente de redémarrer le processus au débit exponentiel de 2, 4, 8, 16, 32, ..., 256 secondes. L'agent de processus réinitialise l'intervalle de temporisation après huit tentatives de démarrage du processus. Si l'exécution du processus échoue pendant plus de *durée_nouvelle_tentative* secondes, le nombre de nouvelles tentatives (RetryCount) (spécifié dans la définition du processus) de ce processus est également décrémenté. Si le processus s'exécute correctement pendant au moins *durée_nouvelle_tentative* secondes, le nombre de nouvelles tentatives (RetryCount) récupère sa valeur d'origine. Si le nombre de nouvelles tentatives (RetryCount) est égal à 0, aucune limite de tentatives de redémarrage n'est définie.
RogueTimeout *entier*	`-roguetimeout` *entier*	Spécifie la durée d'attente en secondes pour la fermeture du processus. Default `30` Le temps minimal est de 5 secondes.
SecureMode `TRUE|FALSE`	`-secure`	Si l'option `-secure` est spécifiée, tous les clients doivent s'authentifier eux-mêmes avec un nom d'utilisateur et un mot de passe valides, spécifiés via les options de ligne de commande `-user` et `-password`. En mode non-FIPS 140–2, les informations de connexion sont automatiquement chiffrées en transmission lorsque l'agent de processus se connecte à un autre agent de processus. En mode FIPS 140–2, les informations de connexion sont transmises en texte clair et la couche SSL doit être utilisée si vous avez besoin d'un chiffrement pendant la transmission.
StackSize *entier*	`-stacksize` *entier*	Spécifie la taille de la pile de l'unité d'exécution.
UNIX Linux **TicketDir** *chaîne*	`-ticketdir` *chaîne*	Répertoire des tickets Kerberos si l'option `-authenticate` est définie sur KERBEROS.
TraceEventQueue	`-traceevtq`	Active le traçage de l'activité de file d'attente d'événements.
TraceMessageQueue	`-tracemsgq`	Active le traçage de l'activité de la file d'attente de messages.
TraceMutexLocks	`-tracemtx`	Active le traçage des verrous mutex.
TraceNetLibrary	`-tracenet`	Active le traçage de la bibliothèque net.

Tableau 79. Options de ligne de commande nco_pad du démon d'agent de processus (suite)

Propriété	Option de ligne de commande	Description
User *entier*	`-user` *chaîne*	Spécifie le nom d'utilisateur employé pour se connecter à un autre agent de processus. Si vous ne définissez pas cette option, c'est l'utilisateur exécutant la commande **nco_pad** qui établit la connexion. Cette option doit être définie si vous vous connectez à un agent de processus exécuté en mode sécurisé (via l'option `-secure`). Ce nom d'utilisateur et le mot de passe associé (indiqué via l'option `-password`) sont utilisés si aucune donnée d'identification de connexion n'est spécifiée dans la section de routage du fichier de configuration de contrôle de processus.
Version	`-version`	Affiche les informations de version sur l'agent de processus et quitte.
UNIX Linux **WalkHostTable**	`-walkhosttable`	Vérifie tous les alias en parcourant la table hôte.

Considérations concernant le sécurité de agent de processus

Si l'agent de processus est exécuté en tant qu'utilisateur disposant de certains privilèges ou en tant que superutilisateur sur la machine hôte, il est possible pour un administrateur Netcool/OMNIbus de configurer les actions externes qui sont ensuite exécutées sur le système hôte en tant qu'utilisateur disposant de certains privilèges. Par exemple, en tant que `superutilisateur` (UNIX) ou qu'utilisateur de système local (Windows). Cela présente un éventuel risque de sécurité. C'est la raison pour laquelle l'agent de processus doit être exécuté en tant qu'utilisateur ne disposant pas de privilège.

Remarque : Il existe des cas où l'agent de processus doit être exécuté en tant qu'utilisateur disposant de certains privilèges. Par exemple, lors de l'exécution de l'analyse du protocole SNMP nécessitant l'ouverture du port 162 et lorsque l'authentification de fichier local est en cours d'utilisation et que l'agent de processus doit être en mesure de lire le fichier `/etc/shadow` afin d'authentifier les utilisateurs (UNIX uniquement).

Tâches associées:

«Exécution de l'agent de processus en tant qu'utilisateur non privilégié (UNIX)», à la page 264
Utilisez cette procédure pour configurer un agent de processus à exécuter en tant qu'utilisateur non privilégié sur les systèmes d'exploitation UNIX.

«Exécution de l'agent de processus en tant qu'utilisateur non privilégié (Windows)», à la page 264
Utilisez cette procédure pour configurer un agent de processus à exécuter en tant qu'utilisateur non privilégié sous Windows.

«Exécution de l'agent de processus en tant qu'utilisateur privilégié», à la page 265
Utilisez cette procédure pour configurer un agent de processus à exécuter en tant qu'utilisateur privilégié.

Exécution de l'agent de processus en tant qu'utilisateur non privilégié (UNIX)

Utilisez cette procédure pour configurer un agent de processus à exécuter en tant qu'utilisateur non privilégié sur les systèmes d'exploitation UNIX.

Procédure

Pour configurer un agent de processus à exécuter en tant qu'utilisateur non privilégié sous UNIX, procédez comme suit :

1. En tant que superutilisateur, installez le script de démarrage de l'agent de processus. Le script de démarrage se trouve dans le répertoire `/etc/`.
2. En tant que superutilisateur, localisez l'entrée suivante dans le script de démarrage. Cette entrée permet de démarrer l'agent de processus en tant que superutilisateur.

```
if [ "$SECURE" = "Y" ]; then
${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -authenticate PAM -
secure > /dev/null 2> /dev/null
else
${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -authenticate PAM >
/dev/null 2> /dev/null
fi
```

3. Pour démarrer l'agent de processus en tant qu'utilisateur «netcool», remplacez l'entrée indiquée à l'étape 2 par l'entrée suivante :

```
if [ "$SECURE" = "Y" ]; then
su - netcool -c "${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -
authenticate PAM -secure > /dev/null 2> /dev/null"
else
su - netcool -c "${OMNIHOME}/bin/nco_pad -name ${NCO_PA} -
authenticate PAM > /dev/null 2> /dev/null"
fi
```

Résultats

L'agent de processus s'exécute en tant qu'utilisateur local netcool. Les droits d'accès aux actions externes sur le serveur ObjectServer sont limités aux privilèges de cet utilisateur sur l'ordinateur hôte.

Exécution de l'agent de processus en tant qu'utilisateur non privilégié (Windows)

Utilisez cette procédure pour configurer un agent de processus à exécuter en tant qu'utilisateur non privilégié sous Windows.

Procédure

Pour configurer un agent de processus à exécuter en tant qu'utilisateur non privilégié sous Windows, procédez comme suit :

1. En tant qu'administrateur, installez l'agent de processus en tant que service Windows. Pour plus d'informations, voir Exemple : installation et exécution de l'agent de processus en tant que service.
2. Dans le panneau de configuration Windows, cliquez deux fois sur **Administrative Tools (Outils d'administration)**, puis sur **Services**.
3. Dans la fenêtre Services Details (Détails des services), sélectionnez le nouveau service d'agent de processus, puis cliquez sur **Action(s)** > **Properties**.

4. Cliquez sur l'onglet **Log On (Connexion)** et changez la sélection du bouton radio **Log on as: (Connecté en tant que)** en la faisant passer de **Local System account (Compte de système local)** à **This account: (Ce compte :)**, puis sélectionnez l'utilisateur local `netcool`.
5. Cliquez sur **OK**.

   Remarque : Vous pouvez redémarrer le service s'il a été précédemment automatiquement démarré en tant qu'utilisateur privilégié.

   L'agent de processus est exécuté en tant qu'utilisateur local `netcool` et les autorisations d'action externe ObjectServer sont désormais limitées aux privilèges d'utilisateur `netcool` sur le système hôte.

Exécution de l'agent de processus en tant qu'utilisateur privilégié

Utilisez cette procédure pour configurer un agent de processus à exécuter en tant qu'utilisateur privilégié.

Pourquoi et quand exécuter cette tâche

Il existe des cas où l'agent de processus doit être exécuté en tant qu'utilisateur disposant de certains privilèges. Par exemple, lors de l'exécution de l'analyse du protocole SNMP nécessitant l'ouverture du port 162 ou lorsque l'authentification de fichier local est en cours d'utilisation et que l'agent de processus doit être en mesure de lire le fichier `/etc/shadow` afin d'authentifier les utilisateurs (UNIX uniquement).

Pour configurer un agent de processus à exécuter en tant qu'utilisateur privilégié :

Procédure

1. Installez et exécutez l'agent de processus principal en tant qu'utilisateur privilégié et configurez-le pour qu'il soit exécuté sur la machine hôte à l'aide du script de démarrage (UNIX) ou d'un service (Windows).
2. Exécutez un agent de processus secondaire en tant qu'utilisateur non privilégié et configurez-le pour qu'il exécute des actions ObjectServer externes. Vous devez également le configurer pour l'exécuter en mode non démon en tant que processus enfant de l'agent de processus principal. C'est la raison pour laquelle vous devez créer une entrée de processus supplémentaire dans votre fichier de configuration d'agent de processus principal, comme le montre l'exemple ci-dessous :

   ```
   nco_process 'NON_ROOT_PA'
   {
   Command '$OMNIHOME/bin/nco_pad -name NON_ROOT_PA -nodaemon -configfile
   $OMNIHOME/etc/NON_ROOT_PA.conf' run as 1000
   Host = 'hostx'
   Managed         =       True
   RestartMsg  =  '${NAME} running as ${EUID} has been restored on ${HOST}.'
   AlertMsg = '${NAME} running as ${EUID} has died on ${HOST}.'
   RetryCount      =       0
   ProcessType     =       PaPA_AWARE
   }
   ```

   Remarques supplémentaires :

   - Dans l'exemple précédent, l'agent de processus secondaire est paramétré pour être démarré en tant qu'utilisateur `1000` (UID). Il est également paramétré pour démarrer avec le commutateur de démarrage `-nodaemon`.

Cela l'empêche de bifurquer vers un processus enfant et lui permet d'être géré par l'agent de processus principal.

- Vous devez définir un fichier de configuration pour l'agent de processus secondaire. Les autorisations pour le fichier de configuration doivent lui permettre d'être accessible par l'utilisateur qui exécute l'agent de processus secondaire.
- Vous devez modifier le fichier de propriétés associé au serveur ObjectServer qui est actuellement exécuté en tant que processus via l'agent de processus principal. Cela implique le paramétrage des propriétés suivantes : **PA.Name**, **PA.Username** et **PA.Password**.

Démarrage automatique des agents de processus sous UNIX

Sous UNIX, des scripts de démarrage permettent de démarrer automatiquement l'agent de processus lorsque le système démarre.

Avant de commencer

Vous pouvez modifier les scripts de démarrage avant de les installer, le cas échéant.

Sur les systèmes d'exploitation Linux, modifiez le script de démarrage appelé **nco** pour la version de Linux que vous utilisez. Le script contient des sections pour les versions Red Hat et SUSE de Linux et ces sections sont délimitées par des commentaires dans le script comme indiqué ci-dessous :

```
### REDHAT ONLY
...
### END REDHAT ONLY


### SUSE ONLY
...
### END SUSE ONLY
```

Notez que le script contient un certain nombre de sections distinctes pour les versions Red Hat and SUSE de Linux.

Toujours sous Linux, modifiez le script **nco** si vous utilisez les modules d'authentification enfichables (PAM) pour l'authentification en ajoutant l'argument `-authentication PAM` au script. Pour l'authentification UNIX par défaut, vous n'avez pas à ajouter quoi que ce soit.

Pourquoi et quand exécuter cette tâche

Ces scripts se trouvent dans le répertoire suivant :

```
$NCHOME/omnibus/install/startup
```

Ce répertoire contient l'un des scripts d'installation suivants, selon le système d'exploitation :

- **aix5install**
- **hpux11hpiainstall**
- **solaris2install**
- **linux2x86install**
- **linux2s390install**

Pour utiliser les scripts de démarrage d'agent de processus, vous devez exécuter le script d'installation correspondant à votre système d'exploitation (vous devez préalablement le rendre exécutable).

Pour installer les scripts de démarrage d'agent de processus, procédez comme suit :

Procédure

1. Exécutez le script d'installation en tant qu'utilisateur `root`. Par exemple, pour installer les scripts sous Solaris, exécutez **`solaris2install`** à partir de l'emplacement `$NCHOME/omnibus/install/startup`. La sortie suivante s'affiche :

   ```
   Nom du démon de l'agent de processus [NCO_PA]
   ```

2. Appuyez sur Entrée pour accepter le nom du serveur de l'agent de processus par défaut NCO_PA ou entrez un autre nom de serveur. La sortie suivante s'affiche :

   `Should pa_name run in secure mode (y/n)? [y]` (pa_name doit-il s'exécuter en mode sécurisé (o/n) [o]

3. Appuyez sur Entrée pour inclure l'option de ligne de commande `-secure` au démarrage de l'agent de processus. Le mode sécurisé permet de contrôler l'authentification des demandes de connexion avec un nom d'utilisateur et un mot de passe.

   Le message suivant s'affiche :

   ```
   Enter value for environment variable NETCOOL_LICENSE_FILE,
   if required [27000@localhost]:
   ```

   Remarque : Bien que l'exécution de Tivoli Netcool/OMNIbus n'implique pas la présence d'une clé de licence, certaines sondes et passerelles n'ayant pas fait l'objet d'un récent cycle de maintenance demandent encore des clés de licence. Si vous exécutez ces anciens modules de sonde ou de passerelle, la variable d'environnement NETCOOL_LICENSE_FILE doit toujours être définie ainsi que la disponibilité d'un serveur de licences Netcool.

4. Si vous n'utilisez pas de serveur de licences, vous pouvez en toute sécurité appuyer sur Entrée pour exécuter le script. Si vous détenez un serveur de licences, appuyez sur Entrée pour accepter la valeur par défaut de la variable d'environnement de licence ou entrez une autre valeur.

Résultats

Chaque script d'installation copie ou lie les fichiers de configuration requis dans le répertoire de démarrage du système. Certains systèmes (Solaris et HP-UX, par exemple) offrent également la possibilité d'arrêter les processus à l'arrêt du système.

Que faire ensuite

Pour plus d'informations relatives à la modification des scripts de démarrage, consultez la documentation de votre système d'exploitation.

Démarrage automatique des agents de processus sous Windows

Sous Windows, vous pouvez installer l'agent de processus en tant que service Windows.

Pourquoi et quand exécuter cette tâche

Utilisez la fenêtre Services du Panneau de configuration pour attribuer l'un des comptes d'ouverture de session suivants au service :

- Compte système local (LocalSystem).
- Compte appartenant au groupe d'administrateurs de l'ordinateur local.

Pour plus d'informations relatives à l'installation et la configuration d'un agent de processus en tant que service Windows, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Gestion de la configuration du système de contrôle de processus

A l'issue de la configuration du système de contrôle de processus et de l'exécution des agents de processus, vous pouvez apporter des modifications à la configuration en exécutant les utilitaires de contrôle de processus. Toutes les modifications de configuration ne s'appliquent pas uniquement à la session en cours et ne sont pas sauvegardées dans le fichier de configuration.

Pourquoi et quand exécuter cette tâche

Vous pouvez également utiliser Netcool/OMNIbus Administrator pour modifier la configuration. Toutes les modifications apportées à partir de Netcool/OMNIbus Administrator peuvent être sauvegardées dans le fichier de configuration.

Concepts associés:

«Utilisation de Netcool/OMNIbus Administrator pour gérer le contrôle de processus», à la page 289
Netcool/OMNIbus Administrator possède une interface visuelle à partir de laquelle vous pouvez gérer le contrôle de processus. Vous pouvez utiliser Netcool/OMNIbus Administrator pour visualiser et gérer les agents de processus, les processus et les services sur vos hôtes Tivoli Netcool/OMNIbus.

Tâches associées:

«Configuration et gestion du contrôle de processus à partir de la ligne de commande», à la page 269
Vous pouvez définir des processus, services et hôtes dans le fichier de configuration du contrôle de processus. Vous pouvez également utiliser les utilitaires de ligne de commande pour démarrer, arrêter et ajouter un service ou processus, afficher l'état des services et processus, et arrêter un agent de processus.

Configuration et gestion du contrôle de processus à partir de la ligne de commande

Vous pouvez définir des processus, services et hôtes dans le fichier de configuration du contrôle de processus. Vous pouvez également utiliser les utilitaires de ligne de commande pour démarrer, arrêter et ajouter un service ou processus, afficher l'état des services et processus, et arrêter un agent de processus.

Pourquoi et quand exécuter cette tâche

Avant de pouvoir gérer le contrôle de processus à l'aide de l'unes de ces fonctions, vous devez créer la configuration système de contrôle de processus.

Concepts associés:

«Utilisation de Netcool/OMNIbus Administrator pour gérer le contrôle de processus», à la page 289
Netcool/OMNIbus Administrator possède une interface visuelle à partir de laquelle vous pouvez gérer le contrôle de processus. Vous pouvez utiliser Netcool/OMNIbus Administrator pour visualiser et gérer les agents de processus, les processus et les services sur vos hôtes Tivoli Netcool/OMNIbus.

Tâches associées:

«Création et démarrage d'un système réseau de contrôle de processus», à la page 252
Pour gérer le contrôle de processus, vous devez en premier lieu déterminer vos exigences en matière de configuration du contrôle de processus, puis réaliser un certain nombre de tâches de configuration.

Définition de processus, de services et d'hôtes pour le contrôle de processus

Pour exécuter le contrôle de processus, les processus, services et hôtes doivent être définis dans un fichier de configuration d'agent de processus. Au démarrage de l'agent de processus, ce dernier lit ce fichier afin d'établir les paramètres de configuration.

Pourquoi et quand exécuter cette tâche

Le fichier de configuration d'agent de processus est

```
$NCHOME/omnibus/etc/nco_pa.conf
```

Ce fichier est composé de définitions, chacun d'elles contenant les attributs et valeurs associées pour chaque processus, service et hôte. Les définitions figurent dans le fichier dans l'ordre suivant :

1. Définitions de processus
2. Définitions de service
3. Définitions de sécurité (facultatif)
4. Définitions de routage

Editez directement ce fichier pour ajouter ou modifier des définitions. Conservez les fichiers de configuration sur tous les hôtes afin de synchroniser les informations de configuration de l'hôte entre tous les agents de processus de la configuration.

Remarque : Pour interdire l'accès aux utilisateurs non autorisés, la sécurité du système d'exploitation doit être correctement configurée pour les fichiers

susceptibles de contenir les noms d'utilisateur et mots de passe (les fichiers de configuration, par exemple).

Définition des processus dans le fichier de configuration de l'agent de processus

Le fichier de configuration de l'agent de processus permet de définir la liste des processus à exécuter par les agents de processus.

Exemple de définition de processus

Une définition de processus du fichier de configuration `$NCHOME/omnibus/etc/nco_pa.conf` se présente comme suit :

```
nco_process 'ObjectServer'
{
Command '$NCHOME/omnibus/bin/nco_objserv -name NCOMS -pa SFOSYS1_PA' run as 0
Host   =  'sfosys1'
Managed   =  True
RestartMsg  =  '${NAME} running as ${EUID} has been restored on ${HOST}.'
AlertMsg  =  '${NAME} running as ${EUID} has died on ${HOST}.'
RetryCount  =  0
ProcessType  =  PaPA_AWARE
}
```

Description de la définition du processus

La table suivante utilise l'exemple précédent pour décrire les informations de définition du processus contenues dans le fichier de configuration.

Tableau 80. Description de la définition du processus

Informations sur la configuration	Description
`nco_process 'ObjectServer'`	Définit le nom du processus. Cet exemple s'adresse à un serveur ObjectServer. **Remarque :** Les noms de processus doivent être uniques pour cet agent de processus. Si vous utilisez le même nom de processus plus d'une fois, toutes les définitions de processus, à l'exception de la première, sont ignorées et un message d'avertissement est généré.

Tableau 80. Description de la définition du processus (suite)

Informations sur la configuration	Description

Command	Chaîne de commande qui démarre le processus, saisie dans la ligne de commande. Utilisez le chemin complet pour la commande. Par exemple, pour configurer un serveur ObjectServer nommé NCOMS, entrez la commande suivante :
	
	`'$NCHOME/omnibus/bin/nco_objserv -name NCOMS -pa SFOSYS1_PA' run as 0`
	
	Ou entrez la commande suivante :
	
	`'$NCHOME/omnibus/bin/nco_objserv -name NCOMS -pa SFOSYS1_PA' run as 'root'`
	
	Dans cet exemple :
	• L'option `-pa` spécifie l'agent de processus que le serveur ObjectServer utilise pour exécuter des automatisations externes. Dans cet exemple, le nom d'agent de processus spécifié est SFOSYS1_PA.
	• L'option `run as` demande à l'ordinateur hôte d'exécuter le serveur ObjectServer comme utilisateur spécifié. Sous UNIX, vous pouvez soit entrer l'ID utilisateur (généralement `0`), soit entrer le nom d'utilisateur entre guillemets simples (généralement `root`). Lorsqu'un nom d'utilisateur est saisi, l'agent de processus recherche l'ID utilisateur à employer. Si l'agent de processus n'est pas exécuté en tant que `root`, l'option `run as` est ignorée et le processus est exécuté comme l'utilisateur qui exécute l'agent de processus.
	Sous Windows, tous les processus sont exécutés sous le même compte utilisateur que l'agent de processus ; définissez toujours l'option `run as` sur `0`.
	Conseil : Sous Windows, vous pouvez utiliser %NCHOME%, $NCHOME ou la forme étendue de la variable d'environnement dans le chemin de la commande. Il est également possible d'utiliser les barres obliques (`/`) , les barres obliques inverses (`\`)ou les barres obliques inverses doubles (`\\`) comme séparateurs.
	
	Vous pouvez définir les attributs de processus supplémentaire suivants en les ajoutant en début de chaîne de commande :
	• `CWD` : définissez le répertoire de travail en cours sur la valeur spécifiée.
	Sous Windows, vous pouvez spécifier le répertoire dans l'un des formats suivants : MS-DOS (par exemple, `C:\temp`), UNIX (par exemple, `/temp/`*`mon_répertoire`*), et UNC (par exemple, `\\`*`serveur`*`\`*`partage`*`\`*`mon_répertoire`*). Les barres obliques inverses simples et doubles peuvent être utilisées comme séparateurs.
	Lorsque vous exécutez l'agent de processus à partir de la ligne de commande sous UNIX et Windows, le répertoire de base de tous les processus enfant est le répertoire dans lequel l'agent de processus a été démarré. Lorsque vous exécutez l'agent de processus en tant que démon UNIX, le répertoire de base de tous les processus enfant est `$NCHOME/omnibus`. Lorsque vous exécutez l'agent de processus en tant que service Windows, le répertoire de base par défaut pour l'agent de processus et tous les processus enfant générés par l'agent de processus est `%NCHOME%\omnibus\log`.
	• `SETGID` : définissez la valeur spécifiée pour l'ID groupe du processus. Il s'agit d'un attribut spécifique à UNIX.
	• `UMASK` : définissez la valeur spécifiée pour le umask. Il s'agit d'un attribut spécifiques à UNIX.
	
	Le format de spécification de chacun de ces attributs se présente comme suit :
	
	`Command '[CWD=`*`chemin_répertoire`*`]`*`chemin_commande options`*`' run as `*`utilisateur`*
	
	`Command '[SETGID=`*`ID_groupe`*`]`*`chemin_commande options`*`' run as `*`utilisateur`*
	
	`Command '[UMASK=`*`autorisation`*`]`*`chemin_commande options`*`' run as `*`utilisateur`*
	
	Remarque : Vous devez spécifier les attributs comme suit, entre crochets et sans espaces.

Tableau 80. Description de la définition du processus (suite)

Informations sur la configuration	Description

Command	*(suite de la page précédente)*
	
	Exemples (UNIX) :
	
	`Command '[CWD=/opt/netcool/]$NCHOME/omnibus/bin/nco_objserv -name NCOMS2 -pa NCO_PA' run as 1253`
	
	`Command '[SETGID=ncoadmin]$NCHOME/omnibus/bin/nco_objserv -name NCOMS2 -pa NCO_PA' run as 1253`
	
	`Command '[UMASK=u=rwx,g=rx,o=rx]$NCHOME/omnibus/bin/nco_objserv -name NCOMS2 -pa NCO_PA' run as 1253`
	Conseil : Dans l'exemple précédent avec le paramètre UMASK, les droits en écriture sont affectés à l'utilisateur en cours, mais supprimés pour tous les autres utilisateurs. Vous pouvez sinon spécifier cela sous la forme suivante : `[UMASK=022]`.
	
	`Command '[UMASK=077]$NCHOME/omnibus/bin/nco_objserv -name NCOMS2 -pa NCO_PA' run as 1253`
	
	Vous pouvez spécifier un ou plusieurs attributs dans la chaîne de commande. Par exemple :
	
	`Command '[CWD=/tmp][SETGID=ncoadmin][UMASK=u=rwx,g=,o=]$NCHOME/omnibus/bin/nco_objserv -name NCOMS2 -pa NCO_PA' run as 1253`
	
	Exemple (Windows) :
	
	`Command '[CWD=C:\temp]%NCHOME%\omnibus\bin\nco_objserv -name NCOMS2 -pa NCO_PA' run as 0`

Host	Nom de l'hôte sur lequel le processus doit être exécuté. Le contrôle de processus résout automatiquement le nom de l'agent de processus lorsque cela est nécessaire.
Managed	Peut posséder les valeurs suivantes : • `True` : le processus redémarre automatiquement s'il est interrompu. • `False` : le processus n'est pas surveillé par l'agent de processus ou ne redémarre pas automatiquement s'il est interrompu. Le statut d'un processus non géré peut être signalé de façon incorrecte par **nco_pa_status** ou **nco_config**. Par exemple, le statut renvoyé peut être `DEAD` lorsque le processus s'exécute correctement. Cela n'est pas une erreur et se produit car l'agent de processus démarre les processus non gérés mais ne les surveille pas.
RestartMsg	Contient le message à envoyer au système de consignation UNIX ou au visualiseur d'événements Windows si le processus redémarre. Par exemple, `The ObjectServer has been restarted.`
AlertMsg	Contient le message à envoyer au système de consignation UNIX ou au visualiseur d'événements Windows si le processus est interrompu. Par exemple, `The ObjectServer has gone down.`
RetryCount	Spécifie le nombre de tentatives de redémarrage à effectuer si le processus est interrompu dans le temps spécifié par l'option de ligne de commande **nco_pad** `-retrytime`. Si l'option est définie sur 0, aucune limite de tentatives de redémarrage n'est définie. La valeur par défaut est `0`.
ProcessType	Peut posséder la valeur `PaPA_AWARE` pour les processus compatibles avec l'agent de processus et la valeur `PaNOT_PA_AWARE` pour les processus non compatibles avec l'agent de processus.

Mots clés d'extension

Vous pouvez inclure des mots clés d'extension dans les entrées `RestartMsg` et`AlertMsg` du fichier de configuration. Les mots clés d'extension agissent comme variables et contiennent des informations sur le processus qui a redémarré.

Les mots clés d'extension sont présentés dans la table suivante.

Tableau 81. Mots clés d'extension

Mot clé d'extension	Description
${NAME}	Nom du processus.
${HOST}	Nom de l'hôte exécutant le processus.
${EUID}	ID utilisateur effectif sous lequel le processus est en cours d'exécution.
${COMMAND}	Commande définissant le processus.

Messages d'alerte et de redémarrage du système de consignation et du visualiseur d'événements

Lorsqu'un message d'alerte ou de redémarrage est généré par le démon d'agent de processus nco_pad, il est transmis au système de consignation UNIX ou au visualiseur d'événements Windows. Tivoli Netcool/OMNIbus possède une sonde de système de consignation capable de surveiller ces message et de les convertir en alertes du serveur ObjectServer. Pour plus d'informations sur la sonde du système de consignation, voir la documentation de la sonde disponible dans le Centre de documentation Tivoli Network Management à l'adresse suivante :http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp.

Les messages d'alerte et de redémarrage sont envoyés au système de consignation UNIX ou au visualiseur d'événements Windows en tant qu'avertissements. Le format du message est le suivant :

```
NOM D'HOTE : MSG_D'ALERTE_OU_DE_REDEMARRAGE : MSG
```

NOM D'HOTE est le nom de l'hôte qui a rapporté le problème. *MSG_D'ALERTE_OU_DE_REDEMARRAGE* décrit le type de message. *MSG* est le texte qui est défini dans le fichier de configuration pour ce processus.

Concepts associés:

«Résolution du nom d'hôte au démarrage», à la page 249
Un hôte doit être spécifié pour chaque automatisation externe ou processus sous le contrôle d'un agent de processus. L'hôte est défini dans le fichier de configuration de l'agent de processus, dans le cadre de la définition d'un processus, ou bien dans l'automatisation externe du serveur ObjectServer.

Définition des services dans le fichier de configuration d'agent de processus

Le fichier de configuration d'agent de processus permet de définir les services pour regrouper des processus liés et de configurer les interdépendances des processus. Les processus doivent déjà être définis dans la liste des processus du fichier.

Exemple de définition de service

Une définition de service du fichier de configuration `$NCHOME/omnibus/etc/nco_pa.conf` se présente comme suit :

```
nco_service 'Omnibus'
{
 ServiceType  =  Master
 ServiceStart  =  Non-Auto
 process 'ObjectServer' NONE
 process 'Proxy' 'ObjectServer'
 process 'Probe' 'Proxy'
 process 'Probe-1' 'ObjectServer'
 process 'Sleep' 5
}
```

Description de la définition du service

La table suivante utilise l'exemple précédent pour décrire les informations de définition de service contenues dans le fichier de configuration.

Tableau 82. Description de la définition de service

Informations sur la configuration	Description
`nco_service 'Omnibus'`	Définit le nom du service (par exemple, `Omnibus`). **Remarque :** Chaque nom de service doit être unique dans le réseau du contrôle de processus.
`ServiceType`	Définit si ce service doit redémarrer avant tous les autres services ou être géré en tant que service maître duquel dépendent tous les autres services. La valeur peut être soit `Master`, soit `Non-Master`.
`ServiceStart`	La valeur est soit `Auto` pour démarrer le service au démarrage de la commande **nco_pad**, soit `Non-Auto` si le service doit démarrer manuellement avec la commande **nco_pa_start**.
`process`	Chaque entrée de processus définit un processus qui doit être exécuté dans le cadre du service. Vous pouvez indiquer les dépendances de processus afin qu'un processus ne puisse démarrer tant qu'un autre processus est en cours d'exécution. **Remarque :** Vous devez inclure un processus une seule fois dans chaque définition de service du fichier de configuration.

Spécification des dépendances de processus

Lors de la définition d'un service, vous pouvez utiliser l'attribut `process` afin de définir les processus qui doivent être exécutés dans le cadre du service. Vous pouvez ajouter des dépendances sur chaque processus du service. Le format de l'attribut `process` est le suivant :

`process '`*`nom_processus`*`' `*`dépendance`*

Dans cet attribut, *nom_processus* est le nom du processus défini dans la liste des processus et *dépendance* peut être une valeur numérique, une valeur de chaîne ou `NONE`.

Si *dépendance* est un nombre, il indique une dépendance de temps en secondes pour démarrer le processus dépendance. Une dépendance de temps est toujours mesurée à partir du début du service. Par exemple, si vous entrez `5`, le processus début cinq secondes après le début du service.

Si *dépendance* est une chaîne, il indique un autre processus PA aware dans le même service.

Restriction : Un processus ne peut pas être dépendant d'un autre processus dépendant du temps. Si vous spécifiez une dépendance sur un processus dépendant du temps, un message d'erreur est ajouté au fichier journal de contrôle de processus et le processus dépendant et les processus enfants se voient affecter le statut DEAD. Le fichier journal par défaut est `$NCHOME/omnibus/log/`*`nom_ap`*`.log`, où *nom_ap* est le nom de l'agent de processus.

Le type de *dépendance* `NONE` ne spécifie aucune dépendance.

Dans l'exemple de définition de service précédent du service `Omnibus`, le processus `ObjectServer` démarre en premier car il ne possède pas de dépendances. Cinq secondes après le démarrage du serveur ObjectServer, le processus `Sleep` démarre. Lorsque le serveur ObjectServer s'exécute correctement, `Proxy` et `Probe-1` démarrent. Lorsque le serveur proxy est en cours d'exécution, le processus `Probe` démarre. Si l'un des processus est dépendant du processus `Sleep` dépendant du temps, ce processus ne démarre pas et se voit affecter le statut DEAD.

Référence associée:

«Définition des processus dans le fichier de configuration de l'agent de processus», à la page 270
Le fichier de configuration de l'agent de processus permet de définir la liste des processus à exécuter par les agents de processus.

Définition des hôtes sécurisés dans le fichier de configuration d'agent de processus

Vous pouvez définir que seuls certains hôtes peuvent se connecter aux agents de processus en ajoutant une définition de sécurité au fichier de configuration d'agent de processus. Si vous ne créez pas de définition de sécurité, n'importe quel processus peut se connecter à n'importe quel hôte.

Dans le fichier de configuration `$NCHOME/omnibus/etc/nco_pa.conf`, la définition de la sécurité est insérée entre les définitions de service et les définitions de routage des hôtes. Vous pouvez créer une définition de sécurité sans hôte spécifié, comme suit :

```
nco_security
{
}
```

Lorsqu'aucun hôte n'est spécifié, seuls les processus en cours d'exécution sur l'hôte actuel ou sur tout hôte répertorié dans la définition de routage peuvent se connecter.

Les processus exécutés sur les hôtes et qui ne sont pas répertoriés dans la définition de routage ne peuvent se connecter que si leur hôte est répertorié dans la définition de la sécurité.

L'agent de processus compare l'adresse IP de la connexion entrante et l'adresse IP de chaque entrée des définitions de sécurité et de routage. L'agent de processus vérifie également l'adresse IP de l'hôte local. Seule l'adresse principale de l'hôte exécutant le démon d'agent de processus est automatiquement ajoutée à la définition de la sécurité. Vous devez ajouter l'adresse de bouclage (127.0.0.1) et les interfaces secondaires, si nécessaire.

Remarque : Lorsqu'un processus se connectant à l'agent de processus est exécuté sur un hôte avec plusieurs interfaces, vous devez ajouter l'adresse de l'interface la plus proche du démon d'agent de processus. Il n'est pas nécessaire que ce soit l'adresse principal de cet hôte. En outre, dans le cas du serveur ObjectServer (**nco_objserv**) ou du démon d'agent de processus (**nco_pad**), il n'est pas nécessaire que ce soit l'adresse de l'éditeur de serveur (**nco_xigen**).

Vous pouvez spécifier les types d'entrée suivants dans la définition de la sécurité :

- Un nom d'hôte. Dans ce cas, une recherche est effectuée pour rechercher l'adresse IP correspondante
- Une adresse IPv4 complète en notation décimale avec point
- Une adresse IPv6 en notation complète, abrégée ou mixte, avec des valeurs séparées par des deux-points

Une adresse IPv4 en notation décimale à point peut contenir les caractères génériques suivants :

- `?` correspond à un caractère
- `*` correspond à plusieurs caractères

Vous pouvez ajouter */n* à une adresse IPv6 spécifiée, où *n* est un nombre, pour représenter les adresses IPv6 dans lesquelles les *n* premiers bits correspondent à l'adresse IP établie.

Exemple de définition de sécurité

L'exemple de définition de sécurité suivant autorise les connexions à partir des processus sur les hôtes suivants :

- alpha
- 192.9.200.34
- Hôte sur le sous-réseau 193.42.52.0
- Hôte avec une adresse IP où les 10 premiers bits correspondent à fe80::203:baff:fe2a:6bf0
- fe80::203:baff:fe2a:6bf0

```
nco_security
{
 host 'alpha'
 host '192.9.200.34'
 host '193.42.52.*'
 host 'fe80::203:baff:fe2a:6bf0/10'
 host 'fe80::203:baff:fe2a:6bf0'
}
```

Définition des processus dans le fichier de configuration de l'agent de processus

Pour spécifier les hôtes qui participent au système de contrôle du processus, vous devez définir les noms d'hôte d'agent de processus dans le fichier de configuration d'agent de processus.

Chaque entrée d'hôte définit le nom de l'hôte (par exemple, sfosys1) et le nom de l'agent de processus à utiliser dans le système de contrôle de processus (par exemple, SFOSYS1_PA). Pour chaque définition d'hôte, vous pouvez également spécifier les données d'identification de nom d'utilisateur et de mot de passe de connexion à l'agent de processus.

Exemple de définition de routage

Une définition de routage du fichier de configuration `$NCHOME/omnibus/etc/nco_pa.conf` se présente comme suit :

```
nco_routing
{
host 'sfosys1' 'SFOSYS1_PA' 'nom_utilisateur' 'mot_de_passe'
host 'sfosys2' 'SFOSYS2_PA' 'nom_utilisateur' 'mot_de_passe'
}
```

Remarque : Les entrées *nom_utilisateur* et *mot_de_passe* sont obligatoires si vous exécutez l'agent de processus distant en mode sécurisé. Si vous n'exécutez pas l'agent de processus distant en mode sécurisé, les noms d'utilisateur et les mots de passe sont facultatifs.

Si l'agent de processus utilise l'authentification UNIX (par défaut sous UNIX), *nom_utilisateur* doit être un utilisateur du système d'exploitation qui est membre du groupe ncoadmin ou de tout autre groupe administratif créé pour autoriser l'accès au système de contrôle de processus. Un démon d'agent de processus en cours d'exécution en mode sécurisé doit être exécuté par l'utilisateur root.

Sous Windows, *nom_utilisateur* doit être le nom d'utilisateur d'un compte local, compte de domaine ou compte UPN valide.

Remarque : Pour empêcher les utilisateurs non autorisés d'obtenir l'accès, la sécurité du système d'exploitation doit être définie correctement pour les fichiers qui contiennent des noms d'utilisateur et des mots de passe.

Lors de l'exécution du démon d'agent de processus **nco_pad**, vous pouvez également spécifier le nom d'utilisateur et le mot de passe à l'aide des options de ligne de commande `-user` et `-password`. Ces valeurs remplacent les entrées contenues dans le fichier de configuration `nco_pa.conf`.

Chiffrement des mots de passe en texte clair dans les définitions de routage

Vous pouvez coder les mots de passe de connexion en texte clair qui sont stockés dans le fichier `nco_pa.conf`.

Les informations de chiffrement du mot de passe relatives à l'exécution en modes FIPS 140–2 et non-FIPS 140–2 sont décrites dans la table suivante.

Tableau 83. Chiffrement des mots de passe en modes FIPS 140–2 et non-FIPS 140–2

Mode	Action
mode FIPS 140–2	Lorsque le mode FIPS 140–2 est activé, les mots de passe peuvent être spécifiés en texte normal ou dans un format chiffré. Vous pouvez coder les mots de passe à l'aide du chiffrement des valeurs de propriété de la façon suivante : 1. Si vous ne disposez pas encore d'une clé pour le chiffrement du mot de passe, créez-en une en exécutant l'utilitaire **nco_keygen** qui se trouve dans le répertoire `$NCHOME/omnibus/bin`. 2. Exécutez l'utilitaire **nco_aes_crypt** pour coder le mot de passe avec la clé générée par l'utilitaire **nco_keygen**. L'utilitaire **nco_aes_crypt** se trouve également dans le répertoire `$NCHOME/omnibus/bin`. Notez que vous devez spécifier `AES_FIPS` comme algorithme à utiliser pour le chiffrement du mot de passe. 3. Copiez le mot de passe codé dans la définition de routage appropriée du fichier de configuration.
Mode non-FIPS 140–2	En mode non-FIPS 140–2, vous pouvez soit exécuter l'utilitaire **nco_pa_crypt**, soit utiliser le chiffrement de la valeur de propriété pour coder les mots de passe de connexion en texte clair sous UNIX. Sous Windows, vous pouvez exécuter l'utilitaire **nco_g_crypt**, soit utiliser le chiffrement de la valeur de propriété. Effectuez l'une des actions suivantes : • Pour coder un mot de passe à l'aide de l'utilitaire **nco_pa_crypt** ou **nco_g_crypt**, exécutez la commande suivante : – UNIX : `$NCHOME/omnibus/bin/nco_pa_crypt` *`mot_de_passe_texte_clair`* – Windows : `%NCHOME%\omnibus\bin\nco_g_crypt` *`mot_de_passe_texte_clair`* Dans ces commandes, *mot_de_passe_texte_clair* représente la forme non codée du mot de passe. L'utilitaire de chiffrement affiche une version codée du mot de passe. Copiez le mot de passe codé dans la définition de routage appropriée du fichier de configuration. • Pour coder un mot de passe à l'aide du chiffrement de la valeur de la propriété, vous avez besoin d'une clé générée à l'aide de l'utilitaire **nco_keygen**. Vous pouvez alors exécuter **nco_aes_crypt** pour coder le mot de passe avec la clé. Notez que vous pouvez spécifier `AES_FIPS` ou `AES` comme algorithme de chiffrement du mot de passe. 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. Copiez le mot de passe codé dans la définition de routage appropriée du fichier de configuration.

Remarque : Sous UNIX, même si le mot de passe est spécifié dans la ligne de commande, il n'apparaît pas dans la sortie de la commande **ps**.

Les mots de passe qui sont codés à l'aide de la commande **nco_pa_crypt** sont décodés par l'agent de contrôle de processus distant.

Les mots de passe qui sont codés à l'aide de **nco_aes_crypt** sont décodés à l'aide du démon d'agent de processus distant et sont transmis aux agents de processus distant en texte clair. Pour décoder les mots de passe, vous devez définir les

options de ligne de commande `-cryptalgorithm` et `-keyfile` lors de l'exécution de **nco_pad**. Ces options spécifient l'algorithme et le fichier clé à utiliser dans le cadre du déchiffrement.

Pour plus d'informations sur l'utilisation du chiffrement de la valeur de propriété, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Tâches associées:

«Création de groupes d'utilisateurs UNIX pour le système de contrôle de processus», à la page 253
Le démon de contrôle de processus permet de contrôler les personnes qui s'y connectent. Sous UNIX, tous les utilisateurs qui souhaitent accéder au système de contrôle de processus doivent appartenir à groupe d'utilisateurs UNIX que vous identifiez comme un groupe d'administration à cette fin.

Référence associée:

«Propriétés des agents de processus et options de ligne de commande», à la page 257
Lors de l'exécution de l'agent de processus à l'aide de la commande **nco_pad**, vous pouvez spécifier des options de ligne de commande supplémentaires pour affiner la configuration.

Exemple : fichier de configuration d'agent de processus

Cet exemple présente le contenu d'un fichier de configuration d'agent de processus `$NCHOME/omnibus/etc/nco_pa.conf`.

```
###############################################################################
#NCO_PA3#
# Fichier de configuration de démon d'agent de processus version 1.1
#
# Ident: $Id: nco_pa.conf 1.3 2002/05/21 15:28:10
#
#
# Liste des processus
#
nco_process 'NO1_PROXY_ProxyServer'
{
 Command '$NCHOME/omnibus/bin/nco_proxyserv -name NETPROXY -server NETOPS1' run as 0
 Host  = 'objser1'
 Managed  = True
 RestartMsg = '${NAME} running as ${EUID} has been restored on ${HOST}.'
 AlertMsg = '${NAME} running as ${EUID} has died on ${HOST}.'
 RetryCount = 0
 ProcessType = PaPA_AWARE
}
nco_process 'SFOSYS_ObjectServer'
{
 Command '$NCHOME/omnibus/bin/nco_objserv -name NETOPS1 -pa OBJSER1_PA' run as 0
 Host  = 'objser1'
 Managed  = True
 RestartMsg = '${NAME} running as ${EUID} has been restored on ${HOST}.'
 AlertMsg = '${NAME} running as ${EUID} has died on ${HOST}.'
 RetryCount = 0
 ProcessType = PaPA_AWARE
}
nco_process 'Syslog_Probe'
{
 Command '$NCHOME/omnibus/probes/nco_p_syslog' run as 0
 Host  = 'objser1'
 Managed  = True
 RestartMsg = '${NAME} running as ${EUID} has been restored on ${HOST}.'
 AlertMsg = '${NAME} running as ${EUID} has died on ${HOST}.'
 RetryCount = 0
 ProcessType = PaNOT_PA_AWARE
}
nco_process 'Mttrapd_Probe'
{
 Command '$NCHOME/omnibus/probes/nco_p_mttrapd' run as 0
 Host  = 'objser1'
 Managed  = True
 RestartMsg = '${NAME} running as ${EUID} has been restored on ${HOST}.'
```

```
 AlertMsg = '${NAME} running as ${EUID} has died on ${HOST}.'
 RetryCount = 0
 ProcessType = PaNOT_PA_AWARE
}
nco_process 'MyScript'
{
 Command '$HOME/myscript.sh' run as 0
 Host  = 'objser1'
 Managed  = False
 RestartMsg = '${NAME} running as ${EUID} has been restored on ${HOST}.'
 AlertMsg = '${NAME} running as ${EUID} has died on ${HOST}.'
 RetryCount = 0
 ProcessType = PaNOT_PA_AWARE
}
#
# Liste des services
#
nco_service 'Core'
{
 ServiceType = Master
 ServiceStart = Auto
 process 'MyScript' NONE
# Le serveur ObjectServer a démarré au bout de 20 secondes pour permettre au script de s'arrêter
 process 'SFOSYS_ObjectServer' 20
# Le serveur proxy a démarré après le démarrage du serveur ObjectServer
 process 'NO1_PROXY_ProxyServer' 'SFOSYS_ObjectServer'
# La sonde Trapd et la sonde Syslog a démarré après le démarrage du serveur proxy
 process 'Mttrapd_Probe' 'NO1_PROXY_ProxyServer'
 process 'Syslog_Probe' 'NO1_PROXY_ProxyServer'
}
#
# TABLE DE ROUTAGE
#
# 'user' - (facultatif) uniquement requis pour l'assembleur désassembleur de paquets
en mode sécurisé sur un hôte cible
# 'user' doit être le membre du groupe d'administration UNIX si vous utilisez
l'authentification UNIX
# Sous Windows, 'user' doit correspondre à un compte local, de domaine ou UPN valide
# 'password' - (facultatif) uniquement obligatoire pour l'assembleur
désassembleur de paquets en mode sécurisé sur l'hôte cible
# peut être en texte clair ou chiffré à l'aide de l'outil de chiffrement fourni
# pour les exigences spécifiques en matière de plateforme et de sécurité
nco_routing
{
 host 'objser1' 'OBJSER1_PA'
}
```

Gestion du contrôle de processus à l'aide des utilitaires de contrôle de processus

Le système de contrôle de processus possède des utilitaires de ligne de commande pour gérer et modifier la configuration de Tivoli Netcool/OMNIbus. Vous pouvez exécuter ces utilitaires pour démarrer, arrêter et ajouter un service ou un processus, afficher le statut des services et des processus et interrompre un agent de processus.

Les utilitaires de ligne de commande sont les suivants :

- **nco_pa_status**
- **nco_pa_start**
- **nco_pa_stop**
- **nco_pa_shutdown**
- **nco_pa_addentry**

Chaque utilitaire vous demande un mot de passe.

Affichage du statut des services et des processus (nco_pa_status)

Vous pouvez exécuter l'utilitaire **nco_pa_status** pour extraire le statut des services dans la configuration du système de contrôle de processus. Pour chaque service, l'utilitaire **nco_pa_status** retourne une liste de processus définis, le statut de chaque processus, et l'identificateur de processus.

Pour afficher le statut de service, entrez la commande suivante :

```
$NCHOME/omnibus/bin/nco_pa_status -server chaîne
```

Dans cette commande, *chaîne* est le nom de l'agent de processus. Vous pouvez, si vous le souhaitez, exécuter la commande avec des options de ligne de commande supplémentaires.

Voici un exemple de sortie :

```
------------------------------------------------------------------------
Service Name Process Name Hostname User Status PID
------------------------------------------------------------------------
Master Service ObjectServer SFOSYS1  root RUNNING 16751
   Proxy SFOSYS1  root RUNNING 16752
   Sleep SFOSYS1  root RUNNING 16753
   Probe SFOSYS1  root RUNNING 16754
------------------------------------------------------------------------
```

La valeur `PID` des processus gérés est l'identificateur de processus UNIX ou le PID affiché dans le gestionnaire de tâches Windows.

La table suivante décrit chaque niveau de statut.

Tableau 84. Descriptions du statut du service

Niveau du statut	Description
RUNNING	Le processus est en cours d'exécution.
STARTING	Une requête de démarrage a été transmise.
PENDING	Le processus attend une dépendance de temps pour se terminer. Ce statut peut également indiquer que le processus n'a pas démarré correctement (quelles que soient les dépendances de processus).
WAITING	Le processus attend une dépendance pour démarrer.
DEAD	Le processus n'est pas en cours d'exécution.
ERROR	Il n'a pas été possible d'extraire un statut de l'agent de processus.

S'il est demandé à un agent de processus d'exécuter un processus par un agent de processus exécuté sur un autre poste, l'agent de processus distant ne conserve pas un enregistrement du processus. Si l'agent de processus distant s'arrête, le processus poursuit son exécution. Lorsque l'agent de processus distant redémarre, il ne possède pas d'enregistrement du processus. Par conséquent, le statut de ce processus enfant possède le statut DEAD. Vous pouvez redémarrer manuellement le processus via l'utilitaire **nco_pa_start**.

Si un processus est défini comme non géré (`Managed = False`) dans le fichier de configuration d'agent de processus (`nco_pa.conf`), **nco_pa_status** peut renvoyer un

état aléatoire lorsqu'il vérifie le processus. Par exemple, le statut renvoyé peut être DEAD lorsque le processus s'exécute correctement. Cela n'est pas une erreur et se produit car l'agent de processus démarre les processus non gérés mais ne les surveille pas.

Options de ligne de commande de nco_pa_status

Les options de ligne de commande de l'utilitaire **nco_pa_status** sont décrites dans la table suivante.

Tableau 85. Options de ligne de commande de nco_pa_status

Option de ligne de commande	Description
`-help`	Affiche l'aide sur les options de ligne de commande et quitte.
`-nosecure`	Se connecte aux agents de processus dans un mode non sécurisé qui ne chiffre pas les informations de connexion pendant la transmission.
`-password` *chaîne*	Mot de passe à utiliser pour l'agent de processus.
`-server` *chaîne*	Nom de l'agent de processus à contacter.
`-user` *chaîne*	Nom d'utilisateur de l'agent de processus. La valeur par défaut est l'utilisateur qui exécute la commande.
`-version`	Affiche les informations de version de logiciel et quitte.

Référence associée:

«Démarrage d'un service ou d'un processus (nco_pa_start)»
Vous pouvez exécuter l'utilitaire **nco_pa_start** pour démarrer un service ou un processus à n'importe quel emplacement de la configuration du système de contrôle de processus.

Démarrage d'un service ou d'un processus (nco_pa_start)

Vous pouvez exécuter l'utilitaire **nco_pa_start** pour démarrer un service ou un processus à n'importe quel emplacement de la configuration du système de contrôle de processus.

Si le service ou le processus a déjà été arrêté, la commande est ignorée.

Pour démarrer un service ou un processus, entrez la commande suivante :

```
$NCHOME/omnibus/bin/nco_pa_start options_ligne_commande
```

Dans cette commande, *options_ligne_commande* représente une ou plusieurs options de ligne de commande que vous spécifiez pour l'utilitaire **nco_pa_start**. Vous ne pouvez spécifier qu'un seul service/processus.

Options de ligne de commande pour nco_pa_start

Les options de ligne de commande de l'utilitaire **nco_pa_start** sont décrites dans la table suivante.

Tableau 86. Options de ligne de commande pour nco_pa_start

Option de ligne de commande	Description
`-help`	Affiche l'aide sur les options de ligne de commande et quitte.

Tableau 86. Options de ligne de commande pour nco_pa_start (suite)

Option de ligne de commande	Description
`-nosecure`	Se connecte aux agents de processus dans un mode non sécurisé qui ne chiffre pas les informations de connexion pendant la transmission.
`-password` *chaîne*	Mot de passe à utiliser pour l'agent de processus.
`-process` *chaîne*	Nom du processus à démarrer.
`-server` *chaîne*	Nom de l'agent de processus à contacter.
`-service` *chaîne*	Nom du service à démarrer.
`-user` *chaîne*	Nom d'utilisateur de l'agent de processus. L'utilisateur par défaut est celui qui exécute la commande.
`-version`	Affiche les informations de version de logiciel et quitte.

Arrêt d'un service ou d'un processus (nco_pa_stop)

Vous pouvez exécuter l'utilitaire **nco_pa_stop** pour arrêter un service ou un processus à n'importe quel emplacement de la configuration du système de contrôle de processus.

Si le service ou le processus a déjà été arrêté, la commande est ignorée.

Lorsque vous arrêtez un service, tous les processus définis dans ce service sont également arrêtés. Lorsque vous arrêtez un processus, l'état des processus dépendants reste inchangé. Par exemple, si un ObjectServer est arrêté, l'exécution des analyses dépendantes de ce serveur se poursuit.

Pour arrêter un service ou un processus, entrez la commande suivante :

```
$NCHOME/omnibus/bin/nco_pa_stop options_ligne_commande
```

Dans cette commande, *options_ligne_commande* représente une ou plusieurs options de ligne de commande que vous spécifiez pour l'utilitaire **nco_pa_stop**. Vous ne pouvez spécifier qu'un service/processus.

Options de ligne de commande pour nco_pa_stop

Les options de ligne de commande de l'utilitaire **nco_pa_stop** sont décrites dans la table suivante.

Tableau 87. Options de ligne de commande pour nco_pa_stop

Option de ligne de commande	Description
`-force`	Si vous spécifiez cette option, aucun avertissement n'est émis si le processus ou le service n'est pas en cours d'exécution.
`-help`	Affiche l'aide sur les options de ligne de commande et quitte.
`-nosecure`	Se connecte aux agents de processus dans un mode non sécurisé qui ne chiffre pas les informations de connexion pendant la transmission.
`-password` *chaîne*	Mot de passe à utiliser pour l'agent de processus.

Tableau 87. Options de ligne de commande pour nco_pa_stop (suite)

Option de ligne de commande	Description
`-process` *chaîne*	Nom du processus à arrêter.
`-server` *chaîne*	Nom de l'agent de processus à contacter.
`-service` *chaîne*	Nom du service à arrêter.
`-user` *chaîne*	Nom d'utilisateur de l'agent de processus. L'utilisateur par défaut est celui qui exécute la commande.
`-version`	Affiche les informations de version de logiciel et quitte.

Fermeture d'un agent de processus (nco_pa_shutdown)

Vous pouvez exécuter l'utilitaire **nco_pa_shutdown** pour arrêter un agent de processus et éventuellement arrêter les services et processus associés.

Pour arrêter un agent de processus, entrez la commande suivante :

```
$NCHOME/omnibus/bin/nco_pa_shutdown options_ligne_commande
```

Dans cette commande, *options_ligne_commande* représente une ou plusieurs options de ligne de commande que vous pouvez spécifier pour l'utilitaire **nco_pa_shutdown**.

Options de ligne de commande pour nco_pa_shutdown

Les options de ligne de commande de l'utilitaire **nco_pa_shutdown** sont décrites dans la table suivante.

Tableau 88. Options de ligne de commande pour nco_pa_shutdown

Option de ligne de commande	Description
`-help`	Affiche l'aide sur les options de ligne de commande et quitte.
`-nosecure`	Se connecte aux agents de processus dans un mode non sécurisé qui ne chiffre pas les informations de connexion pendant la transmission.
`-option` *chaîne*	indique le mode d'arrêt. Il peut s'agir de `STOP`, qui arrête tous les processus que l'agent de processus gère localement, ou de `LEAVE`, qui maintient l'exécution des processus gérés localement après l'arrêt. Si l'option `-option` n'est pas spécifiée dans la ligne de commande, l'utilitaire affiche un menu avec les options de fermeture et vous invite à définir le type d'arrêt à effectuer. **Conseil :** Pour arrêter un processus géré localement, vous devez exécuter l'utilitaire **nco_pa_stop**.
`-password` *chaîne*	Mot de passe à utiliser pour l'agent de processus.
`-server` *chaîne*	Nom de l'agent de processus à arrêter.
`-user` *chaîne*	Nom d'utilisateur de l'agent de processus. L'utilisateur par défaut est celui qui exécute la commande.
`-version`	Affiche les informations de version de logiciel et quitte.

Conseil : Si vous exécutez un agent de processus en tant que service Windows et que vous arrêtez le service, les processus gérés sont également arrêtés. Cela signifie que tous les processus gérés sont arrêtés de manière contrôlée lorsque le système est arrêté.

Référence associée:

«Arrêt d'un service ou d'un processus (nco_pa_stop)», à la page 283
Vous pouvez exécuter l'utilitaire **nco_pa_stop** pour arrêter un service ou un processus à n'importe quel emplacement de la configuration du système de contrôle de processus.

Ajout de services ou de processus (nco_pa_addentry)

Vous pouvez exécuter l'utilitaire **nco_pa_addentry** pour ajouter des services ou des processus lors de l'exécution de l'agent de processus.

Exécutez cet utilitaire pour :

- Ajouter des services à un agent de processus en cours d'exécution.
- Démarrer un processus *autonome après son lancement*. Ces processus démarrent automatiquement et ne peuvent pas être modifiés. Lors de la configuration d'un processus autonome après son lancement, définissez le processus en tant que processus non géré si vous souhaitez vous assurer qu'il ne s'exécute qu'une seule fois. Pour ce faire, spécifiez l'option de ligne de commande `-unmanaged` lors de l'exécution de l'utilitaire **nco_pa_addentry**.
- Ajouter un processus géré à un service.

Remarque : Le nouveau service ou processus n'est pas ajouté au fichier de configuration de l'agent de processus, à moins que vous ne choisissiez de mettre à jour le fichier de configuration lors de l'utilisation de Netcool/OMNIbus Administrator.

Pour ajouter un service ou un processus, entrez la commande suivante. Les crochets indiquent les entrées facultatives.

```
$NCHOME/omnibus/bin/nco_pa_addentry [-process chaîne | -service chaîne]
options_ligne_commande
```

Dans cette commande, *options_ligne_commande* représente une ou plusieurs lignes de commande que vous spécifiez pour le service ou le processus, ou pour l'utilitaire **nco_pa_addentry**.

Options de ligne de commande de nco_pa_addentry

Les options de ligne de commande de l'utilitaire **nco_pa_addentry** sont décrites dans la table suivante.

Important : Pour s'assurer que le processus ou le service est correctement créé, toutes les options de ligne de commande pertinentes doivent être spécifiées de façon explicite avec une valeur *chaîne*. Ce critère s'applique également à tout paramètre par défaut que vous souhaiteriez appliquer.

Tableau 89. Options de ligne de commande de nco_pa_addentry

Option de ligne de commande	Description
`-alert_msg` *chaîne*	Spécifie le message à envoyer au système de consignation UNIX ou au visualiseur d'événements Windows si le processus est interrompu. Sous UNIX, intégrez la valeur *chaîne* entre guillemets simples si le texte contient des espaces. Sous Windows, intégrez la valeur *chaîne* entre guillemets doubles si le texte contient des espaces.

`-auto`
 	
`-nonauto`
	Si l'option `-auto` est spécifiée, le service ou le processus démarre en même temps que l'agent de processus. Par défaut, le service doit démarrer manuellement à l'aide de la commande **nco_pa_start**.

`-command` *chaîne*	Spécifie la ligne de commande du processus. Par exemple : `$NCHOME/omnibus/bin/nco_objserv -name NCOMS -pa SFOSYS1_PA`
`-delay` *chaîne*	Spécifie le délai d'attente en secondes avant le démarrage du processus spécifié.
`-depend` *chaîne*	Spécifie un processus duquel le processus spécifié dépend.
`-help`	Affiche l'aide sur les options de ligne de commande et quitte.
`-host` *chaîne*	Spécifie l'hôte sur lequel exécuter le processus.

`-managed`
 	
`-unmanaged`
	Si l'option `-managed` est spécifiée, le processus redémarre automatiquement s'il est interrompu. La valeur par défaut est `-managed`.

`-master`
 	
`-nonmaster`
	Si l'option `-master` est spécifiée, le type de service est défini sur master. La valeur par défaut est `-master`.

`-nosecure`	Se connecte aux agents de processus dans un mode non sécurisé qui ne chiffre pas les informations de connexion pendant la transmission.

`-pa_aware`
 	
`-not_pa_aware`
	Si `-pa_aware` est spécifié, `ProcessType` est défini sur `PaPA_AWARE`. Par défaut, le processus est défini sur PA aware.

`-parentservice` *chaîne*	Spécifie le service auquel ajouter le processus. **Remarque :** Lors de l'ajout d'un processus à un service, vous *devez* définir le service parent à l'aide de `-parentservice`.
`-password` *chaîne*	Définit le mot de passe à utiliser lors de la connexion à l'agent de processus.
`-process` *chaîne*	Spécifie le nom du processus à ajouter.
`-restart_msg` *chaîne*	Spécifie le message à envoyer au système de consignation UNIX ou au visualiseur d'événements Windows si le processus redémarre. Sous UNIX, intégrez la valeur *chaîne* entre guillemets simples si le texte contient des espaces. Sous Windows, intégrez la valeur *chaîne* entre guillemets doubles si le texte contient des espaces.

Tableau 89. Options de ligne de commande de nco_pa_addentry (suite)

Option de ligne de commande	Description
`-retrycount` *entier*	Spécifie le nombre de tentatives de redémarrage à effectuer si le processus est interrompu dans le temps spécifié par l'option de ligne de commande **nco_pad** `-retrytime`. Si l'option est définie sur `0`, aucune limite de tentatives de redémarrage n'est définie. La valeur par défaut est `0`.
`-runas` *entier*	Spécifie l'ID utilisateur sous lequel exécuter le processus.
`-server` *chaîne*	Spécifie le nom de l'agent de processus. La valeur par défaut est `NCO_PA`.
`-service` *chaîne*	Spécifie le nom du service à ajouter.
`-user` *chaîne*	Spécifie le nom d'utilisateur à utiliser lors de la connexion à l'agent de processus. La valeur par défaut est l'utilisateur qui exécute la commande.
`-version`	Affiche les informations de version de logiciel et quitte.

Exemple : utilisation de nco_pa_addentry pour ajouter un processus autonome après son lancement (UNIX) sur les systèmes d'exploitation 32 bits

1. Entrez la commande suivante pour ajouter un processus autonome après son lancement nommé simnet1, qui démarre automatiquement et ne s'exécute qu'une fois (comme processus non géré) :

   ```
   ./nco_pa_addentry -server TEST_PA -process 'simnet1' -command
   '$NCHOME/omnibus/probes/nco_p_simnet' -host 'owl' -retrycount 0
   -unmanaged -restart_msg 'test' -alert_msg 'testalert'
   ```

2. Exécutez l'utilitaire **nco_pa_status** pour extraire le statut des services dans la configuration :

   ```
   $NCHOME/omnibus/bin/nco_pa_status -server TEST_PA
   ```

   Où TEST_PA est le nom de l'agent de processus.

Lorsque l'utilitaire **nco_pa_status** est exécuté, la sortie n'affiche par de définition de processus `simnet1` dans une entrée de service. Toutefois, la commande **ps -ef** affiche le processus simnet1 comme étant en cours d'exécution, bien qu'il ne redémarre pas automatiquement s'il est interrompu.

Exemple : utilisation de nco_pa_addentry pour ajouter un processus autonome après son lancement (Windows)

1. Entrez la commande suivante pour ajouter un processus autonome après son lancement nommé simnet1, qui démarre automatiquement et ne s'exécute qu'une fois (comme processus non géré) :

   ```
   "%NCHOME%"\omnibus\bin\nco_pa_addentry -server TEST_PA -process simnet1
   -command "%NCHOME%"\omnibus\probes\win32\nco_p_simnet -host owl
   -retrycount 0 -unmanaged -restart_msg "Probe restarted" -alert_msg
   "Probe stopped" -password secret
   ```

2. Exécutez l'utilitaire **nco_pa_status** pour extraire le statut des services dans la configuration :

   ```
   "%NCHOME%"\omnibus\bin\nco_pa_status -server TEST_PA
   ```

   Où TEST_PA est le nom de l'agent de processus.

Lorsque l'utilitaire **nco_pa_status** est exécuté, la sortie n'affiche par de définition de processus simnet1 dans une entrée de service. Toutefois, le gestionnaire de tâches Windows affiche le processus simnet1 comme étant en cours d'exécution, bien qu'il ne redémarre pas automatiquement s'il est interrompu.

Exemple : utilisation de nco_pa_addentry pour ajouter un processus géré à un service (Solaris) sur les systèmes d'exploitation 64 bits

1. Entrez la commande suivante pour ajouter un processus nommé simnet2 à un service Core :

   ```
   ./nco_pa_addentry -server TEST_PA -process 'simnet2' -command
   '$NCHOME/omnibus/platform/solaris2/probes64/nco_p_simnet' -host 'owl'
   -retrycount 0 -managed -restart_msg 'test' -alert_msg 'testalert'
   -parentservice 'Core'
   ```

2. Exécutez l'utilitaire **nco_pa_status** pour extraire le statut des services dans la configuration :

   ```
   $NCHOME/omnibus/bin/nco_pa_status -server TEST_PA
   ```

   Où TEST_PA est le nom de l'agent de processus.

Lors de l'exécution de **nco_pa_status**, la sortie affiche un processus simnet2 doté d'un statut DEAD dans la définition de service Core. Le processus simnet2 ne démarre pas automatiquement car il fait partie d'un service. Pour exécuter le service, utilisez l'utilitaire **nco_pa_start**.

Exemple : utilisation de nco_pa_addentry pour ajouter un processus géré à un service (Windows)

1. Entrez la commande suivante pour ajouter un processus nommé simnet2 à un service Core :

   ```
   "%NCHOME%"\omnibus\bin\nco_pa_addentry -server TEST_PA -process simnet2
   -command "%NCHOME%"\omnibus\probes\win32\nco_p_simnet -host owl
   -retrycount 0 -managed -restart_msg "test" -alert_msg "testalert"
   -parentservice "Core"
   ```

2. Exécutez l'utilitaire **nco_pa_status** pour extraire le statut des services dans la configuration :

   ```
   "%NCHOME%"\omnibus\bin\nco_pa_status -server TEST_PA
   ```

   Où TEST_PA est le nom de l'agent de processus.

Lors de l'exécution de **nco_pa_status**, la sortie affiche un processus simnet2 doté d'un statut DEAD dans la définition de service Core. Le processus simnet2 ne démarre pas automatiquement car il fait partie d'un service. Pour exécuter le service, utilisez l'utilitaire **nco_pa_start**.

Concepts associés:

«Résolution du nom d'hôte au démarrage», à la page 249
Un hôte doit être spécifié pour chaque automatisation externe ou processus sous le contrôle d'un agent de processus. L'hôte est défini dans le fichier de configuration de l'agent de processus, dans le cadre de la définition d'un processus, ou bien dans l'automatisation externe du serveur ObjectServer.

Référence associée:

«Démarrage d'un service ou d'un processus (nco_pa_start)», à la page 282
Vous pouvez exécuter l'utilitaire **nco_pa_start** pour démarrer un service ou un processus à n'importe quel emplacement de la configuration du système de contrôle de processus.

Utilisation de Netcool/OMNIbus Administrator pour gérer le contrôle de processus

Netcool/OMNIbus Administrator possède une interface visuelle à partir de laquelle vous pouvez gérer le contrôle de processus. Vous pouvez utiliser Netcool/OMNIbus Administrator pour visualiser et gérer les agents de processus, les processus et les services sur vos hôtes Tivoli Netcool/OMNIbus.

Par exemple, vous pouvez visualiser le statut des services du contrôle de processus d'un ordinateur hôte, puis démarrer ou arrêter les processus de ces services.

Vous devez vous connecter à un agent de processus pour gérer ses services et processus. Les modifications de configuration apportées aux services et processus peuvent être enregistrées dans le fichier de configuration du contrôle de processus, qui est écrasé à chaque sauvegarde.

Remarque : Vous ne pouvez pas utiliser Netcool/OMNIbus Administrator pour spécifier que seuls certains hôtes se connectent aux agents de processus. Pour définir ces hôtes sécurisés, vous devez ajouter manuellement une définition de sécurité au fichier de configuration d'agent de processus.

Tâches associées:

«Configuration et gestion du contrôle de processus à partir de la ligne de commande», à la page 269
Vous pouvez définir des processus, services et hôtes dans le fichier de configuration du contrôle de processus. Vous pouvez également utiliser les utilitaires de ligne de commande pour démarrer, arrêter et ajouter un service ou processus, afficher l'état des services et processus, et arrêter un agent de processus.

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Connexion à un agent de processus

Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Pourquoi et quand exécuter cette tâche

Vous devez démarrer l'agent de processus automatiquement à l'aide des scripts de démarrage fournis sous UNIX ou en exécutant l'agent de processus comme un service Windows. Vous pouvez également démarrer l'agent de processus en exécutant la commande `$NCHOME/omnibus/bin/nco_pad`.

Pour vous connecter à un agent de processus, procédez comme suit :

Procédure

1. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Reports** (Rapports).
2. Cliquez sur l'icône **PA**. La fenêtre Process Agent Report (Rapport de l'agent de processus) s'ouvre. Elle affiche tous les agents de processus sélectionnés lors de la dernière exécution de l'assistant d'importation des connexions.

   Conseil : Après avoir démarré Netcool/OMNIbus Administrator, vous pouvez sélectionner **Fichier** > **Importer** à tout moment pour importer des nouvelles

informations de communication de réseau spécifiée dans l'éditeur de serveurs. Ces informations facilitent la communication entre les composants du serveur Tivoli Netcool/OMNIbus, tels que les serveurs ObjectServer, les passerelles, et agents de processus et les serveurs proxy.

3. Sélectionnez l'agent de processus auquel vous souhaitez vous connecter, puis réalisez l'une des actions suivantes :
   - S'il s'agit de votre première connexion ou que vous souhaitez entrer des informations d'authentification mises à jour à utiliser lors de la connexion, cliquez sur **Connect As** (Se connecter sous) dans la barre d'outils. La fenêtre Process Agent Security (Sécurité de l'agent de processus) s'ouvre. Allez à l'étape 4.
   - Si vous souhaitez vous connecter à l'aide des informations que vous venez d'indiquer, cliquez sur **Connect** (Connexion). La fenêtre Process Agent Security (Sécurité de l'agent de processus) s'ouvre avec les détails d'authentification précédemment spécifiés. Allez à l'étape 5.
4. Renseignez la fenêtre Process Agent Security (Sécurité de l'agent de processus) comme suit :

   Nom d'utilisateur
   Entrez le nom d'utilisateur permettant de vous connecter à l'agent de processus.

   Sous UNIX, tout utilisateur devant accéder à l'agent de processus doit être membre d'un groupe d'utilisateurs UNIX défini comme groupe d'administration à cette fin. Sous Windows, l'utilisateur doit être un utilisateur valide avec un compte local ou de domaine.

   Mot de passe
   Entrez le mot de passe utilisé pour se connecter à l'agent de processus.

   Always use for this connection (Toujours utiliser pour cette connexion)
   Cochez cette case pour indiquer que le nom d'utilisateur et le mot de passe doivent être sauvegardés et utilisés automatiquement lors des tentatives ultérieures de connexion à cet agent de processus. Ces paramètres sont valides pour toute la durée de la session d'application.

   Use as default (Utiliser comme valeur par défaut)
   Cochez cette case si vous voulez que les valeurs spécifiées pour le nom d'utilisateur et le mot de passe soient automatiquement renseignées lors du prochain affichage de cette fenêtre. Ces paramètres sont valides pour toute la durée de la session d'application.

   Remarque : Si vous cochez les deux cases, le paramètre **Always use for this connection** (Toujours utiliser pour cette connexion) prévaut.
5. Vérifiez ou annulez l'authentification comme suit :

   OK Cliquez sur ce bouton pour vérifier les données d'identification de connexion et fermer la fenêtre. Le panneau Service/Process Details (Détails de service/processus) s'ouvre. Ce panneau contient des informations sur les processus et les services configurés pour l'agent de processus sélectionné.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre Process Agent Security (Sécurité sur l'agent de processus) sans vous connecter à l'agent de processus. Vous retournez sur la fenêtre Process Agent Report (Rapport sur l'agent de processus).

Que faire ensuite

Lors de votre première connexion aux agents de processus existants avec Netcool/OMNIbus Administrator, les processus éloignés sont uniquement visibles s'ils ont été définis manuellement dans la configuration de l'agent de processus local. Pour les rendre visibles à tous les agents de processus liés, sélectionnez chaque service ou processus, cliquez sur le bouton **Edit** (Editer), puis fermer la fenêtre sans apporter de modifications. Sauvegardez chaque fichier de configuration à l'issue de ce processus.

Tâches associées:

«Démarrage automatique des agents de processus sous UNIX», à la page 266
Sous UNIX, des scripts de démarrage permettent de démarrer automatiquement l'agent de processus lorsque le système démarre.

«Démarrage automatique des agents de processus sous Windows», à la page 268
Sous Windows, vous pouvez installer l'agent de processus en tant que service Windows.

«Démarrage manuel des agents de processus», à la page 256
Vous pouvez démarrer manuellement les agents de processus à partir de la ligne de commande.

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

Affichage et configuration des informations d'état d'un agent de processus

Vous pouvez afficher les détails de version d'un agent de processus auquel vous êtes connecté et modifier le niveau de consignation des messages que génère l'agent de processus. Vous pouvez également configurer le routage de l'hôte en ajoutant les agents de processus à un groupe de routage.

Avant de commencer

Avant de tenter d'ajouter un agent de processus à un groupe de routage, vous devez utiliser l'option **File** > **Import** (Fichier > Importer) et l'assistant d'importation des connexions pour importer les détails de l'agent de processus dans Netcool/OMNIbus Administrator.

Pourquoi et quand exécuter cette tâche

Pour afficher et configurer les informations d'état d'un agent de processus, procédez comme suit :

Procédure

1. Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus à l'aide de Netcool/OMNIbus Administrator. Une fois la connexion établie, le panneau Service/Process Details (Détails du service/processus) s'ouvre par défaut.
2. Cliquez sur l'icône **Info** qui s'affiche à gauche de ce panneau. Le panneau PA Status Information (Information d'état PA) s'ouvre.
3. Renseignez ce panneau comme suit :

   Version
   Cette zone affiche la version de l'agent de processus.

Remarque : La version de l'agent de processus ne peut pas être déterminée si elle est inférieure à V7.0.1.

Niveau de débogage
Pour modifier le niveau de débogage d'un fichier journal, sélectionnez une autre valeur dans la liste déroulante. Cliquez sur la coche située à droite de la liste déroulante pour appliquer le niveau de débogage à la session en cours.

Pour sauvegarder le niveau de débogage mis à jour, cliquez sur **Save the Process Agent configuration** (Sauvegarder la configuration de l'agent de processus) dans la barre d'outils.

Defined Hosts (UNIX)/Related Process Agents (Windows) (Hôtes définis (UNIX) /Agents de processus associés (Windows))
Cette zone affiche une liste d'agents de processus connu de cet agent de processus. Les informations sont lues à partir du fichier de configuration de l'agent de processus, et peuvent être différentes sur les hôtes importés depuis le fichier d'interfaces par l'assistant d'importation de connexions.

Utilisez les boutons de droite pour supprimer des hôtes ou en ajouter à partir d'une liste d'hôtes connus.

Pour sauvegarder les modifications de définition d'hôte, cliquez sur **Save the Process Agent configuration** (Sauvegarder la configuration de l'agent de processus) dans la barre d'outils. Si vous ne suivez pas cette procédure, les modification seront perdues après fermeture.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Affichage des processus et services d'un agent de processus

Vous pouvez utiliser le panneau Service/Process Details (Détails du service/processus) dans Netcool/OMNIbus Administrator pour afficher les détails des processus et services configurés pour un agent de processus, et gérer ces processus et services.

Pourquoi et quand exécuter cette tâche

Pour afficher les processus et services configurés pour un agent de processus, exécutez l'une des actions suivantes :

Procédure

- Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus à l'aide de Netcool/OMNIbus Administrator. Une fois la connexion établie, le panneau Service/Process Details (Détails du service/processus) s'ouvre par défaut.
- Si vous êtes déjà connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Status** (Etat) dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.

Résultats

Dans le panneau Service/Process Details (Détails de service/processus), les détails suivants s'affichent pour chaque service et processus configuré :

- Le nom unique attribué au service ou au processus
- Le statut du service ou du processus
- L'hôte sur lequel le processus s'exécute ; cette colonne est vide pour les services
- L'identificateur d'un processus en cours d'exécution ; cette colonne est vide pour les services et les processus qui ne sont pas en cours d'exécution

Dans la colonne **Nom**, l'icône située à gauche du nom identifie l'entrée comme un service, et l'icône identifie l'entrée comme un processus. Les processus sont également regroupés sous le service dans lequel ils s'exécutent, et leur nom s'affichent sous le format suivant :

nom_service:*nom_processus*

Par exemple : `Core:MasterObjectServer`

Dans la colonne **Status**, l'icône de statut a une forme de cercle, et sa couleur indique si le processus ou service est en cours d'exécution :

- Vert : le service ou processus est en cours d'exécution.
- Bleu : le service est marginal. Tous les processus ne sont pas en cours d'exécution.
- Jaune : le processus est en attente. Le processus attend que le délai de dépendance soit écoulé. Ce statut indique également que le processus n'a pas pu démarrer correctement, quelles que soient les dépendances de processus.
- Gris : le processus est inactif ou le service s'est arrêté.
- Rouge : identifie un niveau d'erreur de statut, qui indique que le niveau de statut ne peut être récupéré auprès de l'agent de processus.

Dans le panneau Service/Process Details (Détails de service/processus), vous pouvez gérer des processus et des services pour l'agent de processus sélectionné et effectuer les actions suivantes :

- Créer ou modifier un service
- Créer ou modifier un processus
- Supprimer un service ou un processus sélectionné
- Démarrer un service ou un processus sélectionné
- Arrêter un service ou un processus sélectionné
- Copier-coller un service ou processus dans le même agent de processus, ou sur d'autres hôtes d'agent de processus
- Arrêter l'agent de processus
- Exécuter une action externe
- Envoyer un signal à un processus
- Sauvegarder le fichier de configuration de l'agent de processus sur le disque
- Actualiser le contenu du panneau (en cliquant sur le bouton **Actualiser** dans la barre d'outils)

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus

Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Configuration des services d'un agent de processus

Vous pouvez utiliser Netcool/OMNIbus Administrator pour créer, modifier, supprimer, démarrer et arrêter des services définis pour être exécutés sous un agent de processus. Lorsque vous modifiez un service, vous pouvez choisir d'enregistrer vos modifications dans le fichier de configuration du contrôle de processus.

Création et édition de services

Vous pouvez configurer le démarrage automatique ou manuel d'un service Tivoli Netcool/OMNIbus.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un service, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.
2. Pour ajouter un service, cliquez sur **New Service** (Nouveau service) dans la barre d'outils. Pour éditer un service, sélectionnez-le, puis cliquez sur **Edit** (Editer) dans la barre d'outils.

   La fenêtre Service Details (Détails du service) s'ouvre.
3. Complétez cette fenêtre comme suit :

   Nom Entrez le nom du service. Ce nom doit être unique dans le réseau de contrôle de processus. Si vous modifiez un service, vous ne pouvez pas le renommer.

   Auto Start Service (Démarrage automatique du service)
   Cochez cette case pour indiquer que le service doit démarrer automatiquement lorsque l'agent de processus est lancé. Décochez cette case si vous souhaitez démarrer le service manuellement en utilisant la commande **nco_pa_start**.

   Master Service (Service maître)
   Cochez cette case pour signaler qu'il s'agit d'un service maître. Un service maître démarre avant les autres services et il est géré comme un service dont les autres services dépendent.

Si vous définissez plusieurs services maîtres dans le même fichier de configuration de contrôle de processus, les services maîtres démarrent selon leur ordre d'apparition dans le fichier de configuration.

Processes and their order within the configuration file (Les processus et leur ordre dans le fichier de configuration)
Cette liste s'affiche uniquement lorsque vous modifiez un service ; elle indique les processus définis pour s'exécuter au sein d'un service. L'ordre d'affichage des processus reflète leur ordre dans le fichier de configuration. Utilisez les touches de déplacement pour modifier l'ordre des processus.

4. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails de service et fermer la fenêtre. Les nouveaux services sont ajoutés au panneau Service/Process Details (Détails de service/processus).

Conseil : Les modifications de configuration ne sont pas automatiquement sauvegardées dans le fichier de configuration de contrôle de processus. Pour mettre à jour le fichier, cliquez sur **Write Process Agent Config file** (Ecrire le fichier de configuration de l'agent de processus) dans la barre d'outils.

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Suppression d'un service

Lorsque vous supprimez un service, vous le retirez de la configuration de contrôle de processus.

Pourquoi et quand exécuter cette tâche

Pour supprimer un service, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.

2. Sélectionnez le service à supprimer, cliquez sur **Delete** (Supprimer) dans la barre d'outils et confirmez la suppression. Le service est supprimé et le panneau Service/Process Details (Détails du service/processus) mis à jour.

Résultats

Conseil : Les modifications apportées à la configuration ne sont pas automatiquement sauvegardées dans le fichier de configuration de contrôle de processus. Pour mettre à jour le fichier, cliquez sur **Write Process Agent Config File** (Ecrire le fichier de configuration d'agent de processus) dans la barre d'outils.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Démarrage d'un service

Vous pouvez uniquement démarrer les services à l'état Stopped (Arrêté).

Pourquoi et quand exécuter cette tâche

Pour démarrer un service, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.
2. Sélectionnez le service à démarrer et cliquez sur **Start** (Démarrer) dans la barre d'outils.

Résultats

Le service et ses processus associés sont démarrés, et le panneau Service/Process Details (Détails du service/processus) est mis à jour.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Arrêt d'un service

Vous pouvez uniquement arrêter les services à l'état Marginal ou Running (En cours d'exécution).

Pourquoi et quand exécuter cette tâche

Pour arrêter un service, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.
2. Sélectionnez le service à arrêter et cliquez sur **Stop** (Arrêter) (Démarrer) dans la barre d'outils.

Résultats

Le service et ses processus définis sont arrêtés, et le panneau Service/Process Details (Détails du service/processus) est mis à jour.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Configuration des processus

Netcool/OMNIbus Administrator permet de créer, de modifier, de démarrer et d'arrêter les processus d'un service configuré pour être exécuté sous un agent de processus. Vous pouvez également envoyer des signaux aux processus.

Lorsque vous modifiez la configuration d'un processus, vous pouvez choisir d'enregistrer vos modifications dans le fichier de configuration du contrôle de processus.

Création et édition des processus

Vous pouvez configurer un processus de manière à l'exécuter dans le cadre d'un service.

Avant de commencer

Au moins un service doit exister avant de configurer des processus.

Pourquoi et quand exécuter cette tâche

Pour créer ou éditer un processus, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.
2. Pour ajouter un processus, cliquez sur **New Process** (Nouveau processus) dans la barre d'outils. Pour éditer un processus, sélectionnez-le, puis cliquez sur **Edit** (Editer) dans la barre d'outils.

   La fenêtre Process Details (Détails du processus) s'ouvre.
3. Définissez un nouveau processus comme suit :

   Nom Entrez le nom de processus. Ce nom doit être unique pour l'agent de processus. Lorsque vous modifier un processus, vous ne pouvez pas le renommer.

   Commande
   Entrez la chaîne de commande qui lance le processus, comme si vous la saisissiez dans la ligne de commande. Utilisez le chemin complet de la commande. Par exemple, pour configurer un serveur ObjectServer appelé NCOMS, avec l'agent de processus appelé SFOSYS1_PA, entrez :

   ```
   $NCHOME/omnibus/bin/nco_objserv -name NCOMS -pa SFOSYS1_PA
   ```

   Hôte Entrez le nom de l'hôte sur lequel le processus s'exécute. Le contrôle de processus résout automatiquement le nom de l'agent de processus, lorsque cela est requis.

   Maintenance
   Sélectionnez le service sous lequel le service sera exécuté. Lorsque vous modifiez un processus, vous ne pouvez pas changer le service.
4. Dans l'onglet **Process** (Processus), indiquez des détails supplémentaires sur le processus. Renseignez l'onglet comme suit :

   Managed (Géré)
   Cochez cette case si vous souhaitez que le processus redémarre automatiquement lorsqu'il échoue.

Run As ID (Exécuter en tant qu'ID)
Entrez 'ID utilisateur sous lequel le processus est exécuté. Cette valeur est généralement 0, ce qui correspond au nom d'utilisateur `root`.

Remarque : Si l'agent de processus n'est pas en cours d'exécution sous le compte utilisateur `root`, cette option est ignorée, et le processus est exécuté sous le compte d'utilisateur qui exécute l'agent de processus.

Retry Count (Nombre de tentatives)
Indiquez le nombre de tentatives de redémarrage à effectuer si le processus se ferme dans le délai spécifié par l'option de ligne de commande `-retrytime` de l'agent de processus. Si ce délai est défini sur `0`, il n'y a pas de limite au nombre de tentatives de redémarrage. la valeur par défaut est `0`.

Process Type (Type de processus)
Dans cette liste déroulante, sélectionnez `PA Aware` pour vous assurer que le processus sait que le contrôle de processus est actif, et pour activer l'utilisation de toutes les fonctions de contrôle de processus, telles que les dépendances. Sélectionnez `Not PA Aware` si le processus peut être géré par le contrôle de processus, mais ne peut pas utiliser toutes les fonctions de contrôle de processus.

Dependency (Dépendance)
Cette liste déroulante permet d'indiquer si le processus a des dépendances. L'option sélectionnée détermine l'affichage des zones restantes :

- Sélectionnez `Aucune` pour indiquer que le processus n'a aucune dépendance sur un autre processus. Aucune autre zone n'est affichée.
- Sélectionnez `Delay` (Délai) pour indiquer un délai d'attente avant le lancement du processus. Une zone supplémentaire nommée **Start delay** (Délai de lancement) s'affiche pour que vous puissiez définir une dépendance de temps.
- Sélectionnez `Process` (Processus) pour indiquer une dépendance relative à un autre processus. Une zone supplémentaire nommée **Select Delay from** (Sélectionnez le délai à partir de) ou **Dependent Name** (Nom dépendant) est affichée pour que vous indiquiez un processus.

Start delay (Délai de lancement)
Indiquez le délai d'attente (en secondes) avant le lancement du processus. Ce délai est mesuré à partir du lancement du service.

Select Delay from/Dependent Name (Sélectionnez le délai à partir de/Nom dépendant)
Cette liste déroulante permet de sélectionner un autre processus compatible avec l'agent de processus et dont le processus créé ou modifié dépend. Pour que le processus en cours de création ou modification démarre, le processus dont il dépend doit être en cours d'exécution.

5. Dans l'onglet **Messages**, indiquez les informations relatives au message à envoyer au journal système UNIX ou à l'Observateur d'événements Windows lorsque vous redémarrez ou quittez le processus. Renseignez l'onglet comme suit :

Restart (Redémarrage)
Entrez le message à envoyer au syslog UNIX ou à l'Observateur

d'événements Windows lorsque le processus est redémarré. Par exemple : `le serveur ObjectServer a été redémarré`.

Alerte Entrez le message à envoyer au syslog UNIX ou à l'Observateur d'événements Windows si le processus se ferme. Par exemple : `le serveur ObjectServer a fermé`.

Lorsque l'agent de processus génère un message d'alerte ou de redémarrage, ce message est transféré dans le syslog ou l'Observateur d'événements. Tivoli Netcool/OMNIbus dispose d'un sonde Syslog qui permet de surveiller ces messages et de les convertir en alertes ObjectServer.

Les messages d'alerte et de restauration envoyés au syslog UNIX ou à l'Observateur d'événements Windows constituent des avertissements. Le message est formaté comme suit :

NOMHOTE **:** *MSG_ALERTE_OU_REDEMARRAGE* **:** *MSG*

La valeur *NOMHOTE* est le nom de l'hôte qui a signalé le problème. *MSG8ALERTE_OU_REDEMARRAGE* décrit le type de message. *MSG* est le texte défini dans le fichier de configuration pour ce processus.

Vous pouvez utiliser les mots clés d'expansion décrits dans la table suivante dans les entrées d'alerte et de redémarrage spécifiées dans l'onglet **Messages**. Les mots clés d'expansion agissent comme des variables et contiennent des informations sur le processus qui s'est fermé ou a redémarré.

Tableau 90. Mots clés d'expansion

Mot clé d'expansion	Description
${NAME}	Nom du processus.
${HOST}	Nom de l'hôte exécutant le processus.
${EUID}	L'ID utilisateur effectif sous lequel le processus s'exécute.
${COMMAND}	La commande définissant le processus.

6. Sauvegardez ou annulez les modifications comme suit :

OK Cliquez sur ce bouton pour sauvegarder les détails de processus et fermer la fenêtre. Les nouveaux processus sont ajoutés au panneau Service/Process Details (Détails de processus/service).

Conseil : Les modifications de configuration ne sont pas automatiquement sauvegardées dans le fichier de configuration du contrôle de processus. Pour mettre à jour ce fichier, cliquez sur **Write Process Agent Config File** (Ecrire le fichier de configuration de l'agent de processus) dans la barre d'outils.

Annuler
Cliquez sur ce bouton pour fermer la fenêtre sans enregistrer vos modifications.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Référence associée:

«Définition des processus dans le fichier de configuration de l'agent de processus», à la page 270
Le fichier de configuration de l'agent de processus permet de définir la liste des processus à exécuter par les agents de processus.

Suppression d'un processus

Lorsque vous supprimez un processus, vous le retirez du service auquel il a été attribué.

Pourquoi et quand exécuter cette tâche

Pour supprimer un processus, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.
2. Sélectionnez le processus à supprimer, cliquez sur **Delete** (Supprimer) dans la barre d'outils et confirmez la suppression. Le processus est supprimé de la définition de service et le panneau Service/Process Details (Détails du service/processus) est mis à jour.

Résultats

Conseil : Les modifications apportées à la configuration ne sont pas automatiquement sauvegardées dans le fichier de configuration de contrôle de processus. Pour mettre à jour le fichier, cliquez sur **Write Process Agent Config File** (Ecrire le fichier de configuration d'agent de processus) dans la barre d'outils.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Démarrage d'un processus

Vous pouvez uniquement démarrer les processus à l'état Dead (Inactif).

Pourquoi et quand exécuter cette tâche

Pour démarrer un processus, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.
2. Sélectionnez le processus à démarrer et cliquez sur **Start** (Démarrer) dans la barre d'outils.

Résultats

Le processus démarre et le panneau Service/Process Details (Détails du service/processus) est mis à jour avec l'état du processus, y compris l'ID processus du processus d'exécution.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Arrêt d'un processus

Vous pouvez uniquement arrêter les processus à l'état Pending (En attente) ou Running (En cours d'exécution).

Pourquoi et quand exécuter cette tâche

Lorsque vous arrêtez un processus, l'état des processus dépendants reste inchangé. Par exemple, si un ObjectServer est arrêté, l'exécution des analyses dépendantes de ce serveur se poursuit.

Pour arrêter un processus, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :

- Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
- Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
- Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.

2. Sélectionnez le processus à arrêter et cliquez sur **Stop** (Arrêter) (Démarrer) dans la barre d'outils.

Résultats

Le processus est arrêté et le panneau Service/Process Details (Détails du service/processus) mis à jour.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Envoi de signaux aux processus

Vous pouvez utiliser Netcool/OMNIbus Administrator pour envoyer un signal UNIX à un processus.

Pourquoi et quand exécuter cette tâche

Par exemple, une sonde peut réécrire son fichier de règles. Pour ce faire, envoyez un signal SIGHUP(1) au processus de la sonde.

Pour envoyer un signal à un processus, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.
2. Sélectionnez le processus auquel vous souhaitez envoyer un signal, puis cliquez sur **Send Signal** (Envoyer un signal) dans la barre d'outils. La fenêtre Send Signal (Envoyer un signal) s'ouvre.

3. Complétez cette fenêtre comme suit :

   Process Name (Nom de processus)
   Sélectionnez le processus auquel le signal doit être envoyé.

   Signal Sélectionnez le signal que vous souhaitez envoyer. Les signaux valides sont les suivants :

   - `SIGHUP(1)` : Signal de raccrochage pour arrêter et redémarrer un processus
   - `SIGINT(2)` : Signal d'interruption
   - `SIGTERM(15)` : Signal de fin

   Sous Windows, les signaux `SIGINT(2)` et `SIGTERM(15)` sont pris en charge uniquement par les processus Tivoli Netcool/OMNIbus ; par exemple, les serveurs ObjectServer, les serveurs proxy et les sondes. Vous pouvez également utiliser d'autres méthodes disponibles pour arrêter les processus. Lorsqu'une sonde est en cours d'exécution sous contrôle de processus, le signal `SIGHUP(1)` peut être utilisé pour que la sonde relise son fichier de règles.

4. Confirmez ou annulez vos modifications comme suit :

   OK Cliquez sur ce bouton pour fermer la fenêtre et envoyer le signal au processus sélectionné.

   Appliquer
   Cliquez sur ce bouton pour envoyer le signal au processus sélectionné et laisser la fenêtre ouverte.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre et ne pas procéder à l'envoi du signal.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Copie et collage d'un service ou d'un processus entre les hôtes de l'agent de processus

Vous pouvez copier et coller un service sélectionné et ses processus associés ou un processus sélectionné, à l'intérieur d'un agent de processus ou d'un agent de processus vers un autre.

Pourquoi et quand exécuter cette tâche

Pour copier et coller un service ou un processus, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.

- Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
- Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.

2. Dans le panneau Service/Process Details (Détails du service/processus) de l'agent de processus à partir duquel vous souhaitez copier un service ou un processus, sélectionnez le service ou le processus, puis cliquez sur **Copy** (Copier) dans la barre d'outils. Le service sélectionné et ses processus associés ou le processus sélectionné sont copiés dans le presse-papiers.
3. Pour coller le contenu du presse-papiers dans le même agent de processus, cliquez sur **Paste** (Coller) dans la barre d'outils du panneau Service/Process Details (Détails du service/processus) en cours.

   Pour coller le contenu du presse-papiers dans un autre agent de processus connecté, accédez au panneau Service/Process Details (Détails du service/processus) de cet agent de processus, puis cliquez sur **Paste** (Coller) dans la barre d'outils.
4. Si vous avez copié un processus, le panneau Process Details (Détails du processus) s'ouvre. Les noms de processus devant être uniques par agent de processus, renommez le processus si nécessaire, et procédez aux autres modifications requises. Cliquez sur **OK** pour fermer cette fenêtre et collez les détails du processus dans le panneau Service/Process Details (Détails du service/processus). Par ailleurs, cliquez sur **Cancel** (Annuler) pour annuler l'opération de collage.
5. Si vous avez copié un service, l'assistant Process Agent Consistency Checker (Vérificateur de cohérence de l'agent de processus) s'ouvre. Les noms de service et noms de processus devant être uniques dans le réseau de contrôle de processus et par agent de processus, respectivement, il peut s'avérer nécessaire de les renommer. Réalisez les actions suivantes :
   a. Cliquez sur **Next** (Suivant) pour passer à la page suivante, puis modifiez les détails du service ou du processus, la cas échéant. Vous pouvez spécifier un nouveau nom pour le service en écrasant son nom. Pour modifier les détails de chaque processus, cliquez deux fois sur l'entrée de processus pour ouvrir la fenêtre Process Details (Détails du processus), puis procédez aux modifications de votre choix. Vous pouvez également indiquer si vous souhaitez inclure ou exclure un processus dans l'action de collage. Sous UNIX, cliquez dans la cellule **Include in Paste** (Inclure dans le collage) pour basculer entre les options **true** et **false**. Sous Windows, cliquez à l'aide du bouton droit de la souris sur l'entrée de processus, puis sélectionnez ou désélectionnez l'option **Include in Paste** (Inclure dans le collage) dans le menu contextuel.
   b. Cliquez sur **Next** (Suivant) pour afficher un récapitulatif du service et des processus à coller.
   c. Cliquez sur **Finish** (Terminer) pour fermer l'assistant et coller les détails dans le panneau Service/Process Details (Détails du service/processus). Vous pouvez également cliquer sur **Cancel** (Annuler) pour annuler l'opération de collage.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Exécution d'une action externe

Vous pouvez utiliser Netcool/OMNIbus Administrator pour exécuter une commande sur un hôte.

Pourquoi et quand exécuter cette tâche

Pour exécuter une action externe, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous n'êtes pas encore connecté, connectez-vous à l'agent de processus. Lorsque la connexion a abouti, le panneau Service/Process Details (Détails sur le service/processus) s'ouvre.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.
2. Pour spécifier les détails de la commande, cliquez sur **Run External Action** (Exécuter une action externe) dans la barre d'outils. La fenêtre External Action Details (Détails de l'action externe) s'ouvre.
3. Complétez cette fenêtre comme suit :

   Commande
   Entrez une commande valide à exécuter sur l'hôte.

   Hôte Sélectionnez l'hôte sur lequel la commande doit être exécutée.
4. Confirmez ou annulez vos modifications comme suit :

   OK Cliquez sur ce bouton pour fermer la fenêtre et exécuter la commande.

   Annuler
   Cliquez sur ce bouton pour fermer la fenêtre sans exécuter la commande.

Tâches associées:

«Démarrage de Netcool/OMNIbus Administrator», à la page 70
Vous devez exécuter l'utilitaire **nco_config** pour démarrer Netcool/OMNIbus Administrator.

«Connexion à un agent de processus», à la page 289
Avant de vous connecter à un agent de processus à l'aide de Netcool/OMNIbus Administrator, vous devez vous assurer qu'il a été démarré.

Arrêt d'un agent de processus

Vous pouvez arrêter un agent de processus et tous ses processus gérés ou arrêter un agent de processus et laisser ses processus fonctionner.

Pourquoi et quand exécuter cette tâche

Pour arrêter un agent de processus, procédez comme suit :

Procédure

1. Exécutez l'une des actions suivantes à partir de Netcool/OMNIbus Administrator :
   - Si vous êtes connecté à l'agent du processus et que le panneau Service/Process Details (Détails sur le service/processus) est ouvert, allez à l'étape suivante.
   - Si vous êtes connecté à l'agent de processus, mais que le panneau Service/Process Details (Détails sur le service/processus) n'est pas affiché, cliquez sur l'icône **Etat** dans la fenêtre de configuration de l'agent de processus pour que ce panneau s'affiche.
2. Cliquez sur **Stop the Process Agent** (Arrêter l'agent de processus) dans la barre d'outils.
3. A l'invite, choisissez l'option permettant d'indiquer si vous souhaitez arrêter l'agent de processus et tous ses processus gérés ou arrêter l'agent de processus et laisser ses processus fonctionner. Cliquez sur **OK**.

Résultats

L'agent de processus et, éventuellement, ses processus sont arrêtés.

Utilisation du contrôle de processus pour exécuter des procédures externes dans les automatisations

Le système de contrôle de processus exécute les procédures externes spécifiées dans les automatisations. Les procédures externes sont exécutées sur un hôte local ou distant.

Une automatisation n'exécute pas de programmes directement. Elle envoie une requête à un agent de processus. Si nécessaire, l'agent de processus transmet la requête à l'agent de processus exécuté sur l'hôte spécifié. L'agent de processus distant exécute ensuite le programme demandé.

Les procédures externes peuvent être transmises entre différents environnements de système d'exploitation et les agents de processus d'un système d'exploitation peuvent exécuter des automatisations envoyées par les agents de processus d'un autre système d'exploitation.

Lorsque vous exécutez l'agent de processus à partir de la ligne de commande sous UNIX et Windows, le répertoire de base de tous les processus enfant est le répertoire dans lequel l'agent de processus a été démarré.

Lorsque vous exécutez l'agent de processus en tant que démon UNIX, le répertoire de base de tous les processus enfant est `$NCHOME/omnibus`.

Sous Windows, le répertoire par défaut d'un agent de processus s'exécutant en tant que service Windows est `%NCHOME%\omnibus\log`. Ce répertoire est également le répertoire de base par défaut de tout processus enfant généré par l'agent de processus.

Conseil : Lors de l'exécution de procédures externes, les propriétés du serveur ObjectServer **PA.Username** et **PA.Password** doivent posséder un nom d'utilisateur et un mot de passe valides dans le fichier de propriétés du serveur ObjectServer à des fins d'authentification. La propriété du serveur ObjectServer **PA.Name** doit également posséder le nom de l'agent de processus que le serveur ObjectServer utilise pour exécuter des automatisations externes. Ces paramètres garantissent la réussite de la connexion à l'agent de processus et de l'exécution de la procédure externe.

Concepts associés:

«Résolution du nom d'hôte au démarrage», à la page 249
Un hôte doit être spécifié pour chaque automatisation externe ou processus sous le contrôle d'un agent de processus. L'hôte est défini dans le fichier de configuration de l'agent de processus, dans le cadre de la définition d'un processus, ou bien dans l'automatisation externe du serveur ObjectServer.

Référence associée:

«Propriétés et options de ligne de commande du serveur ObjectServer», à la page 4
Utilisez les propriétés ou les options de ligne de commande ObjectServer pour configurer les paramètres du serveur ObjectServer. Pour éviter les erreurs, ajoutez autant de propriétés que possible au fichier de propriétés plutôt que des options de ligne de commande. D'autres utilitaires sont fournis que vous pouvez utiliser pour chiffrer les valeurs de propriété.

Chapitre 8. Optimisation des performances

Il est possible de mesurer les performances de Tivoli Netcool/OMNIbus en termes de temps de réponse, de débit et de disponibilité.

Les métriques de ces performances sont affectées par différents facteurs, notamment :

- Le nombre d'événements, notamment les détails et les journaux, dans le serveur ObjectServer

  Ce nombre a un rapport direct avec le volume de données transférées lors des opérations de resynchronisation et avec le nombre de lignes devant être analysées lors de requêtes SQL provenant de clients et de déclencheurs.
- Débit des événements dans le système à partir des sondes ou des passerelles entrantes
- Débit des événements en dehors du système à partir des passerelles sortantes
- Le nombre de clients de l'interface graphique Web et d'affichage de bureau concomitant ainsi que le nombre d'autres clients connectés qui se disputent le temps du serveur ObjectServer
- Le nombre et la complexité des filtres de l'interface graphique Web et du bureau ainsi que le nombre de connexions et la complexité des actions dans les règles soumises par les clients Netcool/Impact, le cas échéant.
- L'efficacité des automatisations du serveur ObjectServer personnalisées et si elles soumettent le serveur ObjectServer à une charge de travail inutile
- Paramètres de granularité du serveur ObjectServer (fréquence d'envoi de diffusions IDUC aux clients)
- Votre configuration matérielle et système

Pour obtenir un niveau de performances optimal au niveau du serveur ObjectServer, vous pouvez surveiller le nombre et le débit des clients qui se connectent, évaluer l'efficacité des automatisations du serveur ObjectServer et concevoir des index efficaces prenant en charge vos requêtes SQL. Tenez également compte d'une architecture à plusieurs niveaux pour prendre en charge la haute disponibilité de votre installation Tivoli Netcool/OMNIbus.

Concepts associés:

«Surveillance automatique de Tivoli Netcool/OMNIbus», à la page 326
Vous pouvez surveiller une plage d'indicateurs clés de performances (KPI) au sein de l'infrastructure du serveur ObjectServer en activant l'extension de surveillance automatique de Tivoli Netcool/OMNIbus. Les données de performances sont signalées sous forme d'événements synthétiques et peuvent être affichées dans la liste d'événements ou dans un tableau de bord de l'interface graphique Web Netcool Health. Les données d'événement sont également consignées dans un fichier journal de surveillance automatique.

© Copyright IBM Corp. 1994, 2014

Indicateurs clés de performances Tivoli Netcool/OMNIbus

Pour être certain qu'un système Tivoli Netcool/OMNIbus s'exécute efficacement, vous avez la possibilité de surveiller plusieurs indicateurs clés de performances (KPI).

Chaque description de KPI définit le KPI, indique comment le configurer, explique son utilité et donne des indications sur ce qu'il faut surveiller, c'est-à-dire quelles les valeurs KPI indiquent de bonnes performances et quelles sont celles qu'il serait bon de vérifier. Pour de nombreux KPI, les valeurs indiquées ne sont pas absolues ; à la place, vous pouvez utiliser ces valeurs à titre indicatif. Pour votre propre système Tivoli Netcool/OMNIbus, vous devez établir des valeurs de base de référence sur un système connu pour être en bon état de marche (par exemple sur un environnement de test) et les comparer avec les valeurs actuelles et futures sur le système de production.

Les KPI fournis ne constituent pas une liste exhaustive. Selon votre environnement Tivoli Netcool/OMNIbus, certains KPI demandent une surveillance accrue, tandis que pour d'autres, la surveillance peut être insignifiante.

Indicateurs clés de performances du serveur ObjectServer

Surveillez ces indicateurs clés de performances (KPI) pour déterminer l'efficacité du serveur ObjectServer.

Les indicateurs clés de performances du serveur ObjectServer suivants peuvent être surveillés :

Durée totale utilisée pour la granularité

Pour configurer ce KPI, exécutez le serveur ObjectServer en activant le profilage et exécutez l'automatisation trigger_stats_report. Les données de profilage indiquent pendant combien de temps le serveur ObjectServer a profilé les demandes provenant des clients, y compris le temps passé sur les automatisations causées par l'activité du client. Le fichier `trigger_stats.log` affiche la durée utilisée par chaque déclencheur dans la dernière période de profilage. Pour montrer que le serveur ObjectServer peut traiter la totalité des demandes tout en respectant le temps alloué, le temps de traitement doit être inférieur au temps de la période de rapport. Notez que la tendance de ce KPI au fil du temps est moins importante que sa valeur à un moment précis.

Conseil : Sur les serveurs multiprocesseurs, le temps passé à traiter les demandes des clients et les automatisations peut être supérieur à la période de rapport à cause de la nature à unités d'exécutions multiples du serveur ObjectServer.

Utilisation de l'unité centrale du processus nco_objserv

Pour surveiller l'utilisation de l'unité centrale du processus nco_objserv, configurez et utilisez un agent IBM Tivoli Monitoring.

L'utilisation de l'unité centrale des processus est l'une des métriques les plus importantes dans l'évaluation des performances ; pour un serveur ObjectServer avec une charge importante, la performance est liée à l'utilisation de l'unité centrale.

Conseil : Lors de l'identification et de la résolution des problèmes liés aux performances, les fichiers journaux du profil et des statistiques de déclencheur sont le premier endroit où débuter vos recherches.

Généralement, si la durée totale combinée des clients et des déclencheurs est constamment supérieure à 60 secondes (la période de granularité par défaut), vous devez effectuer certaines actions. Diverses métriques du système d'exploitation peuvent également vous être utiles pour identifier si un système est sous contrainte. Les métriques clés sont l'utilisation de l'unité centrale et la taille des processus Tivoli Netcool/OMNIbus.

Nombre de lignes de la table alerts.status

Les performances des demandes non indexées sont proportionnelles au nombre de lignes de la table alerts.status. L'augmentation du nombre de lignes de la table alerts.status est proportionnelle à l'augmentation du temps nécessaire à l'accomplissement de toutes les demandes et à l'exécution de tous les déclencheurs sur la table alerts.status. De plus, la mise en attente des données de ligne représente la majorité de la mémoire utilisée par le serveur ObjectServer et les données de la table alerts.status représentent la plupart de ces données de ligne. Le nombre de lignes de la table alerts.status évolue en fonction des réseaux et des applications sous surveillance. Toutefois, le nombre de lignes devrait rester stable sur quelques semaines. Si le nombre de lignes de la table alerts.status augmente au fil des semaines et que le taux de cette augmentation est important, cela peut indiquer qu'un problème se développe. La table alerts.status utilise plus de mémoire que les tables alert.details et alerts.journal. Pour en savoir davantage sur le nombre d'événements répertoriés dans la table alerts.status, consultez «Autres informations utiles», à la page 313.

Nombre de lignes dans la table alerts.details

Les performances des requêtes émises par les clients d'interface graphique Web pour l'extraction des détails des alertes sont proportionnelles au nombre d'entrées dans la table alerts.details. Plus la table alerts.details contient d'entrées, plus les demandes ralentissent. En outre, la mise en attente des données de ligne représente la majorité de la mémoire utilisée par le serveur ObjectServer et les données de la table alerts.details représentent la plupart de ces données de ligne. Chacune des entrées de la table alerts.details augmente la charge de toute passerelle configurée pour réacheminer ces données vers d'autres serveurs ObjectServer ou d'autres applications. Le nombre de ligne de la table alerts.details évolue en fonction des réseaux et des applications sous surveillance, du jour de la semaine, de l'heure, etc. Toutefois, le nombre de lignes devrait rester stable pendant quelques semaines. Si le nombre de lignes de la table alerts.details augmente au fil du temps, cela peut indiquer qu'un problème se développe. De même, au fur et à mesure que de nouvelles unités et de nouveaux fichiers de règles d'analyse sont introduits, un problème peut se développer. La table alerts.details utilise plus de mémoire que la table alerts.journal, mais moins que la table alerts.status.

Conseil : La fonction **nvp_add** permet de déplacer des données provenant des entrées de la table alerts.details vers la zone ExtendedAttr de la table alerts.status. L'utilisation de la fonction **nvp_add** signifie que le nombre d'insertions effectuées par la sonde est réduit.
Pour en savoir davantage sur le nombre d'événements répertoriés dans la table alerts.details, voir «Autres informations utiles», à la page 313.

Nombre de lignes dans la table alerts.journal

Le nombre de lignes dans la table alerts.journal évolue en fonction de l'automatisation qui traite les événements ou de l'utilisateur chargé de traiter ces événements. Toutefois, le nombre de lignes devrait rester stable

pendant quelques semaines. Si le nombre de lignes de la table alerts.journal augmente au fil des semaines et que le taux de l'augmentation est important, cela peut indiquer qu'un problème se développe. Veillez à ce qu'une entrée ne soit ajoutée à la table alerts.journal que lorsqu'un déclencheur affecte le contenu d'une ligne, et cela peu importe la fréquence du déclencheur. Utilisez les indicateurs pour vous assurer qu'un événement n'est pas inutilement retraité par le même déclencheur.Pour en savoir davantage sur le nombre d'événements répertorié dans la table alerts.journal, voir «Autres informations utiles», à la page 313.

Nombre d'insertions dans la table alerts.status effectuées au cours des *n* dernières secondes

Pour ce KPI, les *insertions* font référence à de nouvelles lignes et des dédoublonnages. Pour surveiller le nombre d'insertions dans la table alerts.status, activez le groupe de déclencheurs stats_triggers et l'automatisation statistics_gather. Les données regroupées par ce groupe de déclencheurs et cette automatisation sont écrites périodiquement dans la table master.stats, dans la colonne StatusInserts. L'attribut de performances peut être déduit en comparant la valeur de la colonne StatusInserts avec la valeur provenant du rapport précédent. Ce KPI vous permet d'identifier d'importantes augmentations du nombre d'entrées vers le système causées par la surveillance des nouvelles unités ou applications. Il peut permettre d'augmenter l'évolutivité de votre système et de votre capacité à planifier. Le nombre de ligne dans la table alerts.status évolue en fonction des réseaux et des applications sous surveillance, du jour de la semaine, de l'heure, etc. Toutefois, le nombre de lignes devrait rester stable pendant quelques semaines. Si le nombre de lignes de la table alerts.status augmente au fil des semaines ou à mesure que de nouvelles unités ou de nouveaux fichiers de règles d'analyse sont introduits, et que le taux de ces augmentations est important, cela peut indiquer un problème (par exemple, une erreur ou un problème lié au fichier de règles d'analyse).

Utilisation de la mémoire du processus nco_objserv

Pour surveiller l'utilisation de la mémoire du processus nco_objserv, configurez et utilisez l'agent IBM Tivoli Monitoring pour Tivoli Netcool/OMNIbus. Surveillez l'utilisation de la mémoire du processus nco_objserv pour vous assurer que cette utilisation n'avoisine aucune limite de mémoire, pour vous aider à identifier la moindre augmentation de l'utilisation de la mémoire et pour déterminer la cause des augmentations. L'utilisation de la mémoire du processus augmente proportionnellement à l'augmentation du nombre de lignes des tables alerts.status, alerts.details et alerts.journal (ou toute autre table supplémentaire que vous avez définie), à l'augmentation du nombre de connexions et à l'utilisation croissante des clients. L'utilisation de la mémoire devrait rester stable dans le temps et toute augmentation devrait correspondre aux augmentations du nombre de lignes de tables ou de clients supplémentaires.

Pour de plus amples informations sur IBM Tivoli Monitoring Agent for Tivoli Netcool/OMNIbus, voir IBM Tivoli Monitoring for *Tivoli Netcool/OMNIbus Agent User's Guide*.

Nombre de connexions vers le serveur ObjectServer

Pour surveiller le nombre de connexions vers le serveur ObjectServer, activez le groupe de déclencheurs stats_triggers ainsi que l'automatisation statistics_gather. Les données regroupées par ce groupe de déclencheurs et cette automatisation sont écrites périodiquement dans la table master.stats, dans la colonne NumClients. Vous avez aussi la possibilité d'exécuter un

comptage manuel du nombre de lignes dans la table catalog.connections. Le nombre maximal de connexions possibles au serveur ObjectServer est de 1024. Lorsque le nombre maximal de connexions est atteint, les nouvelles connexions sont refusées. Le refus d'une connexion peut entraîner une perte temporaire de l'accès aux données ou aux entrées de sondes ou de passerelles. Le nombre maximum de connexions est 1024. Le nombre maximum de connexions autorisées est déterminé par la propriété **Connections** du serveur ObjectServer, avec une valeur par défaut de 256. Le nombre de connexions varie en fonction de votre environnement et de l'utilisation que vous en faites. Cependant, ce nombre devrait rester stable quand on le compare d'un mois à l'autre. Si le nombre de connexions augmente au fil des semaines et que le taux de cette augmentation est important, cela peut indiquer la présence d'un problème.

Utilisation du magasin de mémoire
Pour surveiller le magasin de mémoire, examinez le contenu de la table catalog.memstores. Pour chaque ligne, comparez la valeur de la colonne UsedBytes avec les valeurs des colonnes SoftLimit et HardLimit. Les magasins de mémoire sont des conteneurs gérés par le serveur ObjectServer. Ils conservent les données et les tables du serveur ObjectServer en mémoire. La taille des magasins de mémoire est limitée et quand ils sont complets, ils n'autorisent aucune insertion de données supplémentaire. Par conséquent, assurez-vous que les magasins de mémoire ne se remplissent jamais entièrement. L'utilisation des magasins de mémoire varie en fonction de votre environnement et de l'utilisation que vous en faites. Toutefois, cette utilisation devrait rester stable quand on la compare d'un mois à l'autre. Si l'utilisation des magasins de mémoire augmente au fil des semaines et que le taux de cette augmentation est important, cela peut indiquer qu'un problème se développe.

Autres informations utiles

Pour surveiller le nombre de lignes dans la table alerts.status, la table alerts.details et la table alerts.journal, activez le groupe stats_triggers et l'automatisation statistics_gather. Pour activer les déclencheurs, utilisez la commande **nco_config** pour démarrer Netcool/OMNIbus Administrator, ou utilisez la commande ALTER TRIGGER GROUP. Les données collectées par ce groupe de déclencheurs et cette automatisation sont écrites périodiquement dans la table master.stats. L'intervalle par défaut est de 300 secondes ; cette valeur est configurable. Le tableau ci-dessous décrit la colonne dans laquelle le nombre de numéro de ligne est écrit :

Tableau 91. Colonnes dans lesquelles le nombre d'événements est écrit pour les tables du serveur ObjectServer

Table	Colonne dans laquelle le nombre est écrit
alerts.status	Comptage d'événements
alerts.details	DetailCount
alerts.journal	JournalCount

Plusieurs KPI décrits dans la liste précédente peuvent également être surveillés à l'aide d'agents IBM Tivoli Monitoring. La connaissance pratique de IBM Tivoli Monitoring est obligatoire. Pour plus d'informations sur la configuration d'agents IBM Tivoli Monitoring, consultez le centre de documentation *IBM Tivoli Monitoring* sur http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp.

Référence associée:

«Table master.stats», à la page 440
La table master.stats stocke des informations de temporisation sur les tables alerts.status, alerts.details et alerts.journal. Ces informations de temporisation sont recueillies si le groupe de déclencheurs stats_triggers est activé. Le groupe de déclencheurs stats_triggers est désactivé par défaut dans le fichier `automation.sql`.

Information associée:

IBM Tivoli Monitoring for Tivoli Netcool/OMNIbus Agent

Indicateurs clés de performances de la sonde

Les sondes peuvent être configurées pour générer des événements de pulsations ProbeWatch en tant que mécanisme de surveillance automatique aidant à surveiller les performances, à diagnostiquer les problèmes de performances et à mettre en évidence les goulots d'étranglement des performances avant qu'ils n'affectent le système.

Conseil : Pour plus d'informations sur la configuration d'agents IBM Tivoli Monitoring, consultez le centre de documentation *IBM Tivoli Monitoring* sur http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp

Les KPI suivants peuvent être établis pour surveiller l'état de santé des sondes :

Nombre d'événements reçus par une sonde au cours des *n* dernières secondes
Il est possible de déduire le nombre d'événements reçus par la sonde au cours des *n* dernières secondes à partir de la colonne NumEventsProcessed de la table master.probestats sous forme de valeur delta de la précédente valeur rapportée de chaque sonde. Le débit de la sonde génère du travail pour le serveur ObjectServer. Il est conseillé de vérifier un flot d'événement provenant d'une sonde en particulier car cela peut mettre en évidence un problème lié à la sonde, à son fichier de règles, ou aux unités ou applications placées sous la surveillance de cette sonde. Comparez la valeur actuelle avec les valeurs précédentes de ce KPI pour identifier un comportement anormal.

Utilisation de l'unité centrale d'une sonde
La mesure de l'utilisation de l'unité centrale d'une sonde est contenue dans la colonne CPUTimeSec de la table master.probestats. Un agent IBM Tivoli Monitoring installé sur l'ordinateur de la sonde peut également mesurer l'utilisation de l'unité centrale de la sonde. Les ressources de l'unité centrale sont limitées. Si le processus de la sonde utilise la capacité maximale de l'unité centrale, les événements sont mis en file d'attente dans la sonde jusqu'à ce qu'elle puisse les traiter. Par conséquent, les entrées de la sonde peuvent s'accumuler, ce qui pourrait soit causer des retards dans le traitement, soit, en fonction de la sonde, entraîner une perte de données. La charge des événements entrants et le traitement du fichier de règles contribuent à ce pourcentage.

Empreinte mémoire d'une sonde
L'empreinte mémoire d'une sonde figure dans la colonne ProbeMemory de la table master.probestats. Un agent IBM Tivoli Monitoring installé sur l'ordinateur de la sonde peut également mesurer l'utilisation de la mémoire de la sonde. La mémoire est une ressource limitée et la mémoire de la sonde ne devrait pas augmenter démesurément. L'utilisation de la mémoire d'un processus de la sonde doit être relativement stable, même si une certaine augmentation est à prévoir à mesure que des caches et de la mémoire tampon sont générées. L'empreinte mémoire d'une sonde

augmentera lorsque le premier signal SIGHUP sera envoyé à la sonde pour demander à cette dernière de relire son fichier de règles. Cette augmentation est prévisible puisque le nouveau fichier de règles sera lu et analysé syntaxiquement avant que la mémoire utilisée par le fichier de règles existant ne soit publiée. C'est indispensable pour que la sonde ait toujours un fichier de règles valide. Les signaux SIGHUP ultérieurs demandant la relecture des fichiers de règles ne devraient causer qu'une augmentation mineure de l'utilisation de la mémoire. L'utilisation des tableaux associatifs peut également contribuer à l'augmentation de l'utilisation de la mémoire de la sonde car ils contiennent les événements traités par le fichier de règles. L'empreinte mémoire de la sonde nco_p_mttrapd est caractéristique car elle maintient une large mémoire tampon pour les interruptions entrantes. Cela peut souvent constituer un facteur de croissance de la mémoire supérieure à 50Mo à mesure que les 2000 premières interruption sont reçues. Une fois que l'espace nécessaire est attribué à la mémoire tampon de la file d'attente des interruptions, l'utilisation de la mémoire devrait se stabiliser à la baisse. Les autres poussées de croissances de mémoire démesurées et injustifiées doivent être vérifiées.

Délai moyen de traitement des règles
Le délai moyen de traitement du fichier de règles figure dans la colonne AvgRulesFileTime de la table master.probestats.

La présence de défaillances dans le fichier de règles peut causer des retards dans le traitement d'événement. Le délai de traitement du fichier de règles est un des facteurs majeurs contribuant à la limitation maximale de la capacité de traitement d'une sonde. Si le traitement du fichier de règles prend, en moyenne, 5 000 microsecondes (millionièmes de seconde), alors la sonde sera capable de traiter seulement 200 événements par seconde au maximum.

Indicateurs clés des performances de la passerelle

Ces indicateurs clés de performances (KPI) permettent de surveiller les performances des passerelles.

Conseil : Pour plus d'informations sur la configuration d'agents IBM Tivoli Monitoring, consultez le centre de documentation *IBM Tivoli Monitoring* sur http://publib.boulder.ibm.com/infocenter/tivihelp/v15r1/index.jsp

Les KPI suivants peuvent être utilisés pour surveiller les performances de la passerelle :

Utilisation de l'unité centrale des processus de la passerelle
Un agent IBM Tivoli Monitoring installé sur l'ordinateur passerelle peut mesurer l'utilisation de l'unité centrale du ou des processus de la passerelle. Les ressources de l'unité centrale sont limitées. Si l'unité centrale est saturée à cause du processus de la passerelle, il est possible que la passerelle ne réussisse pas à traiter tous les événements et que le transfert des événements vers le système cible soit retardé.

Utilisation de la mémoire des processus de passerelle
Un agent IBM Tivoli Monitoring installé sur l'ordinateur passerelle peut mesurer l'utilisation de la mémoire du ou des processus de la passerelle. Les ressources de la mémoire sont limitées. Une croissance illimitée de la mémoire peut causer l'arrêt anormal du ou des processus. L'utilisation de la mémoire d'un processus de la passerelle doit être relativement stable,

même si une certaine augmentation est à prévoir à mesure que des caches sont générées et pendant que la passerelle stocke temporairement des données d'alerte à mesure que les données sont transmises entre le serveur ObjectServer et leur destination. Il est important de rechercher les causes d'une croissance illimitée de mémoire. Une croissance de mémoire dans la passerelle ObjectServer peut être due à un problème provenant du serveur ObjectServer.

Pour de plus amples informations sur IBM Tivoli Monitoring Agent for Tivoli Netcool/OMNIbus, voir IBM Tivoli Monitoring for Tivoli Netcool/OMNIbus Agent User's Guide.

Information associée:

IBM Tivoli Monitoring for Tivoli Netcool/OMNIbus Agent

Valeurs recommandées pour l'optimisation des performances

Utilisez les valeurs recommandées pour faciliter la configuration de votre système Tivoli Netcool/OMNIbus et obtenir des performances optimales. Parcourez les étapes pour vous aider à identifier et à résoudre les modifications qui ont un effet négatif sur les performances et à ajuster les performances.

Pour plus d'informations sur le réglage des performances de l'interface graphique Web, voir *Guide d'administration et d'utilisation de l'interface graphique Web d'IBM Tivoli Netcool/OMNIbus* .

Exécution du serveur ObjectServer avec le profilage activé

Utilisez le profilage pour mesurer la durée de l'exécution des requêtes SQL sur le serveur ObjectServer et identifier les connexions client qui utilisent des ressources excessives.

Pour activer le profilage sur le serveur ObjectServer, procédez comme suit :

1. Assurez-vous que la propriété **Profile** est définie sur TRUE. Il s'agit de la valeur par défaut.
2. Utilisez la propriété **ProfileStatsInterval** pour spécifier un intervalle auquel les informations de profilage sont inscrites dans le fichier journal de profil. Un intervalle par défaut de 60 secondes est utilisé si vous ne modifiez pas cette valeur.
3. Assurez-vous que le groupe de déclencheurs profiler_triggers et ses déclencheurs (profiler_group_report, profiler_report, profiler_toggle) sont activés pour la consignation du profil.

Les informations de temporisation pour exécuter des commandes SQL à partir des connexions client sont consignées dans la table catalog.profiles. Netcool/OMNIbus Administrator vous permet d'afficher des détails enregistrés dans la table catalog.profiles. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Système**, puis cliquez sur **Databases (Bases de données)**. Vous pouvez utiliser l'onglet **Data View (Vue de données)** du panneau Databases, Tables and Columns (Bases de données, tables et colonnes) pour visualiser les données de table et utiliser l'onglet **Column Definitions (Définitions de colonnes)** pour visualiser les informations détaillées sur les colonnes de la table.

Les statistiques du profil sont également consignées dans un fichier journal de profil, $NCHOME/omnibus/log/*nom de serveur*_profiler_report.log*n*, où *nom de serveur* représente le nom du serveur ObjectServer et *n* est un nombre. Le fichier

journal du profil affiche la décomposition du temps passé pour chaque connexion client et le temps total passé par type de client pour chaque période de granularité (définie par la propriété **Granularity**). Chaque client présenté dans le fichier journal est identifié par un nom standard par défaut (par exemple, GATEWAY ou PROBE) et l'hôte sur lequel le client s'exécute. Le fichier journal de profil permet d'analyser comment le serveur ObjectServer a passé son temps pendant chaque période de granularité et de calculer le pourcentage de temps utilisé. Par exemple, si la période de granularité est définie sur 60 secondes et que la durée totale passée pour toutes les connexions pendant une période particulière était de 30 secondes, vous pouvez calculer que le serveur ObjectServer a passé 50 % de son temps disponible à exécuter des commandes SQL à partir de connexions client.

Le travail complété au cours d'une période de rapport correspond à une sortie dans une ligne récapitulative pour chaque période de granularité. Les informations présentées dans la ligne récapitulative s'affiche au format suivant : Durée totale au cours de la période de rapport (période de profilage) : durée totale de tous les clients. La durée totale de tous les clients peut être plus longue que la période de profilage à cause de la nature à unités d'exécutions multiples du serveur ObjectServer. Cela est particulièrement vrai pour les systèmes dotés de plusieurs unités centrales. Si la période de profilage est plus longue que la période de profilage configurée, cela signifie que le serveur ObjectServer est trop occupé pour générer un rapport sur la durée de profilage. Cela peut également signifie que le serveur ObjectServer est surchargé. Si la durée totale de tous les clients est plus longue que la période de profilage, cela signifie que la charge du système est trop faible, mais cela n'indique pas nécessairement la présence d'un problème.

Voici un exemple de sortie enregistrée dans un fichier journal de profil pour une période de granularité :

```
[1]  Mon Oct 12 17:39:46 2009: Individual user profiles:
[2]  Mon Oct 12 17:39:46 2009: 'Administrator' (uid = 0) time on adminhost: 0.000000s
[3]  Mon Oct 12 17:39:46 2009: 'isql' (uid = 0) time on omnihost1.ibm.com: 3.770000s
[4]  Mon Oct 12 17:39:46 2009: 'PROBE' (uid = 0) time on probehost.ibm.com: 5.010000s
[5]  Mon Oct 12 17:39:46 2009: 'e@c0B4D@c0142:11.0' (uid = 0) time on omnihost1.ibm.com: 10.010000s
[6]  Mon Oct 12 17:39:46 2009: 'c@xxxxx@xxxxx:11.0' (uid = 45) time on omnihost1.ibm.com: 0.000000s
[7]  Mon Oct 12 17:39:46 2009: 'e@c0B4D@c0142:11.0' (uid = 45) time on omnihost1.ibm.com: 9.870000s
[8]  Mon Oct 12 17:39:46 2009: 'c@xxxxx@xxxxx:11.0' (uid = 55) time on omnihost1.ibm.com: 0.000000s
[9]  Mon Oct 12 17:39:46 2009: 'e@c0B4D@c0142:11.0' (uid = 55) time on omnihost1.ibm.com: 6.020000s
[10] Mon Oct 12 17:39:46 2009: 'GATEWAY' (uid = 0) time on omnihost1.ibm.com: 0.270000s
[11] Mon Oct 12 17:39:46 2009: 'GATEWAY' (uid = 0) time on omnihost1.ibm.com: 0.000000s
[12] Mon Oct 12 17:39:46 2009: 'PROBE' (uid = 0) time on omnihost1.ibm.com: 3.010000s
[13] Mon Oct 12 17:39:46 2009: Grouped user profiles:
[14] Mon Oct 12 17:39:46 2009: Execution time for all connections whose application name is 'PROBE': 8.020000s
[15] Mon Oct 12 17:39:46 2009: Execution time for all connections whose application name is 'GATEWAY': 0.270000s
[16] Mon Oct 12 17:39:46 2009: Execution time for all connections whose application name is 'c@xxxxx@xxxxx:11.0': 0.000000s
[17] Mon Oct 12 17:39:46 2009: Execution time for all connections whose application name is 'e@c0B4D@c0142:11.0': 25.93000s
[18] Mon Oct 12 17:39:46 2009: Execution time for all connections whose application name is 'isql': 3.77000s
[19] Mon Oct 12 17:39:46 2009: Execution time for all connections whose application name is 'Administrator': 0.000000s
[20] Mon Oct 12 17:39:46 2009: Total time in the report period (59.275782s): 29.980000s
```

Les numéros de ligne sont inclus dans la sortie ci-dessus pour vous aider à décrire les entrées :

- Ligne [1] : introduit une liste de clients individuels connectés au serveur ObjectServer.
- Ligne [2] : présente le nom d'application du client connecté (`Administrator`), l'utilisateur associé pour ce client (ID utilisateur `0`), l'ordinateur hôte (`adminhost`) et la durée d'exécution du client durant la dernière période de profilage (`0.000000s`).
- Ligne [13] : introduit une liste qui présente le temps agrégé pour tous les clients du même type.

- Ligne [14] : explique que les deux sondes connectées ont utilisé une durée combinée de 8,02 secondes.
- Ligne [17] : explique que les listes d'événement ont utilisé 25 secondes. Songez à rechercher les temps individuels pour voir quelle liste d'événements utilise le plus de temps.
- Ligne [20] : Indique que la période de profilage dure 59,27 secondes et que la durée totale de tous les clients est de 29,98 secondes. La période de profilage correspond plus ou moins à la période de profilage configurée de 60 secondes ; cette estimation est valable si le système n'est pas surchargé.

Analysez les statistiques de profilage dans le fichier journal et la table de base de données pour identifier les clients qui utilisent le plus de temps et en déterminer la raison :

- Déterminez si toutes les connexions client sont nécessaires, et supprimez toutes les connexions client redondantes. Par exemple, les listes d'événements qui restent connectées après le départ des opérateurs.
- Si un client de la liste d'événements de bureau ou de l'interface graphique Web utilise beaucoup de temps, mettez en évidence les filtres utilisés par ce client. Analysez les filtres pour déterminer le nombre et la complexité des requêtes individuelles afin de les rendre plus efficaces.
- Si le client est une sonde, une dégradation des performances peut être due à des fichiers de règles mal écrits qui permettent le réacheminement d'événements inutiles vers le serveur ObjectServer, à la quantité d'informations détaillées envoyées par événement ou aux rafales d'événements.
- Augmentez la période de granularité du serveur ObjectServer pour atténuer les effets des charges clients importantes. Cette action ralentit le débit auquel le serveur ObjectServer envoie des diffusions IDUC à ses clients et peut améliorer les performances du système. Toutefois, les événements entrants prennent plus de temps pour atteindre les clients, particulièrement si le serveur ObjectServer fait partie d'une architecture à plusieurs niveaux.

Référence associée:

«Propriétés et options de ligne de commande du serveur ObjectServer», à la page 4
Utilisez les propriétés ou les options de ligne de commande ObjectServer pour configurer les paramètres du serveur ObjectServer. Pour éviter les erreurs, ajoutez autant de propriétés que possible au fichier de propriétés plutôt que des options de ligne de commande. D'autres utilitaires sont fournis que vous pouvez utiliser pour chiffrer les valeurs de propriété.

«Table catalog.profiles», à la page 438
La table catalog.profiles contient des informations de temporisation pour les commandes SQL en cours d'exécution à partir des connexions client.

Collecte des informations statistiques sur les déclencheurs

Les informations de temporisation relatives aux déclencheurs, y compris le nombre de fois que le déclencheur a été mis au premier plan et le nombre de fois où le déclencheur a été exécuté, sont enregistrées dans la table catalog.trigger_stats.

Pour collecter les informations statistiques sur les déclencheurs, procédez comme suit :

1. Assurez-vous que la propriété **Auto.Enabled** du serveur ObjectServer est définie sur TRUE. Il s'agit du paramètre par défaut. Il permet d'activer le système d'automatisation.

2. Utilisez la propriété **Auto.StatsInterval** pour contrôler la fréquence à laquelle le système d'automatisation collecte et stocke les informations statistiques dans la table catalog.trigger_stats. Un intervalle par défaut de 60 secondes est utilisé si vous ne modifiez pas cette valeur.
3. Assurez-vous que le groupe de déclencheurs trigger_stat_reports et le déclencheur trigger_stats_report sont activés.

Netcool/OMNIbus Administrator vous permet d'afficher des détails enregistrés dans la table catalog.trigger_stats. Dans la fenêtre Netcool/OMNIbus Administrator, sélectionnez le bouton de menu **Système**, puis cliquez sur **Databases (Bases de données)**. Vous pouvez utiliser l'onglet **Data View (Vue de données)** du panneau Databases, Tables and Columns (Bases de données, tables et colonnes) pour visualiser les données de table et utiliser l'onglet **Column Definitions (Définitions de colonnes)** pour visualiser les informations détaillées sur les colonnes de la table.

Les statistiques du profil sont également consignées dans le fichier `$NCHOME/omnibus/log/`*`nom de serveur`*`_trigger_stats.log`*`n`* où *nom de serveur* représente le nom du serveur ObjectServer et *n* est un nombre. Le fichier journal des statistiques du déclencheur affiche la durée utilisée par chaque déclencheur dans la dernière période de profilage. Utilisez ce fichier journal pour déboguer l'automatisation et déterminer les déclencheurs qui sont lents en raison de requêtes SQL qui s'exécutent lentement. Voici un exemple de sortie enregistrée dans un fichier journal de statistiques de déclencheur :

```
[1]  Mon Oct 12 18:03:56 2009: Trigger Profile Report
[2]  Mon Oct 12 18:03:56 2009: Trigger Group 'compatibility_triggers'
[3]  Mon Oct 12 18:03:56 2009: Trigger Group 'system_watch'
[4]  Mon Oct 12 18:03:56 2009:     Trigger time for 'system_watch_shutdown': 0.000000s
[5]  Mon Oct 12 18:03:56 2009:     Trigger time for 'system_watch_startup': 0.000000s
[6]  Mon Oct 12 18:03:56 2009: Trigger Group 'sae'
[7]  Mon Oct 12 18:03:56 2009:     Trigger time for 'update_service_affecting_events': 0.006790s
[8]  Mon Oct 12 18:03:56 2009: Trigger Group 'default_triggers'
[9]  Mon Oct 12 18:03:56 2009:     Trigger time for 'deduplication': 0.341918s
[10] Mon Oct 12 18:03:56 2009:     Trigger time for 'deduplication_eval': 0.092659s
[11] Mon Oct 12 18:03:56 2009:     Trigger time for 'service_update': 0.000000s
[12] Mon Oct 12 18:03:56 2009:     Trigger time for 'clean_journal_table': 0.000172s
[13] Mon Oct 12 18:03:56 2009:     Trigger time for 'service_insert': 0.000000s
[14] Mon Oct 12 18:03:56 2009:     Trigger time for 'service_reinsert': 0.000000s
[15] Mon Oct 12 18:03:56 2009:     Trigger time for 'clean_details_table': 0.000083s
[16] Mon Oct 12 18:03:56 2009:     Trigger time for 'state_change': 0.075508s
[17] Mon Oct 12 18:03:56 2009:     Trigger time for 'deduplication_copy': 0.022087s
[18] Mon Oct 12 18:03:56 2009:     Trigger time for 'new_row': 0.002637s
[19] Mon Oct 12 18:03:56 2009:     Trigger time for 'deduplicate_details': 0.000000s
[20] Mon Oct 12 18:03:56 2009: Trigger Group 'connection_watch'
[21] Mon Oct 12 18:03:56 2009:     Trigger time for 'connection_watch_connect': 0.000000s
[22] Mon Oct 12 18:03:56 2009:     Trigger time for 'connection_watch_disconnect': 0.000000s
[23] Mon Oct 12 18:03:56 2009: Trigger Group 'primary_only'
[24] Mon Oct 12 18:03:56 2009:     Trigger time for 'generic_clear': 5.879707s
[25] Mon Oct 12 18:03:56 2009:     Trigger time for 'expire': 0.008233s
[26] Mon Oct 12 18:03:56 2009:     Trigger time for 'delete_clears': 0.007219s
[27] Mon Oct 12 18:03:56 2009:     Trigger time for 'enrich_and_correlate': 23.007219s
[28] Mon Oct 12 18:03:56 2009: Trigger Group 'security_watch'
[29] Mon Oct 12 18:03:56 2009:     Trigger time for 'disable_user': 0.000000s
[30] Mon Oct 12 18:03:56 2009:     Trigger time for 'reset_user': 0.000000s
[31] Mon Oct 12 18:03:56 2009:     Trigger time for 'security_watch_security_failure': 0.000000s
[32] Mon Oct 12 18:03:56 2009: Trigger Group 'profiler_triggers'
[33] Mon Oct 12 18:03:56 2009:     Trigger time for 'profiler_group_report': 0.065094s
[34] Mon Oct 12 18:03:56 2009:     Trigger time for 'profiler_report': 0.087705s
[35] Mon Oct 12 18:03:56 2009:     Trigger time for 'profiler_toggle': 0.000000s
[36] Mon Oct 12 18:03:56 2009: Trigger Group 'trigger_stat_reports'
[37] Mon Oct 12 18:03:56 2009:     Trigger time for 'trigger_stats_report': 0.198813s
[38] Mon Oct 12 18:03:56 2009: Trigger Group 'iduc_triggers'
[39] Mon Oct 12 18:03:56 2009:     Trigger time for 'disconnect_iduc_missed': 0.000000s
[40] Mon Oct 12 18:03:56 2009:     Trigger time for 'iduc_stats_update': 0.000949s
[41] Mon Oct 12 18:03:56 2009:     Trigger time for 'iduc_messages_tblclean': 0.000089s
[42] Mon Oct 12 18:03:56 2009:     Trigger time for 'deduplicate_iduc_stats': 0.000000s
[43] Mon Oct 12 18:03:56 2009:     Trigger time for 'iduc_stats_insert': 0.000000s
[44] Mon Oct 12 18:03:56 2009: Trigger Group 'automatic_backup_system'
[45] Mon Oct 12 18:03:56 2009:     Trigger time for 'backup_succeeded': 0.000000s
```

```
[46] Mon Oct 12 18:03:56 2009:     Trigger time for 'backup_failed': 0.000000s
[47] Mon Oct 12 18:03:56 2009:     Trigger time for 'backup_state_integrity': 0.000000s
[48] Mon Oct 12 18:03:56 2009: Trigger Group 'gateway_triggers'
[49] Mon Oct 12 18:03:56 2009:     Trigger time for 'resync_finished': 0.000000s
[50] Mon Oct 12 18:03:56 2009: Time for all triggers in report period (60s): 29.789663s
```

Les numéros de ligne sont inclus dans la sortie ci-dessus pour vous aider à décrire les entrées :

- Ligne [1] : indique le début du rapport et l'horodatage de génération du rapport.
- Ligne [2] : indique que le rapport est ventilé par groupes de déclencheurs. Dans ce cas, le groupe de déclencheurs `compatibility_triggers` ne contient pas de déclencheurs activés.
- Ligne [3] : affiche `system_watch` comme premier groupe de déclencheurs contenant des déclencheurs activés.
- Ligne [4] : affiche le nom du déclencheur et la durée (en secondes) utilisée par le déclencheur lors de la dernière période de profilage.
- Ligne [27] : indique une durée excessive pour le déclencheur. Si cette occurrence est habituelle, il convient d'effectuer plus de recherches sur ce déclencheur. Par exemple, les questions suivantes peuvent s'appliquer : Existe-t-il des analyses de tables dans un déclencheur de base de données ? Utilisez-vous des analyses imbriquées ? Un index peut-il être utilisé pour réduire les analyses ? La durée est-elle directement liée au nombre d'événements traités par le système ? Le déclencheur utilise-t-il une clause EVALUATE qui pourrait être remplacée par une clause FOR EACH, exploitant directement la table à l'aide de la clause ACTION ?
- Ligne [50] : affiche la ligne récapitulative en tant que première entrée du rapport. Cette ligne présente la durée en secondes depuis l'exécution du dernier rapport et la durée totale utilisée par les déclencheurs dans la période de rapport. Dans cet exemple, 29 des 60 secondes est un pourcentage élevé. Des recherches supplémentaires peuvent par conséquent être nécessaires pour en déterminer la cause, en particulier si cette valeur est une occurrence habituelle.

Analysez les statistiques de déclencheur dans les fichiers journaux et les tables de base de données pour déterminer si une charge de travail peut causer une dégradation des performances :

- Si un déclencheur utilise la majorité de la période de granularité, recherchez-en la cause.
- Consultez vos automatisations du serveur ObjectServer personnalisées pour déterminer leur efficacité et réduire la charge de travail sur le serveur ObjectServer.
- Veillez à ce que le temps d'exécution du déclencheur soit maintenu à un minimum car aucune autre écriture ne peut avoir lieu lors de l'exécution d'un déclencheur.

Conseil : Lors de l'identification et de la résolution des problèmes liés aux performances, les fichiers journaux du profil et des statistiques de déclencheur sont le premier endroit où débuter vos recherches. Généralement, si la durée totale combinée des clients et des déclencheurs est constamment supérieure à 60 secondes (la période de granularité par défaut), vous devez effectuer certaines actions. Diverses métriques du système d'exploitation peuvent également vous être utiles pour identifier si un système est sous contrainte. Les métriques clés sont l'utilisation de l'unité centrale et la taille des processus Tivoli Netcool/OMNIbus.

Référence associée:

«Table catalog.trigger_stats», à la page 441
La table catalog.trigger_stats stocke des informations de temporisation sur les déclencheurs, y compris le nombre de fois où le déclencheur a été mis au premier plan et le nombre de fois où il a été mis en application. Ces statistiques sont recueillies sauf si le système d'automatisation est désactivé, c'est-à-dire, sauf si l'option de ligne de commande `-autoenabled` est définie sur `FALSE`.

«Automatisations Tivoli Netcool/OMNIbus standard», à la page 196
Un ensemble d'automatisations standard est inclus à Tivoli Netcool/OMNIbus. Ces automatisations sont créées lors de l'initialisation de la base de données.

Vérification et modification de l'architecture du système

Les personnalisations à plusieurs niveaux fournies avec Tivoli Netcool/OMNIbus vous permettent de déployer votre installation dans une architecture à un, deux ou trois niveaux, dans laquelle les composants se trouvent dans les couches de collecte, d'agrégation et d'affichage.

Au minimum, configurez une paire virtuelle de serveurs ObjectServer de reprise en ligne dans la couche d'agrégation, puis ajoutez des composants de collecte ou d'affichage, le cas échéant.

Si le trafic entrant de la sonde provoque des problèmes, songez à implémenter une couche de collecte (si elle n'est pas encore installée) avec des serveurs ObjectServer de collecte dédiés à la gestion des événements entrants avant de les transmettre à la couche d'agrégation.

Si le trafic du client d'affichage provoque des problèmes, notamment des temps de réponse lents pour les utilisateurs, songez à implémenter une couche d'affichage (si elle n'est pas encore installée) avec des serveurs ObjectServer d'affichage dédiés à la gestion des demandes client.

Si le trafic entrant de la sonde et le trafic du client d'affichage provoquent des problèmes, songez à implémenter l'architecture à trois niveaux avec une couche d'affichage, une couche d'agrégation et une couche d'affichage.

Configurez également la haute disponibilité sur Tivoli Netcool/OMNIbus afin de réduire la durée de synchronisation entre la paire de serveurs ObjectServer de reprise en ligne ainsi que la perte d'événements, et d'améliorer l'intégrité des données.

Activation du groupe de déclencheurs stats_triggers

Dans la configuration par défaut du serveur ObjectServer, le groupe stats_triggers utilise des déclencheurs pour regrouper plusieurs statistiques et métriques. Vous pouvez activer le groupe de déclencheurs dans un environnement de production.

Les déclencheurs constituant le groupe de déclencheurs stats_triggers comptent le nombre d'insertions dans les tables alerts.status, alerts.details et alerts.journal, ainsi que le nombre de dédoublonnages. Le résultat de ces comptages sont conservés dans la table master.activity_stats et sont agrégés par le déclencheur statistics_gather.

Pour surveiller le nombre de lignes dans la table alerts.status, la table alerts.details et la table alerts.journal, activez le groupe stats_triggers et l'automatisation statistics_gather. Pour activer les déclencheurs, utilisez la commande **nco_config** pour démarrer Netcool/OMNIbus Administrator, ou utilisez la commande ALTER TRIGGER GROUP. Les données collectées par ce groupe de déclencheurs et cette

automatisation sont écrites périodiquement dans la table master.stats. L'intervalle par défaut est de 300 secondes ; cette valeur est configurable. Le tableau ci-dessous décrit la colonne dans laquelle le nombre de numéro de ligne est écrit :

Tableau 92. Colonnes dans lesquelles le nombre d'événements est écrit pour les tables du serveur ObjectServer

Table	Colonne dans laquelle le nombre est écrit
alerts.status	Comptage d'événements
alerts.details	DetailCount
alerts.journal	JournalCount

Concepts associés:

Chapitre 4, «Utilisation de Netcool/OMNIbus Administrator pour configurer les serveurs ObjectServer», à la page 69
Le serveur ObjectServer stocke, gère et traite les données d'alerte et de statut collectées par des applications externes telles que des sondes et des passerelles. Vous pouvez utiliser Netcool/OMNIbus Administrator pour configurer vos objets du serveur ObjectServer et configurer le contrôle du processus.

Référence associée:

«ALTER TRIGGER GROUP», à la page 359
La commande `ALTER TRIGGER GROUP` permet d'activer ou de désactiver un groupe de déclencheurs existant.

«Table master.stats», à la page 440
La table master.stats stocke des informations de temporisation sur les tables alerts.status, alerts.details et alerts.journal. Ces informations de temporisation sont recueillies si le groupe de déclencheurs stats_triggers est activé. Le groupe de déclencheurs stats_triggers est désactivé par défaut dans le fichier `automation.sql`.

Vérification et modification des fichiers de configuration des sondes

Vérifiez attentivement le fichier de propriétés de la sonde et les paramètres du fichier de règles afin de vous assurer que ni l'un ni l'autre n'introduisent pas de baisse de l'efficacité dans le traitement des événements entrants.

En règle générale, les deux fichiers de configuration doivent rester aussi simples que possible. Si vous prévoyez un débit d'événements élevé, songez à utiliser plusieurs sondes pour collecter les événements entrants. Il est très probable que cela réduise la lenteur de la sonde et le nombre d'événements supprimés.

Configuration de la détection du flot d'événements

Configurez vos sondes pour détecter une condition de rafale d'événements ou d'autres débits d'événements anormaux, et effectuer des actions correctives.

Vous trouverez des exemples de fichiers de règles dans le répertoire `$NCHOME/omnibus/extensions/eventflood`, que vous pouvez utiliser pour cette configuration.

Gestion du volume d'informations dans la table alerts.details

Lorsqu'un volume élevé d'informations d'alerte est stocké dans la table alerts.details, les performances du serveur ObjectServer se détériorent de manière significative.

Les instructions suivantes permettent de gérer le volume d'informations stockées dans cette table.

- Vérifiez que l'automatisation clean_details_table est activée sur vos serveurs ObjectServer maître et de sauvegarde. Cette automatisation exécute des tâches de nettoyage dans la table alerts.details et supprime toutes les entrées non détectées dans la table alerts.status.
- N'utilisez pas `details($*)` dans les fichiers de règles de sonde pour ajouter des informations d'alerte à la table alerts.details. Pour chaque alerte de la table alerts.status, plusieurs lignes peuvent être ajoutées à la table alerts.details car la commande `details($*)` du fichier de règles enregistre chaque jeton comme une ligne. Après avoir utilisé la commande `details($*)` durant de longues périodes, les tables du serveur ObjectServer deviennent très volumineuses et les performances du serveur ObjectServer s'en ressentent. N'utilisez la commande `details($*)` que lorsque vous déboguez ou écrivez des fichiers de règles.

  Si vous devez ajouter des informations complémentaires dans l'alerte, utilisez l'instruction `details` pour ajouter des éléments spécifiques ou une expression régulière pour extraire des éléments spécifiques des détails. Par exemple, `details($a,$b)` ajoute les éléments $a et $b à la table alerts.details.

  Notez également que chaque entrée `details` requiert une instruction INSERT distincte. Pour un événement comprenant 20 entrées `details`, 21 insertions seront effectuées dans le serveur ObjectServer. Un événement sans détails constitue cependant une seule insertion.
- A partir de l'interface interactive SQL, utilisez la commande DELETE SQL pour effacer tous les enregistrements de la table alerts.details. Par exemple :

  ```
  delete from alerts.details;
  ```

  Remarque : La suppression manuelle de données de la table alerts.details est une solution temporaire car le nombre de détails atteindra à nouveau à un volume élevé si les instructions précédentes de gestion de la table alerts.details ne sont pas respectées. Au lieu d'utiliser la table alerts.details, essayez d'utiliser la colonne ExtendedAttr avec la fonction **nvp_add rules** du fichier de règles et la fonction **nvp_get** ou **nvp_set** pour stocker les zones placées dans les détails.

Utilisation d'un agent de surveillance pour contrôler et gérer les ressources Tivoli Netcool/OMNIbus

Un agent IBM Tivoli Monitoring est disponible pour contrôler l'état et les performances, les déclencheurs d'automatisation et l'activité et la distribution des événements de Tivoli Netcool/OMNIbus. Cet agent de surveillance comprend également un ensemble d'automatisations qui ajoutent de l'instrumentation au serveur ObjectServer.

L'agent IBM Tivoli Monitoring for Tivoli Netcool/OMNIbus est disponible en téléchargement dans le module d'installation de base Tivoli Netcool/OMNIbus. Le logiciel IBM Tivoli Monitoring prérequis est également disponible en téléchargement.

Pour plus d'informations sur l'installation et la configuration de l'agent de surveillance, accédez au centre de documentation IBM Tivoli Network

Management à l'adresse suivante : http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp. Dans le panneau de navigation de gauche, localisez et développez le noeud de version *Tivoli Netcool/OMNIbus* pertinent, puis accédez au noeud *IBM Tivoli Monitoring for Tivoli Netcool/OMNIbus*. Une version PDF est également disponible dans le noeud *Documentations PDF*.

Une fois l'agent de surveillance configuré, surveillez les ressources système, puis menez les actions correctives nécessaires pour améliorer les performances ou la disponibilité.

Vérification et modification de vos requêtes SQL, puis création d'une sélection d'index bien conçus et efficaces

Il est utile de comprendre comment les requêtes SQL sont optimisées pour générer des requêtes SQL efficaces. Consultez les requêtes SQL existantes et modifiez-les pour tirer parti de l'optimisation et de l'indexation.

Parcourez les instructions relatives à l'indexation et aux requêtes SQL fournies.

Lors de la conception et de la création d'index, il est également utile de comprendre les caractéristiques des tables et colonnes de base de données du serveur ObjectServer afin de déterminer les colonnes indexées pouvant nécessiter des performances optimisées. Les comparaisons d'entiers sont plus rapides que les comparaisons de chaînes. Si vos données d'événement contiennent des chaînes constantes, utilisez des entiers pour représenter les chaînes de vos fichiers de règles, de vos automatisations et de vos filtres, puis utilisez des conversions pour afficher les chaînes pour les utilisateurs. Par exemple, la colonne Class du serveur ObjectServer est un type de données entier mais s'affiche en tant que chaîne dans les listes d'événements. Vous pouvez exécuter les actions suivantes pour transmettre des données d'événement en tant qu'entiers, qui sont ensuite affichés en tant que chaînes :

- Modifier vos fichiers de règles de sonde pour définir les valeurs entières qui sont mappées aux chaînes correspondantes.
- Modifier vos automatisations (et les règles Netcool/Impact, le cas échéant) pour utiliser les valeurs entières.
- Modifier vos filtres afin d'utiliser les valeurs entières dans les clauses WHERE.
- Ajouter des conversions mappant les valeurs entières aux valeurs de chaîne qui seront affichées pour les utilisateurs.

Vous pouvez utiliser la commande CREATE INDEX SQL pour créer des index et la commande DROP INDEX pour supprimer les index redondants. La table catalog.indexes contient des informations relatives aux index créés et stockés.

Après avoir modifié vos requêtes SQL et créé des index, définissez temporairement la propriété **MessageLevel** sur `debug` afin que la durée d'exécution des requêtes SQL individuelles et l'indexation des détails soient consignées. Définissez une période d'activité de traitement appropriée.

Examinez le fichier journal du serveur ObjectServer `$NCHOME/omnibus/log/`*`nom_serveur`*`.log` pour :

- Déterminer les requêtes SQL qui ont subi un impact négatif sur leurs performances. Vérifiez également les requêtes trop longues ou exécutées à répétition.
- Déterminer les index utilisés.

- Analyser les temps de réponse des requêtes SQL fréquemment utilisées et déterminer les avantages significatifs ou marginaux.

Les étapes suivantes décrivent comment déterminer le temps nécessaire à l'exécution d'une requête SQL :

1. Recherchez un message de journal du serveur ObjectServer similaire à l'exemple ci-dessous :

   ```
   2010-03-19T14:39:27: Debug: D-OBJ-105-010: Client language command on
   connection ID N : [user1][isql][][hostname.ibm.com] [Instruction SQL].
   ```

   où *Instruction SQL* correspond à l'instruction SQL que vous voulez analyser.
2. Notez le numéro de l'ID connexion *N*.
3. Recherchez un message de journal du serveur ObjectServer similaire au suivant, qui contient les mêmes valeurs pour *N* :

   ```
   2010-03-19T14:39:27: Information: I-OBJ-104-016: Profiler timing submitted from
   connection ID N : durée en secondes
   ```

   où la *durée en secondes* correspond à la durée nécessaire à l'exécution de la requête SQL.

La sortie de journal ObjectServer suivante peut vous servir d'exemple :

```
2010-03-19T14:39:27: Debug: D-OBJ-105-010: Client language command on connection
 ID 1: [user1][isql][][hostname.ibm.com] [select * from catalog.indexes;]. 2010-03-
19T14:39:27: Information:
I-OBJ-104-016: Profiler timing submitted from connection ID 1: 0.000265
```

Vous pouvez constater qu'il aura fallu 0,000 265 secondes pour que la commande `select * from catalog.indexes` soit complète.

Remarque : Vous pouvez définir la consignation du serveur ObjectServer en mode débogage sans mettre le serveur ObjectServer hors ligne. Ce paramètre peut avoir un effet négatif sur les performances. Vous devez donc passer à la consignation en mode débogage une fois la collecte des données terminée.

Tâches associées:

«Consultation des résultats des optimisations de requête automatiques», à la page 333
Un ensemble de règles d'optimisation est appliqué automatiquement aux requêtes SQL pour déterminer la façon la plus efficace d'exécuter des requêtes. Consultez les résultats de cette optimisation pour savoir à quel niveau vous pouvez intervenir pour rendre les requêtes plus efficaces.

Référence associée:

«Table catalog.indexes», à la page 439
La table catalog.indexes stocke des informations sur les index, y compris la colonne et la table de base de données sur lesquelles l'index se base, et le type d'index.

«Instructions de requête SQL», à la page 333
Lorsqu'une instruction SQL est transmise au serveur ObjectServer, l'optimisation de requête et les plans de requête sont utilisés pour évaluer les méthodes disponibles d'accès aux/de modification des données et pour sélectionner la façon la plus efficace d'exécuter la requête.

«Instructions d'indexation», à la page 336
L'indexation peut affecter les performances de vos requêtes SQL. Sans indexation, une analyse de base de données complète est généralement effectuée lors de l'exécution d'une requête SQL. Utilisez l'indexation pour limiter le nombre de lignes à examiner.

«Valeurs recommandées lors de la création de déclencheurs», à la page 340
Lorsque vous créez ou modifiez des déclencheurs, vérifiez que ces derniers sont aussi efficaces que possible et qu'ils possèdent le plus faible temps d'exécution possible.

Suivi des tendances de performances à des intervalles réguliers

Au fur et à mesure, différents facteurs peuvent affecter les performances du serveur ObjectServer. Etablissez une ligne de base pour les performances et surveillez périodiquement les performances pour comparer les métriques avec celles précédemment collectées. Recherchez des différences significatives au-dessus et en dessous de la ligne de base et effectuez des réglages le cas échéant.

Surveillance automatique de Tivoli Netcool/OMNIbus

Vous pouvez surveiller une plage d'indicateurs clés de performances (KPI) au sein de l'infrastructure du serveur ObjectServer en activant l'extension de surveillance automatique de Tivoli Netcool/OMNIbus. Les données de performances sont signalées sous forme d'événements synthétiques et peuvent être affichées dans la liste d'événements ou dans un tableau de bord de l'interface graphique Web Netcool Health. Les données d'événement sont également consignées dans un fichier journal de surveillance automatique.

La surveillance automatique est mise en oeuvre sous la forme d'un ensemble de déclencheurs et d'une table master.sm_threshold qui stocke des informations de seuil pour les différents types d'événements de surveillance automatique. La table contient un ensemble de valeurs de seuil pour chaque indicateur clé de performances surveillé et définit le point de génération des événements ainsi que leur gravité. Les valeurs de seuil sont configurées par serveur ObjectServer. Différents serveurs ObjectServer peuvent utiliser des seuils différents, si nécessaire.

Deux types différents d'événement sont générés par la surveillance automatique de Tivoli Netcool/OMNIbus : les événements d'information et les événements d'alerte.

Remarque : Tous les événements de surveillance automatique, à l'exception des événements d'information de Classe supérieure issus d'une couche Collection multiniveau, comportent une valeur Classe de 99999. Vous pouvez utiliser cette valeur, dans un filtre approprié, pour afficher les événements provenant de n'importe quelle liste d'événements.

Evénements d'information

Les événements d'information sont générés pour signaler les valeurs de métrique en cours des indicateurs clés de performances surveillés. Les événements d'information ont systématiquement une gravité de 2 (Avertissement) et un type de 13 (Information). La zone Grade stocke la valeur de métrique signalée de sorte que la métrique puisse être suivie dans l'archive des événements historiques. Par défaut, les événements d'information sont toujours générés et toujours dédoublonnés. Ces deux comportements peuvent être modifiés dans la table master.sm_threshold.

Les événements d'information comportent un délai d'expiration (ExpireTime) de 30 secondes de plus que leur fréquence de génération. Par exemple, si un événement d'information est généré toutes les 60 seconds minutes, le délai ExpireTime est de 60 + 30 = 90 secondes. Si l'événement d'information est généré toutes les 300

seconds, le délai ExpireTime est de 300 + 30 = 330 secondes.

Evénements d'alerte

Les événements d'alerte sont générés chaque fois qu'un KPI surveillé dépasse un seuil prédéfini. Chaque indicateur KPI surveillé possède une ligne correspondante dans la table master.sm_threshold. Lors de l'exécution de chaque déclencheur de surveillance automatique, celui-ci lit, à partir de cette table, les données de seuil relatives à l'indicateur KPI associé.

Une fois que le déclencheur génère des événements d'information, il vérifie les valeurs de métrique par rapport aux seuils et crée un événement d'alerte supplémentaire si un seuil est dépassé. Chaque ligne de seuil stocke généralement un seuil de gravité 3, 4 et 5 et définit la gravité d'événement d'alerte selon l'emplacement de la valeur de la métrique. Par exemple, les seuils par défaut pour le nombre de lignes sur la table alerts.status sont 80000, 90000 et 100000 pour les niveaux de gravité 3, 4, et 5. Si le nombre de lignes était de 85 000, l'événement d'alerte serait créé avec une gravité de 3 (mineure). Si le nombre de lignes était de 105 000, l'événement d'alerte serait créé avec une gravité de 5 (critique).

A l'instar des événements d'information, la zone Grade est utilisée pour stocker la valeur de métrique objet d'un rapport. Les événements d'alerte sont systématiquement dédoublonnés afin d'éviter les situations où de nombreux événements d'alerte de surveillance automatique commencent à avoir une incidence sur les performances du serveur ObjectServer. Chaque fois qu'un événement d'alerte est dédoublonné, les zones Résumé, Gravité et Grade sont mises à jour pour refléter l'état.

Le délai d'expiration (ExpireTime) des événements d'alerte est paramétré sur 24 heures. Ce délai permet à un administrateur de vérifier les alertes pouvant se présenter au milieu de la nuit.

Chaque fois que la gravité d'un événement d'alerte change, une entrée de journal est ajoutée ; celle-ci enregistre l'heure, le changement de la valeur Gravité et la modification de la valeur de métrique. Un administrateur peut examiner l'historique de l'alerte d'événement, vérifier comment la métrique a évolué au fil du temps et mettre ces informations en corrélation avec d'autres métriques KPI pour la même période. L'administrateur peut ensuite déterminer le motif des violations de seuil et prendre des mesures correctives.

Fichier journal de surveillance automatique

Le fichier journal de surveillance automatique, *nom_ObjectServer*_selfmonitoring.log se trouve dans le répertoire suivant :

```
$NCHOME/omnibus/log
```

Toutes les 5 minutes, les nombres de lignes permanentes pour les tables alerts.status, alerts.journal et alerts.details sont consignées. Le nombre d'insertions au cours des 5 dernières minutes est également consigné. Exemple :

```
Mon Nov 25 14:55:08 2013: RowCounts: alerts.status: 36, alerts.journal: 0,
 alerts.details: 0
Mon Nov 25 15:00:08 2013: InsertCounts: Last 5 mins: alerts.status
 (inserts/deduplications): 86, alerts.journal (inserts): 0,
 alerts.details (inserts): 0
```

Les événements d'alerte sont également consignés. La consignation fournit aux administrateurs un autre moyen pour vérifier ce qui se produit (ou ce qui s'est produit) dans un serveur ObjectServer avant une panne. Si la panne rend le serveur ObjectServer inaccessible, un administrateur a un aperçu de ce qui se passe dans le serveur ObjectServer.

Référence associée:

«Tables de surveillance automatique», à la page 455
Les tables de surveillance automatique contiennent les données requises pour l'exécution de la fonction de surveillance automatique de Tivoli Netcool/OMNIbus.

Chapitre 8, «Optimisation des performances», à la page 309
Il est possible de mesurer les performances de Tivoli Netcool/OMNIbus en termes de temps de réponse, de débit et de disponibilité.

Installation du tableau de bord dans un environnement d'équilibrage de charge

Si vous exécutez l'interface graphique Web avec Tivoli Integrated Portal version 2.2.0.9, l'installation du tableau de bord Netcool Health dans un environnement d'équilibrage de charge requiert une configuration supplémentaire. Cette configuration n'est pas requise si vous exécutez l'interface graphique Web avec Tivoli Integrated Portal version 2.2.0.11 (ou suivante), ou avec Concentrateur des services d'application du tableau de bord version 3.1 (ou suivante).

Avant de commencer

Installez l'infrastructure de surveillance automatique côté serveur.

Pourquoi et quand exécuter cette tâche

Pour installer le tableau de bord Netcool Health dans un environnement d'équilibrage de charge, vous devez dissocier puis réassocier chaque noeud dans l'environnement.

Remarque : Pour éviter de longues durées d'immobilisation, assurez-vous que le premier noeud est corrigé et prêt pour être réassocié à l'environnement d'équilibrage de charge avant que le dernier noeud ne soit dissocié.

Remarque : Une fois le premier noeud dissocié, n'apportez pas à l'environnement d'équilibrage de charge des modifications autre que celles décrites dans cette procédure.

Procédure

1. Dissociez le premier noeud de l'environnement d'équilibrage de charge :
   a. Utilisez l'option `disjoin` pour supprimer le noeud.

      Pour remettre le noeud supprimé en mode de fonctionnement normal, redémarrez le serveur.
   b. Supprimez de la configuration en cluster les informations d'équilibrage de charge du noeud.
2. Installez le tableau de bord sur le premier noeud dissocié.
3. Vérifiez que le noeud dissocié corrigé fonctionne comme prévu.
4. Répétez les étapes suivantes pour chaque noeud restant, à l'exception du dernier noeud, dans l'environnement d'équilibrage de charge :

a. Utilisez l'option `disjoin` pour supprimer le noeud.
b. Supprimez de la configuration en cluster les informations d'équilibrage de charge du noeud.

5. Utilisez l'option `uninstall` pour supprimer le dernier noeud, puis supprimez ses informations d'équilibrage de charge de la configuration en cluster.
6. Réassociez le noeud que vous avez corrigé à l'étape 1 :
   a. Utilisez l'option `configHA` pour associer le noeud au cluster.
   b. Démarrez les opérations d'équilibrage de charge sur le noeud.
   c. Vérifiez que le tableau de bord Netcool Health fonctionne comme prévu sur le noeud.
7. Pour réassocier chaque noeud corrigé restant à l'environnement d'équilibrage de charge, procédez comme suit :
   a. Utilisez l'option `join` pour associer le noeud au cluster.
   b. Démarrez les opérations d'équilibrage de charge sur le noeud.
   c. Vérifiez que le tableau de bord Netcool Health fonctionne comme prévu sur le noeud.

Seuils de surveillance automatique

Les valeurs de seuil de surveillance automatique de Tivoli Netcool/OMNIbus sont stockées dans la table master.sm_thresholds. Vous pouvez modifier les valeurs de seuil à l'aide de l'interface interactive SQL (**nco_sql** ou **isql**) ou de Netcool/OMNIbus Administrator (**nco_config**).

Les valeurs de seuil par défaut sont calculées pour s'aligner sur les meilleures pratiques concernant les performances du serveur ObjectServer. Cependant, il peut être nécessaire de modifier les seuils dans les conditions suivantes :

- Les règles peuvent définir un nombre inférieur à la valeur par défaut du nombre total de journaux devant être présents à un moment donné.
- Le test de charge à l'aide d'un ensemble de déclencheurs et de clients connectés peut révéler qu'un serveur peut traiter un nombre d'événements résidents inférieur à la valeur par défaut. Dans ce cas, le seuil initial doit être déclenché lorsque le nombre total atteint une valeur inférieure.

Le tableau ci-dessous contient la liste des seuils de surveillance automatique, une description pour chacun d'eux et les valeurs de seuil par défaut pour la gravité de l'événement d'alerte résultant. Les événements d'alerte sont générés lorsque le seuil de gravité le plus faible (gravité 3) est dépassé. Lorsqu'un événement d'alerte est généré, la gravité de cet événement d'alerte est déterminée par la valeur de la métrique surveillée.

Les valeurs de gravité sont les points de seuil. Si la valeur de métrique est supérieure ou égale à la valeur de seuil, l'événement d'alerte comporte la gravité correspondante la plus élevée. Par exemple, si la métrique `sm_client_time_individual` a pour valeur 29, aucun événement d'alerte n'est généré. Si elle a pour valeur 35, un événement d'alerte de gravité 3 est généré. Si elle a pour valeur 51, un événement d'alerte de gravité 5 est généré.

Tableau 93. Seuils de surveillance automatique

Nom du seuil	Description	Seuils		
		Gravité 3	Gravité 4	Gravité 5
sm_client_time_ individual	Seuil du temps consommé par un client connecté individuel au cours d'une période de granularité ObjectServer unique. Tout client dont le déclenchement prend 30 secondes ou plus au cours d'une période de granularité génère un événement d'alerte.	30	40	50
sm_client_time_total	Seuil du temps consommé par tous les clients connectés au cours d'une période de granularité ObjectServer unique. Si tous les clients connectés consomment collectivement 40 secondes ou plus de temps ObjectServer au cours d'une période de granularité, un événement d'alerte est généré.	40	60	90
sm_connections	Valeurs de seuil du nombre de connexions disponibles restantes. Si le serveur ObjectServer contient au maximum 50 connexions disponibles, un événement d'alerte est généré.	50	30	10
sm_db_stats_ details_count	Seuil du nombre de lignes dans la table alerts.details. Si 3000 lignes ou plus sont présentes, un événement d'alerte est généré.	3000	10000	20000
sm_db_stats_ event_count	Seuil du nombre de lignes dans la table alerts.status. Si 80000 lignes ou plus sont présentes, un événement d'alerte est généré.	80000	90000	100000
sm_db_stats_ journal_count	Seuil du nombre de lignes dans la table alerts.journal. Si 20000 lignes ou plus sont présentes, un événement d'alerte est généré.	20000	35000	50000
sm_dbops_stats_ details_inserts	Seuil du nombre d'insertions dans la table alerts.details au cours de chaque période de statistiques ObjectServer de cinq minutes. Si 10000 lignes ou plus sont insérées dans une période donnée de cinq minutes, un événement d'alerte de gravité 3 est généré. Cette métrique ne génère jamais un événement de gravité 4 (Majeure) ou 5 (Critique) car il ne s'agit pas d'un indicateur de performances aussi critique que, par exemple, le délai d'exécution des déclencheurs. Toutefois, si d'autres indicateurs indiquent également des alertes, l'alerte créée par ce seuil fournit des informations d'expertise précieuses pour comprendre la situation globale.	10000	0	0
sm_dbops_stats_ journal_inserts	Seuil du nombre d'insertions dans la table alerts.journal au cours de chaque période de statistiques ObjectServer de cinq minutes. Si 10000 lignes ou plus sont insérées dans une période donnée de cinq minutes, un événement d'alerte de gravité 3 est généré. Cette métrique ne génère jamais un événement de gravité 4 (Majeure) ou 5 (Critique) car il ne s'agit pas d'un indicateur de performances aussi critique que, par exemple, le délai d'exécution des déclencheurs. Toutefois, si d'autres indicateurs indiquent également des alertes, l'alerte créée par ce seuil fournit des informations d'expertise précieuses pour comprendre la situation globale.	10000	0	0

Tableau 93. Seuils de surveillance automatique (suite)

Nom du seuil	Description	Seuils		
		Gravité 3	Gravité 4	Gravité 5
sm_dbops_stats_ status_inserts	Seuil du nombre d'insertions dans la table alerts.status au cours de chaque période de statistiques ObjectServer de cinq minutes. Si 10000 lignes ou plus sont insérées dans une période donnée de cinq minutes, un événement d'alerte de gravité 3 est généré. **Remarque :** Cette métrique inclut tant des insertions que des réinsertions. Cette métrique ne génère jamais un événement de gravité 4 (Majeure) ou 5 (Critique) car il ne s'agit pas d'un indicateur de performances aussi critique que, par exemple, le délai d'exécution des déclencheurs. Toutefois, si d'autres indicateurs indiquent également des alertes, l'alerte créée par ce seuil fournit des informations d'expertise précieuses pour comprendre la situation globale.	10000	0	0
sm_memstore	Seuil du pourcentage de l'allocation de mémoire de limite souple utilisée par le serveur ObjectServer. If au moins 75% de la limite souple est utilisée, un événement d'alerte est généré. Si plus de 95% de la limite souple est utilisée, le récapitulatif des événements d'alerte recommande d'augmenter la limite souple.	75	85	95
sm_time_to_display	Seuil du temps moyen mis par les événements pour atteindre les serveurs ObjectServer de la couche d'affichage. Si les événements mettent 60 secondes ou plus pour se propager au serveurs ObjectServer de la couche d'affichage, un événement d'alerte est généré. Si une couche d'affichage n'est pas présente dans votre déploiement, cette métrique n'est pas surveillée.	60	120	180
sm_top_classes	Seuil du nombre d'événements reçus dans la couche Agrégation, par classe, au cours de chaque période de statistiques ObjectServer de cinq minutes. Si 600 événements ou plus sont reçus pour n'importe quelle classe au cours d'une période donnée de cinq minutes, un événement d'alerte est généré. Ce seuil prend en compte tous les événements reçus pour une classe à partir de la couche Agrégation et la couche Collection, le cas échéant.	600	800	1000
sm_top_nodes	Seuil du nombre d'événements reçus dans la couche Agrégation, par noeud, au cours de chaque période de statistiques ObjectServer de cinq minutes. Si 100 événements ou plus sont reçus pour n'importe quel noeud au cours d'une période donnée de cinq minutes, un événement d'alerte est généré. Ce seuil prend en compte tous les événements reçus pour un noeud au niveau de la couche Agrégation et de la couche Collection, le cas échéant.	100	200	500
sm_top_probes	Seuil du nombre d'événements reçus dans la couche Agrégation ou Collection, par sonde, au cours de chaque période de statistiques ObjectServer de cinq minutes. Si 600 événements ou plus sont reçus de n'importe quelle sonde au cours d'une période donnée de cinq minutes, un événement d'alerte est généré. Ce seuil prend en compte toutes les sondes connectées à la couche Agrégation et à la couche Collection, le cas échéant.	600	800	1000

Tableau 93. Seuils de surveillance automatique (suite)

Nom du seuil	Description	Seuils		
		Gravité 3	Gravité 4	Gravité 5
sm_triggers_individual	Seuil relatif à n'importe quel déclencheur individuel au cours d'une période de granularité ObjectServer unique. Tout déclencheur dont le déclenchement prend 20 secondes ou plus au cours d'une période de granularité génère un événement d'alerte.	20	25	30
sm_triggers_total	Seuil relatif à tous les déclencheurs collectivement dans une période de granularité ObjectServer unique. Si le temps total des déclencheurs est de 50 secondes ou plus au cours d'une période de granularité, un événement d'alerte est généré.	50	70	90
sm_triggers_reporting_period	Seuil relatif à la période de rapport de profileur. La période de rapport doit toujours être située dans une fraction de seconde du temps de granularité (la granularité par défaut est de 60 secondes). Si la période de rapport est de 61 secondes ou plus, un événement d'alerte est généré. Si la période de rapport commence à augmenter, cela indique que le serveur ObjectServer est surchargé. Cela peut se produire pour plusieurs raisons, y compris un déclencheur à performances médiocres ou un temps d'interaction client excessif. Si la période de rapport est régulièrement supérieure à la période de granularité, les utilisateurs peuvent constater des retards dans la présentation des événements. Conjointement avec d'autres alertes qui sont susceptibles d'être générées dans ce cas, ces informations sont utiles pour les administrateurs lorsqu'ils déterminent l'origine de la surcharge.	61	70	90

Concepts associés:

«interface interactive SQL», à la page 229
Vous pouvez utiliser l'utilitaire de ligne de commande de l'interface interactive SQL fourni avec Tivoli Netcool/OMNIbus pour émettre des commandes SQL en direction d'un serveur ObjectServer.

Chapitre 4, «Utilisation de Netcool/OMNIbus Administrator pour configurer les serveurs ObjectServer», à la page 69
Le serveur ObjectServer stocke, gère et traite les données d'alerte et de statut collectées par des applications externes telles que des sondes et des passerelles. Vous pouvez utiliser Netcool/OMNIbus Administrator pour configurer vos objets du serveur ObjectServer et configurer le contrôle du processus.

Référence associée:

«Tables de surveillance automatique», à la page 455
Les tables de surveillance automatique contiennent les données requises pour l'exécution de la fonction de surveillance automatique de Tivoli Netcool/OMNIbus.

Instructions de requête SQL

Lorsqu'une instruction SQL est transmise au serveur ObjectServer, l'optimisation de requête et les plans de requête sont utilisés pour évaluer les méthodes disponibles d'accès aux/de modification des données et pour sélectionner la façon la plus efficace d'exécuter la requête.

L'évaluation détermine si les analyses d'index ou les clés primaires peuvent être utilisées ou si une analyse de table entière peut être effectuée à la place. Si des analyses d'index sont utilisées, les index sont automatiquement appliqués aux requêtes SQL qui référencent les colonnes indexées.

Consultation des résultats des optimisations de requête automatiques

Un ensemble de règles d'optimisation est appliqué automatiquement aux requêtes SQL pour déterminer la façon la plus efficace d'exécuter des requêtes. Consultez les résultats de cette optimisation pour savoir à quel niveau vous pouvez intervenir pour rendre les requêtes plus efficaces.

Pourquoi et quand exécuter cette tâche

L'optimisation de requête est activée par défaut. Les clauses WHERE présentes dans les instructions SQL suivantes sont optimisées :

- Opérations DML (SELECT, UPDATE, DELETE, GROUP BY et SELECT agrégée) : la clause WHERE est concaténée avec le filtre de restriction (le cas échéant), puis optimisée. Pour une vue, la clause WHERE est concaténée avec la clause WHERE de la vue.
- Boucle FOR EACH
- Instruction IF THEN ELSE
- Instruction EVALUATE (GROUP BY, SELECT et SELECT agrégée)

Les types de requête suivants ne sont pas optimisés :

- Une clause HAVING d'une commande GROUP BY
- La colonne d'un tableau

Procédure

Pour consulter les résultats de l'optimisation de requête :

1. Utilisez la propriété **MessageLevel** du serveur ObjectServer pour définir le niveau de consignation par `debug`. Les clauses optimisées et non optimisées sont consignées dans le fichier par défaut, `$NCHOME/omnibus/log/`*`nom de serveur`*`.log`, où *nom de serveur* est le nom du serveur ObjectServer.
2. Consultez les informations du fichier journal pour savoir quelles sont les requêtes réécrites par le serveur ObjectServer afin d'accroître leur efficacité et utilisez-les pour optimiser l'efficacité des requêtes SQL que vous écrivez.

Référence associée:

«Propriétés et options de ligne de commande du serveur ObjectServer», à la page 4
Utilisez les propriétés ou les options de ligne de commande ObjectServer pour configurer les paramètres du serveur ObjectServer. Pour éviter les erreurs, ajoutez autant de propriétés que possible au fichier de propriétés plutôt que des options de ligne de commande. D'autres utilitaires sont fournis que vous pouvez utiliser pour chiffrer les valeurs de propriété.

Règles d'optimisation pour les requêtes SQL

Lorsque l'optimisation de requête SQL survient, toutes les règles d'optimisation sont appliquées à la condition dans la requête. Les prédicats sont ensuite réorganisés.

L'optimisation AND et OR, et la réorganisation des prédicats sont appliquées aux requêtes SQL de la manière suivante.

- «optimisation AND»
- «optimisation OR»
- «Réorganisation des prédicats», à la page 335

optimisation AND

Si une clause WHERE se compose de plusieurs prédicats connectés par l'opérateur AND, l'optimisation survient uniquement si la même colonne et le même opérateur de comparaison sont utilisés dans chaque prédicat et que l'opérateur de comparaison est LIKE ou égal à (=).

Le format acceptable de l'optimisation AND est le suivant :

```
where opérateur_colonne_expr1 AND opérateur_colonne_expr2
AND opérateur_colonne_expr3...
```

Dans cette clause WHERE, la variable *opérateur* est LIKE ou = uniquement.

Ce format est optimisé dans une liste ALL comme suit :

```
where opérateur_colonne ALL (expr1, expr2, expr3, ...)
```

Exemple 1

Requête SQL d'origine :

```
where Node like 'ibm' and Node like 'com'
```

Requête optimisée :

```
where Node like all('ibm','com')
```

Exemple 2

Requête SQL d'origine :

```
where Node like 'ibm' and Node like all('com','uk')
```

Requête optimisée :

```
where Node like all('ibm','com','uk')
```

Exemple 3

Requête SQL d'origine :

```
where Node like all('ibm','com') and Node like all('uk','london')
```

Requête optimisée :

```
where Node like all('ibm','com','uk','london')
```

optimisation OR

Si une clause WHERE se compose de plusieurs prédicats connectés par l'opérateur OR, l'optimisation survient uniquement si la même colonne et le même opérateur de comparaison sont utilisés dans chaque prédicat et que l'opérateur de comparaison est LIKE ou égal à (=).

Le format acceptable de l'optimisation OR est le suivant :

```
where opérateur_colonne_expr1 OR opérateur_colonne_expr2
OR opérateur_colonne_expr3...
```

Dans cette clause WHERE, la variable *opérateur* est LIKE ou = uniquement.

Ce format est optimisé dans une liste ANY comme suit :

```
where opérateur_colonne ANY (expr1, expr2, expr3, ...)
```

Exemple 1

Requête SQL d'origine :

```
where Node like 'London' or Node like 'Copenhagen'
```

Requête optimisée :

```
where Node like any('London', 'Copenhagen')
```

Exemple 2

Requête SQL d'origine :

```
where Severity = 1 or Severity = any(2,3)
```

Requête optimisée :

```
where Severity = any(1,2,3)
```

Exemple 3

Requête SQL d'origine :

```
where Severity = any(1,2) or Severity = any(3,4)
```

Requête optimisée :

```
where Severity = any(1,2,3,4)
```

Réorganisation des prédicats

L'optimiseur réorganise l'évaluation des prédicats dans une clause WHERE selon le coût d'exécution qui leur est affecté.

Si le premier prédicat d'une optimisation OR est défini sur TRUE, les prédicats suivants n'ont pas besoin d'être définis. De même, dans une optimisation AND, si le premier prédicat est défini sur FALSE, les prédicats suivants n'ont pas besoin d'être définis.

Le coût d'exécution affecté dans l'ordre croissant est le suivant :

1. True/False
2. Comparaison d'entier
3. Comparaison de chaîne
4. Entier ANY/ALL/IN
5. Chaîne ANY/ALL/IN
6. Sous-requête - il s'agit d'une instruction SELECT imbriquée

Exemple : optimisation AND (a AND b AND c)

Requête SQL d'origine :

```
where Summary like 'tool' and Serial in (1, 2, 3, 4, 5) and Severity > 2
```

Requête réorganisée optimisée :

```
where Severity > 2 and Summary like 'tool' and Serial in (1, 2, 3, 4, 5)
```

Exemple : optimisation OR (a OR b OR c)

Requête SQL d'origine :

```
where Summary like 'tool' or Serial in (1, 2, 3, 4, 5) or Severity > 2
```

Requête réorganisée optimisée :

```
where Severity > 2 or Summary like 'tool' or Serial in (1, 2, 3, 4, 5)
```

Optimisation manuelle des requêtes

Lorsque vous écrivez des requêtes, suivez ces instructions pour être sûr qu'elles seront traitées aussi efficacement que possible. La différence de performances (par exemple au niveau de la vitesse de la liste AEL (Active Event List) et des listes d'événements du bureau) obtenue via des requêtes efficaces peut être significative. Par exemple, dans un système de production où la table alerts.status contient plus de 30 000 alertes et les zones VARCHAR, telles que la zone de récapitulatif (Summary), contiennent des chaînes longues, des requêtes optimisées s'exécuteront mieux que des requêtes moins efficaces.

Procédure

Utilisez des conditions SQL simples au lieu d'expressions régulières. L'exemple suivant montre comment une expression régulière sur une zone VARCHAR peut être remplacée par une condition SQL simple.
Cet exemple présente une mauvaise condition SQL qui diminue les performances.

```
Summary LIKE 'NETCOOL|netcool'
```

L'expression régulière précédente peut être remplacée par la condition simple suivante :

```
Summary LIKE 'NETCOOL' OR Summary LIKE 'netcool'
```

Instructions d'indexation

L'indexation peut affecter les performances de vos requêtes SQL. Sans indexation, une analyse de base de données complète est généralement effectuée lors de l'exécution d'une requête SQL. Utilisez l'indexation pour limiter le nombre de lignes à examiner.

Tivoli Netcool/OMNIbus prend en charge les structures d'index de hachage et d'arborescence d'index. L'index de hachage prend en charge les comparaisons d'égalité dans les requêtes SQL. L'arborescence d'index est un index organisé qui stocke les valeurs de colonne dans une structure triée et élargit les comparaisons aux égalités dans les requêtes SQL. Par conséquent, une arborescence d'index peut être utilisée dans des requêtes de plage et dans des requêtes avec clause ORDER BY.

Les index sont régénérés lors du redémarrage du serveur ObjectServer. Ils utilisent une faible quantité de mémoire au lieu d'un espace disque physique.

Vous pouvez créer des index sur toutes les tables du serveur ObjectServer, à l'exception des tables des bases de données du système, telles que les bases de données de catalogue et de sécurité. Tivoli Netcool/OMNIbus prend également en charge l'utilisation des index par les clauses de sous-requête dans les requêtes SQL.

Bien qu'il n'y ait aucune limite de nombre d'index que vous pouvez créer sur une table, vous devez utiliser les index avec modération. Les index ont un impact sur les performances car ils sont mis à jour lorsque des opérations d'insertion, de mise

à jour ou de suppression sont effectuées sur la table sur laquelle ils se basent. Pour les tables telles que alerts.status, qui sont mises à jour fréquemment, la création d'un grand nombre d'index peut avoir un effet négatif sur les performances globales du serveur ObjectServer. Trouvez un compromis entre l'indexation pour une extraction rapide des données et la dégradation des performances pendant les opérations d'insertion, de mise à jour et de suppression.

Evitez d'indexer des tables qui contiennent uniquement de petites quantités de données. Evitez également d'indexer les colonnes qui contiennent des valeurs de données qui varient fréquemment.

Les colonnes suivantes se prêtent à l'indexation :

- Les colonnes fréquemment utilisées pour des recherches ou lors du tri, à savoir les colonnes généralement utilisées dans les clauses ORDER BY
- Les colonnes fréquemment utilisées dans les clauses WHERE qui contiennent les formats de prédicat pris en charge pour l'indexation (voir tableau 94)
- Les colonnes contenant des données majoritairement uniques

Les colonnes qui sont définies en tant que clés primaires sont, par défaut, uniquement indexées. Ces index implicites spéciaux ne sont pas stockés dans la table catalog.indexes. La colonne Serial de la table alerts.status est indexée par défaut.

Les restrictions d'indexation sur les colonnes sont les suivantes :

- Seul un index par colonne est autorisé.
- Une colonne définie comme opérateur booléen ne peut pas posséder d'index d'arborescence.

Lors du traitement SQL, le filtre de restriction de la table et la clause WHERE de chaque instruction SELECT, UPDATE, DELETE, FOR EACH ROW et EVALUATE sont examinés pour déterminer si une analyse d'index doit être effectuée à la place d'une analyse de table complète. Une analyse d'index est effectuée lorsqu'un ou plusieurs prédicats remplissent les conditions suivantes :

- Le prédicat utilise l'opérateur d'égalité (=) au format *NomColonne* = *ExpressionConstante*, où *NomColonne* est une colonne indexée.
- Le prédicat utilise l'opérateur inférieur à (<), inférieur ou égal à (<=), supérieur à (>) ou supérieur ou égal à (>=), à condition que *NomColonne* soit une colonne indexée de type arborescence.
- Le prédicat n'est pas connecté à un autre prédicat par un opérateur OR. Par exemple, si la zone Severity ou Serial est indexée, un index n'est pas utilisé dans la requête SQL suivante :

```
select Summary from alerts.status where Severity > 3 or Serial = 102;
```

Le tableau suivant récapitule les formats de prédicat pris en charge pour les index de hachage et les arborescences d'index.

Tableau 94. Formats de prédicat pour les index de hachage et les arborescences d'index

Format de prédicat	Index de hachage	Index d'arborescence
NomColonne = ExpressionConstante	Oui	Oui
NomColonne < ExpressionConstante	Non	Oui
NomColonne > ExpressionConstante	Non	Oui
NomColonne < = ExpressionConstante	Non	Oui

Tableau 94. Formats de prédicat pour les index de hachage et les arborescences d'index (suite)

Format de prédicat	Index de hachage	Index d'arborescence
NomColonne > = *ExpressionConstante*	Non	Oui
NomColonne = *ExpressionConstante* *NomColonne* % < *ExpressionConstante* *NomColonne* % > *ExpressionConstante* *NomColonne* %<= *ExpressionConstante* *NomColonne* %>= *ExpressionConstante*	Non	Non

Référence associée:

«Exemple d'utilisation d'index avec les requêtes SQL»
Ces exemples montrent comment des index peuvent être appliqués à des requêtes SQL.

«Exemple d'utilisation d'index avec des déclencheurs ou des procédures», à la page 339
Cet exemple présente comment des index peuvent être appliqués à des déclencheurs ou à des procédures. Dans cet exemple, il s'agit d'une corrélation entre deux types d'événements, Type 14 et Type 15, de sorte que si les deux se produisent sur le même hôte, ils sont effacés.

Exemple d'utilisation d'index avec les requêtes SQL

Ces exemples montrent comment des index peuvent être appliqués à des requêtes SQL.

Exemple 1

Si la zone Severity ou Serial est indexée, vous pouvez utiliser l'index de la zone Severity (en supposant qu'il s'agit d'une arborescence d'index) dans la requête SQL suivante car toutes les lignes doivent respecter l'expression `Severity > 3`. L'index de la zone Serial n'est pas utilisé car un opérateur OR est utilisé pour relier les deux prédicats.

```
select Summary from alerts.status where
   Severity > 3 and (Serial = 102 or ServerName = 'NCOMS');
```

Exemple 2

Si une arborescence d'index est créée dans la zone Severity, vous pouvez utiliser l'index dans la zone Severity de la requête SQL suivante. Cependant, vous ne pouvez pas utiliser d'index pour la zone LastOccurrence en raison de l'opérateur OR entre le prédicat `LastOccurrence > getdate() - 360` et le prédicat `Summary like 'LinkUp'`. Notez néanmoins que l'expression `getdate() - 360` est considérée comme une constante pour la durée de la requête.

```
select Summary, Severity, Node from alerts.status where
   Severity > 1 and (LastOccurrence > getdate() - 360 or Summary like 'LinkUp')
```

Exemple 3

Prenons la requête suivante :

```
select Summary from alerts.status where Severity > 0;
```

Pour utiliser un opérateur de comparaison tel que >, >=, < ou <= avec un index, une arborescence d'index (qui est un index organisé) est nécessaire. Si seules 100 lignes sur 20 000 possèdent le niveau Severity 0, un tel index réduit le nombre de lignes examinées de seulement 0,5 % et ne fournit pas un gain significatif en termes de performances. Par conséquent, les données de ligne réelles doivent être prises en compte pour décider quelle colonne indexer.

Exemple 4

Si un index de hachage est créé dans la colonne Node, lors de l'exécution de la requête SQL suivante, seules trois recherches de hachage sont exécutées pour tool, bar et toolbar au lieu d'examiner chaque ligne à la recherche d'une égalité entre la colonne Node et l'une des trois valeurs.

```
select Identifier from alerts.status where Node in ('tool', 'bar', 'toobar');
```

Exemple 5

Si une arborescence d'index est créée dans la colonne Severity, lors du traitement de l'instruction SELECT suivante, seules les valeurs Severity 2 et 3 sont recherchées et renvoyées.

```
select * from alerts.status where Severity > 1 and Severity < 4;
```

Si une clause ORDER BY inclut plusieurs colonnes, un index est utilisé pour la première colonne, le cas échéant.

```
select Identifier, Serial from alerts.status order by Severity;
```

Exemple 6

Si la table alerts.status comprend 20 000 lignes, et qu'un index de la zone ServerSerial est appliqué à la requête suivante, seules deux lignes sont examinées au lieu de 20 000 :

```
select Summary from alerts.status where ServerSerial in (102,103);
```

Exemple 7

Tivoli Netcool/OMNIbus prend en charge l'utilisation des index par les clauses de sous-requête dans les requêtes SQL, comme le montrent les exemples suivants :

```
select * from alerts.status where Identifier in (select key from alerts.bar);
```

```
select * from alerts.status where Serial in
(select serial from alerts.broken_events);
```

Exemple d'utilisation d'index avec des déclencheurs ou des procédures

Cet exemple présente comment des index peuvent être appliqués à des déclencheurs ou à des procédures. Dans cet exemple, il s'agit d'une corrélation entre deux types d'événements, Type 14 et Type 15, de sorte que si les deux se produisent sur le même hôte, ils sont effacés.

```
create procedure correlation
begin
 for each x in alerts.status where Type = 14
 begin
  for each y in alerts.status where y.Node = x.Node
   and y.AlertGroup = x.AlertGroup and Type = 15
  begin
   update alerts.status set Severity = 0
```

```
   where Identifier in (y.Identifier, x.Identifier)
  end
 end
end;
```

Dans cet exemple, tout événement de type Type 14 et 15 est effacé si les deux existent pour le même noeud et et la même colonne AlertGroup.

S'il existe 10 000 événements de type Type 14 et 10 000 événements de type Type 15, et que, en moyenne, 10 événements se produisent par noeud unique, les résultats suivants sont possibles :

- Sans indexation, la clause WHERE interne analyse plus de 10 000 * 20 000 lignes, soit 200 millions de lignes. Cette opération est lente et explique pourquoi les instructions FOR EACH ROW imbriquées ne sont pas recommandées sans une bonne indexation.
- Avec un index dans la colonne Type, la clause WHERE interne analyse 10 000 * 10 000 lignes, soit 100 millions de lignes. Cette opération est lente mais concerne la moitié moins de lignes que lors d'une analyse sans indexation.
- Avec un index dans la colonne Node, la clause WHERE interne analyse 10 000 * 10 lignes, soit 100 000 lignes. Cette opération est performante.

Remarque : L'analyse exécutée par l'instruction UPDATE utilise toujours la clé primaire.

Valeurs recommandées lors de la création de déclencheurs

Lorsque vous créez ou modifiez des déclencheurs, vérifiez que ces derniers sont aussi efficaces que possible et qu'ils possèdent le plus faible temps d'exécution possible.

Un déclencheur a l'accès exclusif à la base de données du serveur ObjectServer pour le temps de son exécution. En réduisant le temps d'exécution, vous pouvez libérer du temps pour les autres déclencheurs ou clients qui doivent avoir accès à la base de données. Il est important de réduire le temps d'exécution des déclencheurs de base de données car ils interrompent l'exécution d'une opération de base de données et ralentissent donc l'opération. Par exemple, un déclencheur de pré-insertion de la table alerts.status se déclenche pour chaque nouvel événement. Si une rafale d'événements se produit, le déclencheur est exécuté plusieurs fois.

Le serveur ObjectServer enregistre la durée utilisée par chaque déclencheur durant chaque période de granularité et sauvegarde les détails dans le fichier `$NCHOME/omnibus/log/`*`nom de serveur`*`_trigger_stats.log`*`n`*. Vous pouvez utiliser ce fichier pour identifier les déclencheurs utilisant le plus de temps, déterminer des priorités quant aux déclencheurs à passer en revue et surveiller le système. En règle générale, vous devez réviser un déclencheur s'il utilise plus de 3 secondes toutes les 60 secondes (en d'autres termes, la période de granularité par défaut).

Lorsque vous mettez à jour vos déclencheurs, révisez le fichier journal pour vérifier que vos modifications ne provoquent pas de dégradation des performances.

Utilisez les instructions suivantes pour améliorer les performances de vos déclencheurs.

Evitez les analyses de table dans les déclencheurs de base de données

Les analyses de table sont une opération coûteuse et peuvent se produire lorsque des instructions SQL, notamment FOR EACH ROW, sont appliquées à une table de base de données. Dans un déclencheur de base de données, le coût de ces analyses est élevé si le déclencheur est appelé fréquemment et si la table analysée comprend un grand nombre de lignes. Par exemple, si vous modifiez le déclencheur de dédoublonnage sur la table alerts.status chaque fois que le déclencheur est activé, il analyse alerts.status pour rechercher les lignes correspondant à un jeu de critères. Dans ce cas, l'évolutivité du système est limitée car le déclencheur de base de données prend de plus en plus de temps au fur et à mesure que le nombre de lignes nécessitant une analyse augmente. Evitez également les analyses imbriquées.

Vous pouvez utiliser les techniques suivantes pour éviter l'analyse de table dans les déclencheurs de base de données :

- Effectuez l'analyse d'un déclencheur temporel qui est écrite afin qu'une seule analyse puisse correspondre à plusieurs lignes. Pour voir un exemple de cela, voir le déclencheur generic_clear du fichier `$NCHOME/omnibus/etc/automation.sql`.
- Si vous utilisez une table de consultation pour enrichir les événements, accédez à celle-ci à l'aide de sa clé primaire, comme expliqué ci-après. L'utilisation d'une clé primaire a pour résultat une recherche directe de la ligne et non une analyse (version 7.3 ou ultérieure). . Vous pouvez également limiter la taille de la table de consultation. Le nombre de lignes acceptable pour une table de consultation dépend du site. Il dépend également de facteurs tels que la fréquence d'accès à la table de consultation et les performances matérielles.
- Accédez à une table de consultation à l'aide d'un index.

Evitez la clause EVALUATE

Lorsqu'un déclencheur contient une clause EVALUATE, une table temporaire est créée pour conserver les résultats de l'instruction SELECT dans la clause EVALUATE. La quantité de temps et la quantité de ressources consommées par cette table temporaire dépendent du nombre de colonnes sélectionnées et du nombre de lignes mises en correspondance par la condition de la clause WHERE.

Dans la plupart des cas, vous pouvez remplacer la clause EVALUATE par une clause FOR EACH ROW. Cette clause fait passer le lit les données et ne doit pas passer du temps à créer une table temporaire.

L'application d'une clause GROUP BY à une requête SQL peut être une utilisation adéquate d'une clause EVALUATE.

Evitez l'utilisation excessive de l'instruction WRITE INTO pour la consignation dans le fichier

Lorsque des déclencheurs sont déployés dans des environnements de production, placez en commentaire les instructions WRITE INFO ou supprimez-les. Nous vous conseillons de limiter l'utilisation des instructions WRITE INFO aux environnements de développement, pour les déclencheurs de débogage. La quantité de données consignées durant le débogage peut provoquer un goulot d'étranglement dans les environnements de production.

Déterminez les éléments adaptés à votre système. Par exemple, si la consignation est rarement appelée, vous ne rencontrerez probablement aucun problème. Cependant, si la consignation est appelée plusieurs fois par instruction INSERT (par exemple, dans une boucle imbriquée), vous pouvez rencontrer un goulot d'étranglement.

Si possible, utilisez la clé primaire lors de la modification des lignes

Si la clé primaire d'une table de base de données est utilisée dans la clause WHERE d'une instruction UPDATE, vous accédez à la ligne à l'aide d'une recherche directe et non d'une analyse de table. Par exemple :

```
update alerts.status where Identifier = tt.Identifier set Severity = Severity + 1;
```

Utilisez des index lors de l'utilisation de tables de consultation

Le serveur ObjectServer utilise un index pour accéder aux lignes d'une table si la clé primaire est utilisée dans une instruction FOR EACH ROW.

Cette fonctionnalité est particulièrement utile lorsqu'une table ObjectServer est utilisée comme table de consultation, par exemple pour enrichir les événements. Dans un tel cas, vous devez concevoir la table et les déclencheurs pour accéder à la table de consultation via ses clés primaires afin d'éviter des analyses de table complètes, très onéreuses. Par exemple :

```
create table alerts.iplookup persistent
(
 IpAddr  varchar(32) primary key,
 HostName varchar(8),
 Owner  varchar(40)
);

create or replace trigger set_hostname
group madeup_triggers
priority 10
before insert on alerts.status
for each row
begin
 -- Accédez à la table de consultation à l'aide de la clé primaire
 for each row tt in alerts.iplookup where tt.IpAddr = new.Node
 begin
  set new.Hostname = tt.HostName;
 end;
end;
```

Utilisez le déclencheur generic_clear en tant que base pour les déclencheurs de type corrélation

Le déclencheur generic_clear standard (voir `$NCHOME/omnibus/etc/automation.sql`) établit une corrélation entre les événements de résolution et les problèmes qui y sont associés. Une fois que ce déclencheur s'exécute, toutes les lignes mises en correspondance ont une gravité définie sur 0, afin de préparer leur suppression par l'automatisation delete_clears. Si vous avez besoin de types de déclencheurs de corrélation différents, basez ces déclencheurs sur le déclencheur generic_clear.

Le déclencheur generic_clear standard n'utilise pas la clause EVALUATE pour sélectionner les événements. Au lieu de cela, il utilise la construction FOR EACH ROW pour mettre en boucle les événements afin de remplir une table temporaire avec les événements de problème. Etant donné que cette table temporaire contient un sous-ensemble d'événements de la table alerts.status, le coût de l'opération de

mise à jour appliquée pour établir une relation entre les problèmes et les résolutions diminue. En outre, étant donné que l'identificateur de l'événement de problème est stocké dans la table temporaire, les événements de problème peuvent être directement mis à jour dans la table alerts.status à l'aide de la commande UPDATE VIA pour effectuer une recherche directe dans la ligne. Cette opération tire profit du fait que la zone Identifier est une clé primaire.

Utilisez le dédoublonnage pour effacer les événements, si possible

Le déclencheur de dédoublonnage peut être utilisé pour effacer des événements de problème et l'événement de résolution entrant lorsqu'il existe un mappage un à un entre le problème et la résolution. Vous devez apporter la modification suivante à votre système existant :

- Ecrivez les règles de sonde de sorte à que les événements de problème et de résolution aient le même identificateur.
- Modifiez le déclencheur de dédoublonnage de sorte que, lorsqu'il se déclenche, il vérifie la zone Type. Si le type de l'événement entrant est paramétré sur 2 (résolution), définissez la gravité de l'événement existant sur 0.

Cette approche réduit la quantité de traitements effectués par le déclencheur generic_clear. Elle laisse le déclencheur pour résoudre les cas dans lesquels un événement de résolution unique peut régler de nombreux événements de problème.

Remarques sur la création d'automatisations

Utilisez les pratiques recommandées suivantes pour créer de nouvelles automatisations :

- Vérifiez qu'il existe actuellement une automatisation avant de tenter d'en créer de nouvelles avec la même fonction.
- Dans toutes les conditions WHERE de l'automatisation, utilisez les instructions de réorganisation des prédicats dans les requêtes SQL. Par exemple, comparez les entiers, les caractères puis les expressions régulières. Pour plus d'informations, voir «Règles d'optimisation pour les requêtes SQL», à la page 334.
- Assurez-vous que le déclencheur de l'automatisation n'acquiert pas d'événements qui ont déjà été traités, en particulier dans le cas des scripts externes.
- Pour les déclencheurs temporels, définissez l'intervalle d'application des différents déclencheurs afin d'empêcher leur activation simultanée.
- Ajoutez une description pour chaque automatisation nouvellement créée.
- Les automatisations peuvent mettre à jour l'entrée de journal si elles changent des événements dans la base de données du serveur ObjectServer.

Testez vos modifications

Une fois que les nouveaux déclencheurs sont développés et validés, testez leurs performances de la manière suivante :

1. Vérifiez que les données sur lesquelles vous exécutez les tests sont représentatives du système de production.
2. Vérifiez que le nombre de lignes des tables auxquelles accède le déclencheur est représentatif du système de production.

3. Mesurez l'impact sur les performances du système à l'aide du profilage et en collectant les statistiques des déclencheurs.

Annexe A. Aide-mémoire des commandes d'application

Le présent aide-mémoire répertorie les commandes permettant de démarrer les applications et les utilitaires fournis avec Tivoli Netcool/OMNIbus.

Les applications et utilitaires disponibles sont regroupés selon leurs fonctions.

Applications principales

Les commandes suivantes permettent de démarrer les principales applications Tivoli Netcool/OMNIbus.

Tableau 95. Applications principales

Commande	Description
nco_objserv	Démarre un serveur ObjectServer.
nco_g_objserv_uni	Démarre une passerelle ObjectServer unidirectionnelle.
nco_g_objserv_bi	Démarre une passerelle ObjectServer bidirectionnelle.
nco_config (Netcool/OMNIbus Administrator)	Démarre l'interface graphique (Netcool/OMNIbus Administrator) utilisée pour l'administration des serveurs ObjectServer et des agents de processus. Sur les systèmes d'exploitation Windows, vous pouvez également démarrer cette interface à partir du menu suivant : **Démarrer** > **Tous les programmes** > **NETCOOL Suite** > **Administrator**
nco_pad (Démon de l'agent de processus)	Démarre le démon de l'agent de processus.
nco_aen (notification d'événement accéléré)	Démarre l'interface graphique AEN (Accelerated Event Notification). Sur les systèmes d'exploitation Windows, vous pouvez également démarrer l'interface AEN à partir du menu suivant : **Démarrer** > **Tous les programmes** > **NETCOOL Suite** > **Notifier**
nco_event (liste d'événements)	Démarre l'interface graphique de liste d'événements sous UNIX ou Linux.

© Copyright IBM Corp. 1994, 2014

Tableau 95. Applications principales (suite)

Commande	Description
NCOEvent (liste d'événements)	Démarre l'interface graphique de liste d'événements sous Windows. Sur les systèmes d'exploitation Windows, vous pouvez également démarrer la liste d'événements à partir du menu suivant : **Démarrer** > **Tous les programmes** > **NETCOOL Suite** > **Event List**
nco_elct (liste d'événements transitoires)	Démarre l'interface graphique de liste des événements transitoires sous UNIX ou Linux.
NCOElct (liste d'événements transitoires)	Démarre l'interface graphique de liste des événements transitoires sous Windows.
nco_icw (assistant de configuration initiale)	Automatise les tâches de configuration initiale, telles que la création de serveurs ObjectServer et de passerelles.

Utilitaires d'administration

Tableau 96. Utilitaires d'administration

Commande	Description
nco (Netcool/OMNIbus Conductor)	Démarre l'interface graphique Netcool/OMNIbus Conductor sous UNIX et Linux.
NCOConductor (Netcool/OMNIbus Conductor)	Démarre l'interface graphique Netcool/OMNIbus Conductor sous Windows.
nco_bridgeserv	Démarre un serveur de pont pare-feu.
nco_proxyserv	Démarre un serveur proxy.
nco_igen	Génère un fichier d'interfaces de communication serveur pour un système d'exploitation UNIX ou Linux spécifique. Les fichiers d'interfaces peuvent également être générés à l'aide de l'éditeur de serveur.
nco_xigen (éditeur de serveurs)	Démarre l'interface graphique d'éditeur de serveur sous UNIX et Linux. Il n'existe aucun utilitaire de ligne de commande pour démarrer l'éditeur de serveur sur les systèmes d'exploitation Windows. Utilisez plutôt le menu suivant : **Démarrer** > **Tous les programmes** > **Suite NETCOOL** > **Utilitaires système** > **Editeur de serveurs**.
nco_id (outil d'affichage de version de serveur d'objets)	Affiche des informations concernant votre installation de Tivoli Netcool/OMNIbus et inclut des options pour la sortie standard ou détaillée et pour l'écriture d'informations dans un fichier HTML.

Tableau 96. Utilitaires d'administration (suite)

Commande	Description
nco_version	Renvoie des informations de version sur votre installation de Tivoli Netcool/OMNIbus, y compris les versions de bibliothèque et les paramètres de débogage. **nco_id** utilise **nco_version** pour collecter des informations sur les bibliothèques Netcool.
nco_install_ospam	Installe le module PAM (Pluggable Authentication Module) ObjectServer sur les systèmes d'exploitation UNIX et Linux.

Utilitaires ObjectServer

Tableau 97. Utilitaires ObjectServer

Commande	Description
nco_dbinit (Utilitaire d'initialisation de base de données)	Crée un ou plusieurs serveurs ObjectServer. Cet utilitaire a remplacé **nco_new_server** dans la version 7.0 de Tivoli Netcool/OMNIbus.
nco_osreport (ObjectServer Report Generator)	Copie le contenu des tables ObjectServer dans un fichier HTML ou XML. Cet utilitaire peut également être utilisé pour exporter une configuration ObjectServer vers un ensemble de fichiers SQL, qui peut ensuite être utilisé pour créer le contenu initial d'un nouveau serveur ObjectServer.
nco_confpack (outil de création de packages de configuration)	Crée et gère les packages de configuration permettant de transférer les objets de configuration entre les serveurs ObjectServer, de déployer des serveurs ObjectServer en double et de sauvegarder les configurations ObjectServer.
nco_sql (interface interactive SQL)	Démarre l'interface interactive SQL sous UNIX et Linux. Vous pouvez utiliser l'interface interactive SQL pour vous connecter à un serveur ObjectServer et émettre des commandes SQL.
isql (interface interactive SQL)	Démarre l'interface interactive SQL sous Windows. Vous pouvez utiliser l'interface interactive SQL pour vous connecter à un serveur ObjectServer et émettre des commandes SQL.

Tableau 97. Utilitaires ObjectServer (suite)

Commande	Description
nco_postmsg	Génère des instructions SQL INSERT ObjectServer en utilisant des entrées de données provenant de la ligne de commande, de scripts ou de fichiers texte. Vous pouvez utiliser cet utilitaire en remplacement des utilitaires IBM Tivoli Enterprise Console **postemsg** et **postzmsg**.
nco_check_store (Utilitaire de vérification de point de contrôle)	Vérifie la validité des fichiers de point de contrôle existants. Cet utilitaire est conçu pour être utilisé par les automatisations et ne peut être utilisé que pour vérifier les magasins ObjectServer qui ne sont pas en cours d'utilisation.
nco_store_resize	Modifie la taille limite absolue du magasin de mémoires table_store.
nco_baroc2sql (outil de conversion BAROC)	Convertit les fichiers Tivoli Enterprise Console BAROC (BAsic Recorder of Objects in C) en fichiers SQL ObjectServer, que vous pouvez ensuite importer sur un serveur ObjectServer.

Utilitaires de contrôle de processus

Tableau 98. Utilitaires de contrôle de processus

Commande	Description
nco_pa_addentry	Ajoute un service ou un processus à l'agent de processus en cours d'exécution.
nco_pa_start	Démarre les services et les processus qui sont sous le contrôle de l'agent de processus.
nco_pa_stop	Arrête les services et les processus qui sont sous le contrôle de l'agent de processus.
nco_pa_status	Affiche le statut des services et des processus qui sont sous le contrôle de l'agent de processus. Pour chaque service, l'utilitaire renvoie une liste de processus définis, le statut de chaque processus et l'identificateur de processus.
nco_pa_shutdown	Arrête un agent de processus. Les services et les processus sous le contrôle de l'agent de processus peuvent continuer à s'exécuter si nécessaire.

Utilitaires de chiffrement

Tableau 99. Utilitaires de chiffrement

Commande	Description
nco_crypt	Permet de chiffrer les mots de passe de connexion en texte en clair à l'aide de la norme DES (Data Encryption Standard), pour utilisation par les sondes et les passerelles Tivoli Netcool/OMNIbus. Lorsque vous exécutez Tivoli Netcool/OMNIbus en mode FIPS 140-2, utilisez **nco_aes_crypt** pour chiffrer les mots de passe dans des fichiers de propriétés de sonde et de passerelle.
nco_aes_crypt	Chiffre et déchiffre les valeurs et les données de chaîne dans un fichier conformément à la norme AES (Advanced Encryption Standard). Cette commande requiert un fichier de clés généré à l'aide de l'utilitaire **nco_keygen**.
nco_g_crypt	Permet de chiffrer les mots de passe de connexion en texte en clair à l'aide de la norme DES (Data Encryption Standard), pour utilisation par les sondes et les passerelles Tivoli Netcool/OMNIbus. Lorsque vous exécutez Tivoli Netcool/OMNIbus en mode FIPS 140-2, utilisez **nco_aes_crypt** pour chiffrer les mots de passe dans des fichiers de propriétés de sonde et de passerelle.
nco_pa_crypt	Chiffre les mots de passe de connexion en texte en clair sur les systèmes d'exploitation UNIX et Linux (utilisé dans le fichier de configuration d'agent de processus). Lorsque vous exécutez Tivoli Netcool/OMNIbus en mode FIPS 140-2, utilisez **nco_aes_crypt** pour chiffrer les mots de passe dans le fichier de configuration d'agent de processus.
nco_sql_crypt	Chiffre les mots de passe de connexion en texte en clair sur les systèmes d'exploitation UNIX et Linux (utilisé par l'interface interactive SQL (**nco_sql**)). Lorsque vous exécutez Tivoli Netcool/OMNIbus en mode FIPS 140-2, utilisez **nco_aes_crypt** pour chiffrer les mots de passe dans le fichier de propriétés de l'interface interactive SQL.
nco_get_login_token	**nco_sql**, en cas d'exécution en mode sécurisé, utilise **nco_get_login_token** pour générer un jeton de connexion à utilisation unique et à durée de vie fixe.
nco_keygen	Génère des clés de chiffrement AES et les stocke dans des fichiers de clés.

Tableau 99. Utilitaires de chiffrement (suite)

Commande	Description
nc_gskcmd	Gère les certificats SSL et les bases de données de clés sur les déploiements de Tivoli Netcool/OMNIbus qui s'exécutent en mode FIPS 140-2 ou dans des réseaux comportant des clients Java qui nécessitent des communications chiffrées.
nc_ikeyman	Démarre l'interface graphique iKeyman de base de données de clés et de gestion des certificats SSL. iKeyman ne doit pas être utilisé sur les déploiements de Tivoli Netcool/OMNIbus qui s'exécutent en mode FIPS 140-2 ou dans des réseaux comportant des clients Java qui nécessitent des communications chiffrées.

Utilitaires de sonde

Tableau 100. Utilitaires de sonde et de passerelle

Commande	Description
nco_http	Effectue la connexion aux sondes qui possèdent une interface HTTP ou HTTPS et émet des requêtes HTTP.
nco_probereloadrules	Recharge à distance un fichier de règles de sonde sans redémarrer la sonde.
nco_setprobeprop	Met à jour à distance la valeur d'une propriété de sonde.
nco_probeeventfactory	Génère à distance un événement sur une sonde.
nco_mibmanager	Démarre l'interface graphique MIB Manager de Tivoli Netcool/OMNIbus. MIB Manager analyse les fichiers de la base d'informations (MIB) de gestion SNMP et génère des fichiers de règles Netcool pour une utilisation avec la sonde SNMP.
nco_rules_util (Probe Rules Utility)	Vous pouvez utiliser l'outil de ligne de commande Probe Rules Utility afin de faciliter la gestion des fichiers de règles de sonde qui sont stockés dans un référentiel centralisé ou un système de fichiers.

Utilitaires d'aide en ligne

Tableau 101. Utilitaires d'aide en ligne

Commande	Description
IC_start	Démarre le serveur local IEHS (IBM Eclipse Help System).
IC_end	Arrête le serveur local IEHS.
help_end	Arrête le serveur local IEHS lorsqu'il s'exécute en mode autonome.

Autres utilitaires

Tableau 102. Autres utilitaires

Commande	Description
nco_mail	Script d'encapsuleur qui utilise la fonction sendmail pour envoyer des courriers électroniques. Utilisé par le déclencheur mail_on_critical.
nco_functions	Fichier qui peut être appelé à partir de scripts de shell pour fournir des fonctions utiles à l'environnement Tivoli Netcool/OMNIbus.
nco_run	Script d'exécution qui permet de s'assurer que l'environnement est configuré et que le code binaire exécuté est correct.
nco_message	Affiche à l'écran une boîte de dialogue avec un message. Voici un exemple d'utilisation : `./nco_message -title "Statut ObjectServer" -message "Le serveur ObjectServer répond aux pings" -type info`
nco_ping	Envoie une commande ping à un serveur Netcool/OMNIbus afin de vérifier sa disponibilité. Par exemple, cette commande peut être utilisée pour vérifier qu'un serveur ObjectServer ou une passerelle est en cours d'exécution.
nco_xcheck	Vérifiez que la variable d'environnement `$DISPLAY` est définie sur les systèmes d'exploitation UNIX et Linux.

Annexe B. Informations de référence sur les commandes SQL ObjectServer

Informations de référence sur les commandes SQL ObjectServer.

Les commandes sont répertoriées alphabétiques afin de faciliter la référence.

Concepts associés:

Chapitre 5, «SQL ObjectServer», à la page 167
Le serveur ObjectServer possède une interface SQL permettant de définir et de manipuler les objets de base de données relationnelle tels que les tables et les vues.

ADD COLUMN

Pour ajouter des colonnes à une table existante, utilisez le paramètre `ADD COLUMN` avec la commande `ALTER TABLE`.

Dans cette commande, la syntaxe d'ajout des colonnes est la suivante :

```
ADD [COLUMN] nom_colonne type_données [ NODEFAULT | NOMODIFY | HIDDEN ]
```

Lorsque vous ajoutez des colonnes, vous devez spécifier le nom de la colonne et le type de données. Vous pouvez également spécifier des propriétés facultatives.

Vous ne pouvez pas ajouter de clés primaires à une table existante.

Lorsqu'une nouvelle colonne est ajoutée à la table à l'aide de la clause `NODEFAULT`, toute instruction INSERT envoyée depuis les sondes ou passerelles échouent. L'échec se produit car elles ne sont pas conformes à la contrainte `NODEFAULT`.

Lorsque vous marquez une colonne comme `NODEFAULT` et une ligne comme insérée dans cette table, l'instruction d'insertion doit définir explicitement une valeur pour cette colonne, sinon l'instruction INSERT est invalidée.

Ceci se produit car l'instruction INSERT n'essaiera pas de remplir la colonne `NODEFAULT`. Le déclencheur sera également invalidé et ne pourra pas se recompiler.

Référence associée:

«ALTER TABLE», à la page 357
La commande `ALTER TABLE` permet de modifier les caractéristiques d'une table existante et de ses colonnes. Vous pouvez ajouter, supprimer et modifier des colonnes ; vous pouvez également définir manuellement la valeur des zones de type `INCR`.

«Spécification des types de données des colonnes», à la page 171
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.

«Spécification des propriétés facultatives des colonnes», à la page 174
Vous pouvez spécifier les propriétés facultatives des colonnes que vous définissez lors de la création d'une table.

© Copyright IBM Corp. 1994, 2014

ALTER COLUMN

Pour modifier les colonnes d'une table existante, utilisez le paramètre ALTER COLUMN avec la commande ALTER TABLE.

Dans cette commande, la syntaxe de modification des colonnes se présente comme suit :

```
ALTER [COLUMN] nom_colonne SET NOMODIFY { TRUE | FALSE }
ALTER [COLUMN] nom_colonne SET HIDDEN { TRUE | FALSE }
ALTER [COLUMN] nom_colonne SET NODEFAULT { TRUE | FALSE }
ALTER [COLUMN] nom_colonne SET WIDTH valeur
```

Utilisez les instructions suivantes pour modifier les propriétés d'une colonne :

- Pour modifier les propriétés NOMODIFY, HIDDEN et NODEFAULT d'une colonne existante, définissez la propriété appropriée sur TRUE ou FALSE. Une colonne de clé primaire doit posséder une valeur par défaut et ne peut pas être masquée.
- Pour modifier la largeur d'une colonne avec un type de données varchar, utilisez la propriété WIDTH et spécifiez le paramètre *valeur* sous forme de longueur en octets. Vous ne pouvez pas modifier la largeur des clés primaire.

Lorsque vous marquez une colonne comme NODEFAULT et une ligne comme insérée dans cette table, l'instruction d'insertion doit définir explicitement une valeur pour cette colonne, sinon l'instruction INSERT est invalidée.

Ceci se produit car l'instruction INSERT n'essaiera pas de remplir la colonne NODEFAULT. Le déclencheur sera également invalidé et ne pourra pas se recompiler.

Référence associée:

«Spécification des propriétés facultatives des colonnes», à la page 174
Vous pouvez spécifier les propriétés facultatives des colonnes que vous définissez lors de la création d'une table.

«ALTER TABLE», à la page 357
La commande ALTER TABLE permet de modifier les caractéristiques d'une table existante et de ses colonnes. Vous pouvez ajouter, supprimer et modifier des colonnes ; vous pouvez également définir manuellement la valeur des zones de type INCR.

ALTER FILE

La commande ALTER FILE permet de modifier la configuration d'un fichier ObjectServer existant.

Syntaxe

```
ALTER FILE nom_fichier
 TRUNCATE |
 SET ENABLED { TRUE | FALSE };
```

Le paramètre TRUNCATE efface les informations qui ont été inscrites dans le fichier physique. En cas de fichiers physiques multiples, le fichier actuellement en cours d'écriture est tronqué. Les autres fichiers de l'ensemble sont supprimés.

Le paramètre ENABLED active et désactive la consignation. Si l'option est définie sur TRUE, une commande WRITE INTO écrit des données dans le fichier. Si l'option est définie sur FALSE, les commandes WRITE INTO sont ignorées et aucune donnée n'est écrite dans le fichier. La désactivation d'un fichier est utile lorsque vous souhaitez arrêter temporairement la consignation mais que vous ne souhaitez pas

supprimer le fichier que vous avez configuré.

Exemple

```
alter file logit truncate;
```

Référence associée:

«WRITE INTO», à la page 410
La commande WRITE INTO permet de consigner les informations de connexion dans les fichiers ObjectServer. Par exemple, vous pouvez utiliser cette commande pour écrire la sortie d'un déclencheur dans un fichier journal. Un fichier ObjectServer est un fichier logique, qui détient un fichier ou un ensemble de fichiers correspondant sur le système de fichiers physique.

ALTER GROUP

La commande ALTER GROUP permet de modifier les paramètres utilisateur du groupe spécifié. Vous pouvez modifier plusieurs paramètres en une seule commande ALTER GROUP.

Syntaxe

```
ALTER GROUP 'nom_groupe'
 SET COMMENT 'chaîne_commentaire'
 ASSIGN [ RESTRICTION ] FILTER nom_filtre_restriction
 REMOVE [ RESTRICTION ] FILTER nom_filtre_restriction
 ASSIGN MEMBERS 'nom_utilisateur', ...
 REMOVE MEMBERS 'nom_utilisateur', ... ;
```

La variable *nom_groupe* est une chaîne de texte contenant le nom unique du groupe en cours de modification. Vous ne pouvez pas modifier ce nom.

Utilisez le paramètre SET COMMENT pour modifier la description du groupe.

Utilisez le paramètre ASSIGN ou REMOVE RESTRICTION FILTER pour attribuer ou supprimer les filtres de restriction qui s'appliquent au groupe. Un seul filtre de restriction par table peut être attribué à un groupe.

Utilisez le paramètre ASSIGN ou REMOVE MEMBERS pour attribuer des utilisateurs comme membre de groupe ou pour les supprimer du groupe.

Exemple

```
alter group 'AutoAdmin' assign members 'sue';
```

ALTER ROLE

La commande ALTER ROLE permet de modifier la description d'un rôle existant.

Syntaxe

```
ALTER ROLE 'nom_rôle' SET COMMENT 'chaîne_commentaire';
```

La variable *nom_rôle* est une chaîne de texte contenant le nom unique du rôle en cours de modification. Ce nom ne peut pas être modifié.

Utilisez le paramètre COMMENT pour modifier la description du rôle.

Exemple

```
alter role 'SuperAdmin' set comment 'enhanced description of role';
```

ALTER SYSTEM

La commande ALTER SYSTEM permet de modifier les paramètres par défaut et actuels du serveur ObjectServer en paramétrant les propriétés, d'arrêter le serveur ObjectServer, de supprimer des connexions utilisateur ou de sauvegarder le serveur ObjectServer.

- «Syntaxe»
- «Arrêt du serveur ObjectServer»
- «Spécification des propriétés du serveur ObjectServer»
- «Suppression de connexions utilisateur»
- «Sauvegarde du serveur ObjectServer», à la page 357
- «Restauration du serveur ObjectServer», à la page 357

Syntaxe

```
ALTER SYSTEM
 {
 SHUTDOWN |
 SET 'nom_propriété' = valeur [ ... ] |
 DROP CONNECTION id_connexion [, ... ] |
 BACKUP 'nom_répertoire'
 }
;
```

Arrêt du serveur ObjectServer

Vous pouvez arrêter le serveur ObjectServer avec la commande ALTER SYSTEM SHUTDOWN.

Spécification des propriétés du serveur ObjectServer

Vous pouvez définir les propriétés du serveur ObjectServer avec le mot clé SET, suivi du nom de la propriété incluse entre guillemets simples et de la valeur de la propriété. Vous pouvez modifier plusieurs propriétés en une seule commande. Outre la mise à jour de la table catalog.properties, les propriétés modifiées sont inscrites dans le fichier de propriétés.

L'exemple suivant présente le mot clé SET avec la commande ALTER SYSTEM.

```
alter system set 'Auto.StatsInterval' = 15 set 'AlertSecurityModel' = 1;

alter system shutdown;
```

Suppression de connexions utilisateur

Vous pouvez supprimer les connexions utilisateur à l'aide de la commande ALTER SYSTEM DROP CONNECTION. Spécifiez un ou plusieurs identificateurs de connexion dans une liste séparée par des virgules. Vous pouvez rechercher les identificateurs de toutes les connexions en cours en interrogeant la table système catalog.connections. La colonne ConnectionID contient l'identificateur de connexion.

Sauvegarde du serveur ObjectServer

Vous pouvez sauvegarder le serveur ObjectServer à l'aide de la commande ALTER SYSTEM BACKUP. Spécifiez le chemin vers un répertoire existant où vous souhaitez sauvegarder les fichiers. Cette valeur doit être entre guillemets. La sauvegarde génère des copies des fichiers ObjectServer `.tab` dans le répertoire spécifié. Le répertoire ne peut pas être celui dans lequel les fichiers de données du serveur ObjectServer sont stockés (par défaut, `$NCHOME/omnibus/db/`*`nom_serveur`*.

Conseil : Les déclencheurs du groupe de déclenchement automatic_backup_system, définis dans le fichier `$NCHOME/omnibus/etc/automation.sql`, utilisent la commande ALTER SYSTEM BACKUP pour fournir un utilitaire de sauvegarde automatique. Le déclencheur automatic_backup est désactivé par défaut. Vous devez l'activer pour créer automatiquement des sauvegardes. Vous pouvez également personnaliser ce déclencheur afin de l'adapter à votre environnement. Par exemple, vous pouvez modifier le nombre de sauvegardes enregistrées.

Restauration du serveur ObjectServer

Pour rétablir le serveur ObjectServer au moment de la transmission de la commande BACKUP, copiez les fichiers `.tab` du serveur ObjectServer dans le répertoire de fichiers de données du serveur ObjectServer. Vous ne pouvez utiliser les fichiers de sauvegarde que sur un ordinateur doté du même système d'exploitation que l'ordinateur sur lequel ils ont été créés.

Concepts associés:

«Création d'un fichier de point de contrôle», à la page 32
Les fichiers de point de création sont générés pour chaque magasin de mémoire persistant. Seuls les magasins de mémoire persistants font l'objet d'un point de contrôle.

Référence associée:

«Propriétés et options de ligne de commande du serveur ObjectServer», à la page 4
Utilisez les propriétés ou les options de ligne de commande ObjectServer pour configurer les paramètres du serveur ObjectServer. Pour éviter les erreurs, ajoutez autant de propriétés que possible au fichier de propriétés plutôt que des options de ligne de commande. D'autres utilitaires sont fournis que vous pouvez utiliser pour chiffrer les valeurs de propriété.

«ALTER TRIGGER», à la page 358
La commande `ALTER TRIGGER` permet de modifier les paramètres d'un déclencheur existant. Vous pouvez modifier plusieurs paramètres en une seule commande `ALTER TRIGGER`.

ALTER TABLE

La commande `ALTER TABLE` permet de modifier les caractéristiques d'une table existante et de ses colonnes. Vous pouvez ajouter, supprimer et modifier des colonnes ; vous pouvez également définir manuellement la valeur des zones de type `INCR`.

Restriction : Vous ne pouvez pas modifier les tables système.

Syntaxe

```
ALTER TABLE [nom_base_de_données.]nom_table
 SET INCR valeur
 ADD [COLUMN] nom_colonne type_données [ NODEFAULT | NOMODIFY | HIDDEN ]
```

```
DROP [COLUMN] nom_colonne
ALTER [COLUMN] nom_colonne SET NOMODIFY { TRUE | FALSE }
ALTER [COLUMN] nom_colonne SET HIDDEN { TRUE | FALSE }
ALTER [COLUMN] nom_colonne SET NODEFAULT { TRUE | FALSE }
ALTER [COLUMN] nom_colonne SET WIDTH valeur;
```

Vous pouvez spécifier plusieurs paramètres ADD, DROP ou ALTER dans une seule commande ALTER TABLE.

Exemple

```
alter table mytab add col3 real;
```

Définition manuelle des valeurs d'incrément

Les colonnes de type INCR permettent d'affecter à chaque ligne d'une table un numéro unique, comme c'est le cas pour la colonne Serial de la table alerts.status. Lors de chaque insertion, la valeur d'incrément en cours est copiée dans la nouvelle ligne puis est augmentée pour la ligne suivante. Vous pouvez utiliser la clause SET INCR (set increment) pour définir la valeur d'incrément des lignes suivantes. Cela est très utile, par exemple, lorsque vous souhaitez modifier la valeur Serial affectée aux nouvelles alertes de la table alerts.status.

La syntaxe est :

```
ALTER TABLE nom_table SET INCR valeur
```

Où *nom_table* est le nom de la table qui contient la colonne de type INCR et *valeur* est un entier.

Par exemple, pour exécuter la procédure avec l'interface interactive SQL, utilisez les commandes suivantes :

```
1> alter table alerts.status set incr 20100
2> go
```

Remarque : Afin d'éviter les valeurs en double, vous ne pouvez pas définir une valeur d'incrément identique à la valeur d'une ligne existante. En outre, vous ne pouvez pas définir une valeur INCR qui ait pour conséquence la création de futures insertions ayant la même valeur que les lignes qui existent déjà.

Concepts associés:

«Tables système», à la page 171
Les tables système sont des tables spéciales gérées par le serveur ObjectServer et contiennent des métadonnées sur les objets du serveur ObjectServer.

ALTER TRIGGER

La commande ALTER TRIGGER permet de modifier les paramètres d'un déclencheur existant. Vous pouvez modifier plusieurs paramètres en une seule commande ALTER TRIGGER.

Syntaxe

```
ALTER TRIGGER nom_déclencheur
 SET PRIORITY entier
 SET ENABLED { TRUE | FALSE }
 SET GROUP nom_groupe_déclencheurs
 SET DEBUG { TRUE | FALSE };
```

Attribuez une valeur comprise entre 1 et 20 à SET PRIORITY pour modifier la priorité d'un déclencheur. Plus le numéro est bas, plus la priorité est élevée.

Utilisez SET ENABLED TRUE pour activer un déclencheur ou SET ENABLED FALSE pour le désactiver. Si un déclencheur est activé, il est mis en application lorsque l'incident associé se produit. Si un déclencheur n'est pas activé, il n'est pas mis en application lorsque l'incident associé se produit.

Utilisez SET GROUP pour modifier le groupe de déclencheurs selon le nom de groupe spécifié.

Utilisez SET DEBUG pour activer ou désactiver le débogage du déclencheur. S'il est désactivé, les informations de débogage sont envoyées au journal des messages du serveur ObjectServer, si le niveau du message est `debug`.

Exemple

```
alter trigger mytrig set priority 1;
```

ALTER TRIGGER GROUP

La commande `ALTER TRIGGER GROUP` permet d'activer ou de désactiver un groupe de déclencheurs existant.

Syntaxe

```
ALTER TRIGGER GROUP nom_groupe_déclencheurs | expression
 SET ENABLED { TRUE | FALSE };
```

Un groupe de déclencheurs est désactivé par défaut.

Vous pouvez spécifier le nom d'un groupe de déclencheurs ou une expression à l'aide de cette commande. S'il s'agit d'une expression, le nom n'est pas évalué avant la phase d'exécution.

Exemples

Cet exemple désactive le groupe de déclencheurs update_database_triggers.

```
alter trigger group update_database_triggers set enabled false;
```

Cet exemple désactive tous les déclencheurs, à l'exception des déclencheurs de passerelle (qui appartiennent au groupe de déclencheurs gateway_triggers), en répertoriant individuellement les noms de tous les autres groupes de déclencheurs à désactiver.

```
Create trigger disable_triggers
 Group gateway_triggers
 Priority 1
 on signal gw_counterpart_up
 begin
 alter trigger group nom_groupe_déclencheurs_1 set enabled false;
...
 alter trigger group nom_groupe_déclencheurs_n set enabled false;
end;
```

Cet exemple utilise une expression permettant de désactiver tous les déclencheurs, à l'exception des déclencheurs de passerelle appartenant au groupe de déclencheurs gateway_triggers. Lors de la d'exécution, la boucle FOR EACH ROW permet de réaliser les actions dans l'expression, sur chaque ligne de la table catalog.triggers.

```
Create trigger disable_triggers
 Group gateway_triggers
 Priority 1
 on signal gw_counterpart_up
 begin
 for each row tg in catalog.triggers where
  tg.GroupName <> 'gateway_triggers'
 begin
  alter trigger group tg.GroupName set enabled false;
 end;
end;
```

ALTER USER

La commande ALTER USER permet de modifier les paramètres, tels que le mot de passe, d'un utilisateur spécifique. Vous pouvez modifier plusieurs paramètres en une seule commande ALTER USER.

Syntaxe

```
ALTER USER 'nom_utilisateur'
 SET PASSWORD 'mot_de_passe' [ AUTHORIZE PASSWORD 'ancien_mot_de_passe' ]
 [ ENCRYPTED ]
 SET FULL NAME 'nom_utilisateur_complet'
 SET ENABLED { TRUE | FALSE }
 SET PAM { TRUE | FALSE }
 ASSIGN [ RESTRICTION ] FILTER nom_filtre_restriction
 REMOVE [ RESTRICTION ] FILTER nom_filtre_restriction ;
```

La variable *nom_utilisateur* est une chaîne de texte contenant le nom unique de l'utilisateur en cours de modification. Ce nom ne peut pas être modifié.

Utilisez le paramètre PASSWORD pour modifier le mot de passe de l'utilisateur. Notez que vous ne pouvez pas modifier le mot de passe d'un utilisateur qui est authentifié en externe sur un système LDAP. Vous pouvez modifier le mot de passe d'un utilisateur authentifié en externe sur un système PAM uniquement si le système PAM externe est configuré à cet effet. Si la modification du mot de passe d'un utilisateur authentifié via le module PAM est autorisée, vous devez également utiliser les mots clés AUTHORIZE PASSWORD pour indiquer l'ancien mot de passe.

Utilisez le paramètre ENABLED pour activer (TRUE) ou désactiver (FALSE) l'utilisateur spécifié. Un utilisateur activé détient un accès en connexion au système.

Attribuez la valeur TRUE à PAM pour permettre l'authentification externe de l'utilisateur. La valeur PAM ou LDAP doit également être attribuée à la propriété du serveur ObjectServer **Sec.ExternalAuthentication**, selon votre système d'authentification. Si la valeur FALSE est attribuée à PAM ou none à **Sec.ExternalAuthentication**, l'utilisateur ne peut pas être authentifié en externe. Si vous souhaitez procéder à une authentification ObjectServer, attribuez la valeur FALSE à PAM. Pour plus d'informations relatives à PAM ou LDAP, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Utilisez le paramètre ASSIGN ou REMOVE RESTRICTION FILTER pour attribuer ou supprimer les filtres de restriction qui s'appliquent à l'utilisateur. Un seul filtre de restriction par table peut être attribué à un utilisateur.

Exemple

```
alter user 'joe' set password 'topsecret';
```

Référence associée:

«Propriétés et options de ligne de commande du serveur ObjectServer», à la page 4
Utilisez les propriétés ou les options de ligne de commande ObjectServer pour configurer les paramètres du serveur ObjectServer. Pour éviter les erreurs, ajoutez autant de propriétés que possible au fichier de propriétés plutôt que des options de ligne de commande. D'autres utilitaires sont fournis que vous pouvez utiliser pour chiffrer les valeurs de propriété.

CASE WHEN

Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser l'instruction CASE WHEN pour effectuer une ou plusieurs actions basées sur une condition. Si la condition n'est pas remplie, vous pouvez, si vous le souhaitez, effectuer une action différente.

Syntaxe

```
CASE
  WHEN condition THEN liste_commandes_action
  ...
 [ ELSE liste_commandes_action ]
END CASE;
```

Si la première condition est remplie (définie sur TRUE), les instructions suivant le mot clé THEN sont exécutées de façon séquentielle jusqu'à une instruction WHEN, ELSE ou END CASE. Sinon, s'il existe une instruction WHEN pour laquelle la condition est remplie, les instructions suivant le mot clé THEN sont exécutées jusqu'à une instruction WHEN, ELSE ou END CASE. Si aucune condition précédente n'est remplie et qu'une instruction ELSE est présente, les instructions suivant l'instruction ELSE sont exécutées jusqu'à une nouvelle instruction.

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

Référence associée:

«Construction d'une instruction de corps de procédure SQL», à la page 193
Le corps d'une procédure SQL contient un ensemble d'instructions SQL et de constructions de programmation qui gèrent les données du serveur ObjectServer.

CHECK STATEMENT

La commande CHECK STATEMENT permet d'analyser et de vérifier la syntaxe des commandes SQL placées entre guillemets, et de renvoyer un message de succès ou une description des erreurs.

Syntaxe

```
CHECK STATEMENT 'commande; commande; ...';
```

Etant donné que la commande CHECK STATEMENT n'exécute pas les commandes SQL, les erreurs d'exécution ne sont pas détectées. De plus, certaines erreurs superficielles peuvent s'afficher si une série de commandes repose sur les commandes précédentes en cours d'exécution.

CREATE DATABASE

La commande CREATE DATABASE permet de créer une base de données.

Syntaxe

```
CREATE DATABASE nom_base_de_données;
```

Le nom de base de données doit être unique dans le serveur ObjectServer et se conformer aux conventions de dénomination du serveur ObjectServer.

Une base de données est toujours persistante.

Exemple

```
create database mydb;
```

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

CREATE FILE

La commande CREATE FILE permet de créer un fichier ObjectServer.

Syntaxe

```
CREATE [ OR REPLACE ] FILE nom_fichier 'chemin_vers_fichier_physique'
[ MAXFILES nombre_fichiers ]
[ MAXSIZE taille_fichier { GBYTES | MBYTES | KBYTES | BYTES } ];
```

S'il est possible qu'un fichier ObjectServer porte déjà le même nom que celui que vous souhaitez créer, utilisez les mots clés facultatifs OR REPLACE. Si le fichier ObjectServer n'existe pas, un nouveau est créé. Si le fichier ObjectServer existe, il est remplacé par celui que vous créez.

Remarque : Si vous n'utilisez pas les mots clés OR REPLACE, vous devez indiquer un fichier physique qui n'existe pas déjà. Si vous utilisez les mots clés OR REPLACE et que le fichier physique existe déjà, ce dernier est écrasé si aucun fichier ObjectServer ne lui est associé.

Le nom de fichier doit être unique et conforme aux conventions de dénomination ObjectServer.

Le *chemin_vers_fichier_physique* est le chemin d'accès complet et le nom du fichier correspondant sur le système de fichiers physique (`/log/out.log`, par exemple). Sous Windows, vous devez doubler la barre oblique inversée (`\`) (`c:\\tmp\\testfile.txt`, par exemple) ou utiliser le chemin UNIX équivalent (`c:/tmp/testfile.txt`, par exemple).

Vous pouvez éventuellement attribuer une valeur à MAXFILES pour indiquer le nombre de fichiers de l'ensemble de fichiers. La valeur par défaut est 1. Si vous attribuez une valeur supérieure à 1 à MAXFILES, un fichier est créé lorsque la taille du premier fichier est supérieure à la taille maximale. Si la taille de ce fichier est supérieure à la taille maximale, un autre fichier est créé, le processus étant répété tant que le nombre maximal de fichiers n'a pas été atteint. Ensuite, le fichier le plus ancien est supprimé et le processus répété.

Remarque : Un nombre, commençant par 1 et incrémenté en fonction du nombre de fichiers présents dans l'ensemble de fichiers, est toujours ajouté au nom de fichier indiqué (ou à l'extension de fichier, le cas échéant).

Vous pouvez éventuellement attribuer une valeur à MAXSIZE pour indiquer la taille de fichier maximale. Si vous entrez dans le fichier un enregistrement supérieur ou égal à cette taille, un fichier est créé. Le paramètre par défaut est `0`. Si vous attribuez la valeur `0`, la taille de fichier est illimitée et l'ensemble est toujours composé d'un seul fichier.

La taille de fichier minimale est de 1 Ko. La taille maximale est de 4 Go.

Si le serveur ObjectServer est redémarré, de nouvelles données sont ajoutées au fichier existant.

Exemple

```
create file logit '/log/logfile'
maxfiles 3
maxsize 20 KBytes;
```

Si vous exécutez cet exemple de commande, la séquence de fichiers suivante est créée et utilisée :

1. Le serveur ObjectServer crée un fichier vide nommé `logfile1` dans le répertoire `/log`.
2. Le serveur ObjectServer écrit les données dans `logfile1` tant que la taille maximale de 20 Ko n'a pas été dépassée.
3. Le serveur ObjectServer renomme `logfile1` en `logfile2`. Ensuite, il crée un fichier `logfile1` et écrit des données tant que sa taille maximale n'a pas été dépassée.
4. Le serveur ObjectServer renomme respectivement `logfile2` et `logfile1` en `logfile3` et `logfile2`. Ensuite, il crée un fichier `logfile1` et écrit des données tant que sa taille maximale n'a pas été dépassée.
5. Le serveur ObjectServer supprime le plus ancien fichier (`logfile3`). Ensuite, il renomme respectivement `logfile2` et `logfile1` en `logfile3` et `logfile2`. Il crée un fichier nommé `logfile1` et écrit des données tant que sa taille maximale n'a pas été dépassée.

Cette séquence est répétée tant que le fichier n'a pas été modifié ou supprimé.

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

CREATE GROUP

La commande `CREATE GROUP` permet de créer un groupe d'un ou plusieurs utilisateurs.

Syntaxe

```
CREATE GROUP 'nom_groupe'
 [ ID identificateur ]
 [ COMMENT 'chaîne_commentaire' ]
 [ MEMBERS 'nom_utilisateur', ... ] ;
```

La variable *nom_groupe* est une chaîne de texte contenant le nom unique du groupe en cours de création. Ce nom peut contenir 64 caractères au maximum.

Remarque : Les noms de groupe sont sensibles à la casse et doivent être placés entre guillemets. Tout interlignage ou espace de fin est supprimé.

La variable *identificateur* est une valeur entière qui identifie le groupe de manière unique. Si vous ne spécifiez aucun identificateur, un identificateur est automatiquement attribué. Les identificateurs 0 à 7 sont réservés aux groupes de systèmes. Une valeur comprise entre 8 et 2147483647 peut être attribuée aux identificateurs des autres groupes.

Utilisez le paramètre facultatif COMMENT pour ajouter une description du groupe que vous créez.

Utilisez le mot clé MEMBERS pour indiquer les noms d'un ou de plusieurs utilisateurs que vous souhaitez ajouter en tant que membres du groupe.

Exemple

```
create group 'AutoAdmin' id 3 COMMENT 'Group to manage Automations'
   members 'joe', 'bob';
```

CREATE INDEX

La commande `CREATE INDEX` permet de créer un index sur une table de base de données.

Conseil : Des instructions de requête SQL et d'indexation sont disponibles pour vous aider à déterminer les colonnes à indexer et le type d'index à créer pour un colonne.

Syntaxe

```
CREATE INDEX nom_index
 ON nom_base_de_données.nom_table
 [USING { HASH | TREE }] (nom_colonne);
```

La valeur *nom_index* doit être unique dans le serveur ObjectServer et conforme aux conventions de dénomination du serveur ObjectServer. Pour faciliter l'identification et s'assurer du caractère unique, utilisez une convention de dénomination pour vos index (par exemple, *nom_colonne*Idx ou *nom_colonne*Index, par exemple, où *nom_colonne* est le nom de la colonne).

Le nom de table spécifié après le mot clé ON doit être intégralement qualifié avec le nom de base de données (alerts.status, par exemple).

Restriction : Vous ne pouvez pas créer d'index sur les colonnes des tables du système. Ces tables contiennent des métadonnées relatives aux objets du serveur ObjectServer et sont enregistrées dans la base de données de catalogue.

Utilisez le paramètre facultatif USING pour créer un index de hachage ou une arborescence d'index. S'il est ignoré, un index de hachage est créé par défaut. Un index de hachage est uniquement adapté aux requêtes SQL dénotant une égalité. Une arborescence d'index peut également être utilisée pour les requêtes triées.

Restriction : Vous ne pouvez pas créer d'index de hachage sur une seule zone de clé primaire. Vous ne pouvez pas créer d'arborescence d'index avec des valeurs de données booléennes.

Vous devez indiquer le nom de la colonne unique en cours d'indexation.

Exemple

```
create index SeverityIdx on alerts.status (Severity);
create index ExpireTimeIdx on alerts.status using tree (ExpireTime);
```

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

Référence associée:

«Instructions de requête SQL», à la page 333
Lorsqu'une instruction SQL est transmise au serveur ObjectServer, l'optimisation de requête et les plans de requête sont utilisés pour évaluer les méthodes disponibles d'accès aux/de modification des données et pour sélectionner la façon la plus efficace d'exécuter la requête.

«Instructions d'indexation», à la page 336
L'indexation peut affecter les performances de vos requêtes SQL. Sans indexation, une analyse de base de données complète est généralement effectuée lors de l'exécution d'une requête SQL. Utilisez l'indexation pour limiter le nombre de lignes à examiner.

«Tables de catalogues système», à la page 428
La base de données de catalogue contient les tables système créées et gérées par le serveur ObjectServer. Les tables système contiennent des métadonnées sur les objets du serveur ObjectServer.

CREATE PROCEDURE (procédures externes)

La commande CREATE PROCEDURE permet de créer des procédures externes.

Syntaxe

```
CREATE [ OR REPLACE ] PROCEDURE nom_procédure
 ( [ nom_paramètre
 { type_paramètre | ARRAY OF type_paramètre
    | ROW OF nom_base_de_données.nom_table },... ] )
 EXECUTABLE 'nom_exécutable'
 HOST 'nom_hôte'
 USER id_utilisateur
 GROUP id_groupe
 [ ARGUMENTS expression,... ] [;]
```

S'il est possible qu'une procédure porte déjà le même nom que celle que vous souhaitez créer, utilisez les mots clés facultatifs OR REPLACE. Si la procédure existe, elle est remplacée par celle que vous créez. Si la procédure n'existe pas, une nouvelle procédure est créée.

La valeur *nom_procédure* doit être unique dans le serveur ObjectServer et conforme aux conventions de dénomination du serveur ObjectServer.

Après la variable *nom_procédure*, insérez la déclaration de paramètre entre parenthèses ( ) pour spécifier les paramètres qui peuvent être transmis dans la procédure externe. Vous devez insérer les parenthèses après la variable

nom_procédure même si la procédure ne possède pas de paramètres. Chaque variable *nom_paramètre* doit être unique dans la procédure et doit se conformer aux conventions de dénomination du serveur ObjectServer.

Conseil : Les paramètres de procédure externes sont en lecture seule. Elles vous permettent de transmettre des valeurs de variable dans une procédure externe. Vous ne pouvez pas retourner les valeurs à partir d'une procédure externe.

La variable *type_paramètre* définit le type de données que le paramètre peut transmettre dans la procédure. Le type de données peut être tout type de données du serveur ObjectServer valide, à l'exception de VARCHAR ou de INCR.

La variable *nom_exécutable* est le chemin d'accès à un programme exécutable sur un système de fichiers local ou distant.

La variable *nom_hôte* est le nom de l'hôte sur lequel exécuter le programme exécutable pour la procédure.

La variable *id_utilisateur* est l'ID utilisateur effectif sous lequel exécuter le programme exécutable.

La variable *id_groupe* est le groupe effectif sous lequel exécuter le programme exécutable.

Les arguments sont ceux transmis au fichier exécutable. Seuls les espaces peuvent être utilisés pour séparer les arguments. Par exemple : `cool tool` est interprété comme `cool tool`, tandis que `cool'tool` ou `cool"tool` est interprété comme `cooltool`.

Exemple

La procédure externe suivante appelle un programme nommé **nco_mail**, qui envoie un e-mail sur les alertes critiques non reconnues :

```
create or replace procedure send_email
 (in node character(255), in severity integer, in subject character(255),
 in email character(255), in summary character(255), in hostname character(255))
 executable '$NCHOME/omnibus/utils/nco_mail'
 host 'localhost'
 user 0
 group 0
 arguments '\'' + node + '\'', severity, '\'' + subject + '\'',
  '\'' + email + '\'', '\"\''+summary+'\'\"';
```

Cet exemple présente également comment transmettre les chaînes de texte dans un programme exécutable. Vous devez insérer les chaînes entre guillemets et ignorer les guillemets à l'aide de barres obliques inverses. Tous les guillemets de cet exemple sont des guillemets simples.

Remarque : Pour exécuter une procédure externe, vous devez disposer d'un démon d'agent de contrôle de processus (**nco_pad**) exécuté sur l'hôte où la commande exécutable est stockée.

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

Chapitre 7, «Utilisation du contrôle de processus pour gérer des processus et des procédures externes», à la page 247
Le système de contrôle Tivoli Netcool/OMNIbus effectue deux tâches principales. Il gère les processus locaux et distants, puis exécute les processus externes spécifiés dans les automatisations.

«interface interactive SQL», à la page 229
Vous pouvez utiliser l'utilitaire de ligne de commande de l'interface interactive SQL fourni avec Tivoli Netcool/OMNIbus pour émettre des commandes SQL en direction d'un serveur ObjectServer.

Référence associée:

«Spécification des types de données des colonnes», à la page 171
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.

CREATE PROCEDURE (procédures SQL)

La commande CREATE PROCEDURE permet de créer des procédures SQL.

Cette commande définit la structure et le fonctionnement de la procédure, notamment les types de paramètre transmis entre les procédures, ainsi que les variables locales, le test de condition, les opérations sur la ligne et les affectations réalisées dans la procédure.

Syntaxe

```
CREATE [ OR REPLACE ] PROCEDURE nom_procédure
 ([
   [ IN | OUT | IN OUT ] nom_paramètre
  { type_paramètre | ARRAY OF type_paramètre }, ...
])
[ DECLARE déclaration_variable;...[;] ]
BEGIN
 instruction_corps_procédure;...[;]
END
```

S'il est possible qu'une procédure porte déjà le même nom que celle que vous souhaitez créer, utilisez les mots clés facultatifs OR REPLACE. Si la procédure existe, elle est remplacée par celle que vous créez. Si la procédure n'existe pas, une nouvelle procédure est créée.

La valeur *nom_procédure* doit être unique dans le serveur ObjectServer et conforme aux conventions de dénomination du serveur ObjectServer.

Après la variable *nom_procédure*, indiquez les paramètres qui peuvent être transmis entre les procédures, entre parenthèses ( ). Vous devez inclure les parenthèses après la variable *nom_procédure* même si la procédure ne comporte aucun paramètre.

Chaque paramètre de procédure est associé à un mode (IN, OUT ou IN OUT). Selon le mode que vous choisissez pour vos paramètres, vous pouvez les utiliser de différentes manières :

- Un paramètre IN est une variable accessible en lecture seule. Vous pouvez utiliser un paramètre IN dans des expressions afin de faciliter le calcul d'une valeur, mais vous ne pouvez pas attribuer de valeur au paramètre. Si vous ne souhaitez pas modifier la valeur d'une variable dans la procédure, utilisez un

paramètre IN pour transmettre la valeur de variable dans la procédure. Ce paramètre est utilisé par défaut si vous ne spécifiez pas le mode du paramètre.

- Un paramètre OUT est une variable accessible en écriture seulement. Vous pouvez utiliser un paramètre OUT pour attribuer une valeur au paramètre, mais vous ne pouvez pas le lire à partir du corps de la procédure. Par conséquent, vous ne pouvez pas utiliser ce type de paramètre dans une expression. Les paramètres OUT sont utiles pour transmettre des valeurs calculées à l'intérieur d'une procédure.
- Un paramètre IN OUT est une variable accessible en lecture et en écriture, sans les contraintes d'un paramètre IN ou OUT. Ce paramètre est utile pour les variables que vous souhaitez modifier à l'intérieur de la procédure et sortir de la procédure.

La variable *nom_paramètre* doit être unique dans la procédure et conforme aux conventions de dénomination du serveur ObjectServer.

La variable *type_paramètre* définit le type de données que le paramètre transmet ou reçoit de la procédure. Il peut s'agir d'un type de données ObjectServer valide, à l'exception de VARCHAR ou INCR.

`ARRAY OF` *`type_paramètre`* est un tableau des types de paramètre valides.

Dans la section facultative DECLARE d'une procédure, vous pouvez définir (déclarer) des variables locales à utiliser dans une procédure. Une variable locale est un espace réservé pour les valeurs utilisées lors de l'exécution de la procédure. Séparez les déclarations de variable locale par des points-virgules. Les noms de variable doivent être uniques dans la procédure et conformes aux conventions de dénomination du serveur ObjectServer. La variable *déclaration_variable* peut contenir les types de variable suivants :

- Variables simples :

  `nom_variable type_variable`
- Variables de tableau :

  `nom_variable` *`type_variable`* `[ ARRAY ]` **`[`** *`entier`* **`]`**

  Une variable *type_variable* est un type de données ObjectServer valide, à l'exception de VARCHAR ou INCR.

  Définissez la taille d'un tableau en spécifiant une valeur entière supérieure à 1 entre crochets.

  Remarque : Les crochets en gras qui encadrent la valeur entière sont nécessaires pour indiquer la taille du tableau. Ils n'indiquent pas la syntaxe d'un mot clé ou d'une clause facultative.

Le corps d'une procédure est encadré par les mots clés BEGIN et END. Vous pouvez utiliser les instructions SET, IF THEN ELSE et CASE WHEN et les boucles FOR EACH ROW et FOR dans le corps de la procédure.

Exemple

Dans la déclaration de procédure suivante, le paramètre formel est la variable `gravité_courante`. Lors de l'exécution de la procédure, vous transmettez un paramètre réel.

```
CREATE PROCEDURE calculate_average_severity
 ( IN current_severity INTEGER )
```

Par exemple, dans l'appel de procédure suivant, le paramètre effectif est la valeur 5, qui est attribuée au paramètre formel gravité_courante.

```
EXECUTE PROCEDURE calculate_average_severity(5);
```

Exemple

```
CREATE PROCEDURE add_or_concat
 ( IN counter INTEGER, IN one_char_string CHAR(1))
```

Exemple

Dans l'exemple suivant, un tableau de valeurs entières est transmis dans la procédure et utilisé pour calculer la gravité moyenne d'un sous-ensemble d'alertes :

```
CREATE PROCEDURE calculate_average_severity
 ( IN severity_arr ARRAY OF INTEGER)
```

Un tableau de valeurs entières est transmis dans la procédure lors de son exécution. Ces valeurs entières sont attribuées à un tableau nommé severity_arr.

Exemple

Pour créer une variable booléenne utilisée dans la procédure afin d'indiquer qu'une gravité a dépassé une valeur particulière, procédez comme suit :

```
DECLARE SeverityTooHigh BOOLEAN;
```

Pour créer un tableau de 20 valeurs entières utilisées dans la procédure pour enregistrer des noms de noeud, procédez comme suit :

```
DECLARE NodeNameArray INTEGER [20];
```

Concepts associés:

«Composants d'une procédure SQL», à la page 193
Les procédures SQL possèdent les composants principaux suivants : paramètres, déclarations de variable locale et corps de procédure.

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

«Expressions», à la page 227
Une expression est une combinaison syntaxique de valeurs et d'opérations associées pour calculer de nouvelles valeurs. Les expressions peuvent être simples ou complexes.

Référence associée:

«Spécification des types de données des colonnes», à la page 171
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.

«Construction d'une instruction de corps de procédure SQL», à la page 193
Le corps d'une procédure SQL contient un ensemble d'instructions SQL et de constructions de programmation qui gèrent les données du serveur ObjectServer.

CREATE RESTRICTION FILTER

La commande CREATE RESTRICTION FILTER permet de créer un filtre de restriction.

Syntaxe

```
CREATE [ OR REPLACE ] RESTRICTION FILTER nom_filtre
ON nom_base_de_données.nom_table WHERE condition;
```

S'il est possible qu'un filtre de restriction porte déjà le même nom que celui que vous souhaitez créer, utilisez les mots clés facultatifs OR REPLACE. Si le filtre de restriction existe, il est remplacé par celui que vous créez. Si le filtre de restriction n'existe pas, un nouveau est créé.

Remarque : Si vous remplacez un filtre de restriction existant, seule la *condition* peut être remplacée. Un filtre peut être remplacé même s'il a été attribué à des utilisateurs ou des groupes.

Le nom du filtre de restriction doit être unique et conforme aux conventions de dénomination ObjectServer.

Le nom de table spécifié après le mot clé ON doit être intégralement qualifié avec le nom de base de données (alerts.status, par exemple).

La *condition* est composée d'une ou de plusieurs expressions qui renvoient un sous-ensemble de lignes de la table. Le cas échéant, vous devez indiquer des noms de table qualifiés complets dans la clause WHERE et les instructions SELECT de la condition. Utilisez le format *nom_base_de_données.nom_table* pour indiquer le nom de table qualifié complet.

Un filtre de restriction est toujours permanent et recréé à chaque redémarrage du serveur ObjectServer.

Exemple

```
create restriction filter myfilter on alerts.status where Severity = 5;
```

Conseil : Vous pouvez créer des filtres de restriction dans le générateur de filtres.

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

CREATE ROLE

La commande CREATE ROLE permet de créer un rôle qui est une collection d'autorisations.

Syntaxe

```
CREATE ROLE 'nom_rôle'
 [ ID identificateur ]
 [ COMMENT 'chaîne_commentaire' ];
```

La variable *nom_rôle* est une chaîne de texte contenant le nom unique du rôle en cours de création. Ce nom peut contenir 64 caractères au maximum.

Remarque : Les noms de rôle sont sensibles à la casse et doivent être placés entre guillemets. Tous les espaces de gauche et de droite sont supprimés.

La variable *identificateur* est une valeur entière qui identifie le rôle de manière unique. Si vous ne spécifiez aucun identificateur, un identificateur est automatiquement attribué. Le rôle Normal est associé à l'identificateur 3. Le rôle Administrateur est associé à l'identificateur 2. Le rôle Utilisateur root est associé à l'identificateur -1 et détient les droits d'accès à tous les objets. Une valeur comprise entre 13 et 2147483647 peut être attribuée aux identificateurs des autres rôles.

Utilisez le paramètre facultatif COMMENT pour ajouter une description du rôle que vous créez.

Des rôles par défaut sont créés par le script security.sql.

Exemple

```
create role 'SuperAdmin' id 500
comment 'only users with root access should be granted this role';
```

Référence associée:

«GRANT ROLE», à la page 394
Une fois que vous avez créé les rôles en tant que collectes d'autorisations, vous pouvez attribuer les rôles aux groupes et révoquer les rôles des groupes. Utilisez la commande GRANT ROLE pour affecter des rôles aux groupes.

CREATE SIGNAL

La commande CREATE SIGNAL permet de créer un signal défini par l'utilisateur. Il s'agit de définir une liste d'attributs de type données.

Syntaxe

```
CREATE [ OR REPLACE ] SIGNAL nom_signal
 [ (nom_attribut_signal type_données,...) ]
 [ COMMENT 'chaîne_commentaire' ]
```

Le nom du signal doit être unique dans le serveur ObjectServer et conforme aux conventions de dénomination ObjectServer. Vous ne pouvez pas créer un signal défini par l'utilisateur portant le même nom qu'un signal du système.

Lorsque vous définissez des attributs, spécifiez le nom d'attribut et le type de données ObjectServer valide, à l'exception de VARCHAR ou INCR.

Vous pouvez ajouter un commentaire à la suite du mot clé facultatif COMMENT.

Exemple

Pour créer un signal appelé `illegal_delete` avec deux attributs de chaîne de caractères, `user_name` et `row_summary`, utilisez la commande suivante :

```
CREATE SIGNAL illegal_delete( user_name char(40), row_summary char(255) );
```

Ensuite, vous pouvez créer un déclencheur (le déclencheur de base de données d'insertion préalable suivant, par exemple) pour interrompre les suppressions qui ont lieu en dehors des heures de bureau normales et émettre ce signal.

```
create trigger DETECT_AN_ILLEGAL_DELETE
  group default_triggers
  priority 1
  before delete on alerts.status
  for each row
  begin
   if( ( (hourofday() > 17) and (minuteofhour() > 30) ) or (hourofday() < 9) ) then
      raise signal ILLEGAL_DELETE %user.user_name, old.Summary;
         cancel;
      end if;
 end;
```

Le déclencheur de signal suivant défini par l'utilisateur, déclenché par le déclencheur de base de données précédent, permet d'exécuter une procédure externe d'envoi d'une notification par courrier électronique de l'opération de suppression tentée.

```
create trigger AFTER_HOURS_DELETE_WARNING
  group default_triggers
  priority 1
  on signal ILLEGAL_DELETE
  begin
 execute MAIL_THE_BOSS( 'User ' + '%signal.user_name ' +
'attempted to remove the row ' + %signal.row_summary + ' at ' +to_char(getdate) )
  end;
```

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

Référence associée:

«Spécification des types de données des colonnes», à la page 171
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.

CREATE TABLE

La commande `CREATE TABLE` permet de créer une table.

Syntaxe

```
CREATE TABLE [nom_base_de_données.]nom_table
PERSISTENT | VIRTUAL
(nom_colonne type_données [ PRIMARY KEY | NODEFAULT | NOMODIFY | HIDDEN],...
[, PRIMARY KEY(nom_colonne,...) ] );
```

Le nom de table doit être unique dans la base de données et conforme aux conventions de dénomination ObjectServer.

Le type de stockage est PERSISTENT ou VIRTUAL. Une table permanente est recréée et remplie avec toutes ses données au redémarrage du serveur ObjectServer. Une table virtuelle est recréée avec la même description de table, mais sans données, au redémarrage du serveur ObjectServer.

Lors de la définition des colonnes, vous devez spécifier le nom de colonne et le type de données, et pouvez également préciser les propriétés facultatives.

Une table peut contenir au maximum 512 colonnes, à l'exclusion des colonnes gérées par le système. La taille de ligne maximale d'une table, qui est la somme des longueurs de colonne de la ligne, est de 64 Ko.

Exemple

```
create table mydb.mytab persistent
(col1 integer primary key, col2 varchar(20));
```

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

CREATE TRIGGER (déclencheur de base de données)

La commande CREATE TRIGGER permet de créer les déclencheurs de base de données qui s'activent lorsqu'une modification ou une tentative de modification dans une table du serveur ObjectServer survient (ou lorsqu'une modification ou une tentative de modification dans une vue affecte une table de base).

Syntaxe

```
CREATE [ OR REPLACE ] TRIGGER nom_déclencheur
 GROUP nom_groupe
 [ DEBUG { TRUE | FALSE } ]
 [ ENABLED { TRUE | FALSE } ]
 PRIORITY entier
 [ COMMENT 'chaîne_commentaire' ]
 { BEFORE | AFTER } { INSERT | UPDATE | DELETE | REINSERT }
 ON nom_base_de_données.nom_table
 FOR EACH { ROW | STATEMENT }
 [ WHEN condition ]
 [ DECLARE déclaration_variable ]
 BEGIN
 action_déclencheur
END;
```

S'il est possible qu'un déclencheur existe déjà avec le même nom que celui que vous souhaitez créer, utilisez les mots clés facultatifs OR REPLACE. Si le déclencheur existe, il est remplacé par celui que vous créez. Si le déclencheur n'existe pas, un nouveau est créé.

La valeur *nom_déclencheur* doit être unique dans le serveur ObjectServer et respecter les conventions de dénomination du serveur ObjectServer.

La valeur *nom_groupe* peut être tout groupe de déclencheurs déjà créé à l'aide de la commande CREATE TRIGGER GROUP.

Si DEBUG est défini sur TRUE, les informations de débogage sont envoyées au journal des messages du serveur ObjectServer si le niveau du message est debug.

Si ENABLED est défini sur TRUE, le déclencheur s'exécute lorsque le problème associé se produit. Sinon, le déclencheur ne s'exécute pas lorsque l'incident se produit.

La PRIORITE d'un déclencheur détermine l'ordre dans lequel le serveur ObjectServer exécute les déclencheurs lorsque plusieurs d'entre eux sont associés au même incident. Cette priorité est comprise dans un intervalle allant de 1 à 20. Plus le nombre est bas, plus la priorité est haute. Ainsi, un déclencheur de priorité de niveau 2 s'exécute avant un déclencheur de priorité de niveau 3. Si plusieurs déclencheurs de même priorité s'exécutent en raison d'un même incident, l'ordre dans lequel ils s'exécutent est indéterminé.

Utilisez le mot clé facultatif COMMENT pour ajouter un commentaire (*chaîne_commentaire*) au déclencheur.

Le mot clé de temporisation BEFORE ou AFTER indique si le déclencheur s'exécute avant ou après la modification de la base de données qui a activé le déclencheur. Par exemple, vous pouvez créer un déclencheur BEFORE qui détermine le nom de l'utilisateur avant qu'une ligne de la table alerts.status ne soit supprimée. Dans le déclencheur, vous pouvez définir si l'utilisateur est autorisé à effectuer des suppressions à partir de la table alerts.status. Si ce n'est pas le cas, empêchez la modification de base de données de survenir. Avec un déclencheur AFTER, la modification de base de données a toujours lieu.

nom_base_de_données.nom_table est le nom de la base de données et de la table affectées par l'action du déclencheur.

Un déclencheur de base de données s'active aux niveaux suivants :

- FOR EACH ROW (*déclencheur au niveau de la ligne*) : les déclencheurs au niveau de la ligne ne s'activent qu'une fois pour chaque ligne retournée suite à une modification de la base de données.
- FOR EACH STATEMENT (*déclencheur au niveau de l'instruction*) : les déclencheurs au niveau de l'instruction s'activent une fois pour chaque modification de base de données.

Remarque : Seuls les déclencheurs au niveau de la ligne peuvent être définis pour s'activer au niveau des insertions ou des réinsertions.

Remarque : Les déclencheurs au niveau de l'instruction BEFORE s'activent toujours avant les déclencheurs au niveau de la ligne BEFORE et les déclencheurs au niveau de l'instruction AFTER s'activent toujours après les déclencheurs au niveau de la ligne AFTER, quelle que soit la priorité du déclencheur.

Utilisez la clause facultative WHEN pour tester une *condition* particulière avant que l'action du déclencheur ne s'exécute. Si la condition n'est pas remplie, l'action du déclencheur ne s'exécute pas.

Vous pouvez, si vous le souhaitez, déclarer des variables locales de déclencheur à utiliser dans le corps du déclencheur. Ces variables sont déclarées et utilisées de la même manière que les variables de procédure. Cependant, les variables de déclencheur sont statiques, de sorte qu'elles conservent leur valeur entre les différentes exécutions du déclencheur.

Exemple

Un signal de base de données est mis au premier plan en tant que résultat de l'instruction SQL suivante :

```
DELETE FROM alerts.status WHERE Severity = 5;
```

Lorsque cette instruction est exécutée, le serveur ObjectServer supprime toutes les lignes de la table alerts.status possédant un niveau de gravité de 5. Si la table contient 20 lignes de gravité 5 et que le niveau est défini sur FOR EACH ROW, 20 lignes sont supprimées et le déclencheur est mis au premier plan 20 fois. Si `level` est défini sur FOR EACH STATEMENT, le déclencheur est mis au premier plan une fois.

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

Référence associée:

«Exécution de commandes dans les actions du déclencheur», à la page 177
La variable *action_déclencheur* contient un ensemble de commandes permettant de manipuler les données dans le serveur ObjectServer.

«Valeurs recommandées lors de la création de déclencheurs», à la page 340
Lorsque vous créez ou modifiez des déclencheurs, vérifiez que ces derniers sont aussi efficaces que possible et qu'ils possèdent le plus faible temps d'exécution possible.

CREATE TRIGGER (déclencheur de signal)

La commande CREATE TRIGGER permet de créer un déclencheur de signal qui s'active en réponse à des incidents dans le serveur ObjectServer ou en réponse à un signal défini par l'utilisateur.

Syntaxe

```
CREATE [ OR REPLACE ] TRIGGER nom_déclencheur
 GROUP nom_groupe
 [ DEBUG { TRUE | FALSE } ]
 [ ENABLED { TRUE | FALSE } ]
 PRIORITY entier
 [ COMMENT 'chaîne_commentaire' ]
 ON SIGNAL { nom_signal_système | nom_signal_utilisateur }
 [ EVALUATE SELECT_cmd BIND AS nom_variable ]
 [ WHEN condition ]
 [ DECLARE déclaration_variable ]
 BEGIN
action_déclencheur
END;
```

S'il est possible qu'un déclencheur existe déjà avec le même nom que celui que vous souhaitez créer, utilisez les mots clés facultatifs OR REPLACE. Si le déclencheur existe, il est remplacé par celui que vous créez. Si le déclencheur n'existe pas, un nouveau est créé.

La valeur *nom_déclencheur* doit être unique dans le serveur ObjectServer et respecter les conventions de dénomination du serveur ObjectServer.

La valeur *nom_groupe* peut être tout groupe de déclencheurs déjà créé à l'aide de la commande CREATE TRIGGER GROUP.

Si DEBUG est défini sur `TRUE`, les informations de débogage sont envoyées au journal des messages du serveur ObjectServer si le niveau du message est `debug`.

Si ENABLED est défini sur `TRUE`, le déclencheur s'exécute lorsque le problème associé se produit. Sinon, le déclencheur ne s'exécute pas lorsque l'incident se produit.

La PRIORITE d'un déclencheur détermine l'ordre dans lequel le serveur ObjectServer exécute les déclencheurs lorsque plusieurs d'entre eux sont associés au même incident. Cette priorité est comprise dans un intervalle allant de 1 à 20. Plus le nombre est bas, plus la priorité est haute. Ainsi, un déclencheur de priorité de niveau 2 s'exécute avant un déclencheur de priorité de niveau 3. Si plusieurs déclencheurs de même priorité s'exécutent en raison d'un même incident, l'ordre dans lequel ils s'exécutent est indéterminé.

Utilisez le mot clé facultatif COMMENT pour ajouter un commentaire (*chaîne_commentaire*) au déclencheur.

Le nom ON SIGNAL peut être identique au signal du système ou au signal défini par l'utilisateur qui met le déclencheur en application.

La clause facultative EVALUATE permet de générer un ensemble de résultats temporaire défini à partir d'une seule instruction SELECT traitée dans *action_déclencheur*. L'instruction SELECT ne peut pas contenir de clause ORDER BY.

Remarque : La clause EVALUATE doit intégralement qualifier les tables (qui sont incluses dans l'instruction SELECT) avec un nom de base de données. Par exemple, la syntaxe `evaluate select Node from status...` est supposée non valide, la syntaxe correcte étant `evaluate select Node from alerts.status...`

Si un signal système ou un signal défini par l'utilisateur est émis, les attributs qui identifient la cause sont associés au signal. Ces attributs sont transmis sous forme de variables implicites dans le déclencheur de signal associé.

Utilisez la clause facultative WHEN pour tester une *condition* particulière avant que l'action du déclencheur ne s'exécute. Si la condition n'est pas remplie, l'action du déclencheur ne s'exécute pas.

Vous pouvez, si vous le souhaitez, déclarer des variables locales de déclencheur à utiliser dans le corps du déclencheur. Ces variables sont déclarées et utilisées de la même manière que les variables de procédure. Cependant, les variables de déclencheur sont statiques, de sorte qu'elles conservent leur valeur entre les différentes exécutions du déclencheur.

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

Référence associée:

«Signaux système et leurs attributs», à la page 181
Si un signal système est émis, les attributs qui identifient la cause sont définis. Ces attributs sont transmis sous forme de variables implicites dans le déclencheur de signal associé.

«Valeurs recommandées lors de la création de déclencheurs», à la page 340
Lorsque vous créez ou modifiez des déclencheurs, vérifiez que ces derniers sont aussi efficaces que possible et qu'ils possèdent le plus faible temps d'exécution possible.

CREATE TRIGGER (déclencheur temporel)

La commande CREATE TRIGGER permet de créer des délencheurs temporels qui s'activent selon une fréquence spécifiée.

Syntaxe

```
CREATE [ OR REPLACE ] TRIGGER nom_déclencheur
 GROUP nom_groupe
 [ DEBUG { TRUE | FALSE } ]
 [ ENABLED { TRUE | FALSE } ]
 PRIORITY integer
 [ COMMENT 'chaîne_commentaire'
 EVERY entier { HOURS | MINUTES | SECONDS }
 [ EVALUATE commande_SELECT BIND AS nom_variable ]
 [ WHEN condition ]
 [ DECLARE déclaration_variable ]
BEGIN
action_déclencheur
END;
```

S'il est possible qu'un déclencheur existe déjà avec le même nom que celui que vous souhaitez créer, utilisez les mots clés facultatifs OR REPLACE. Si le déclencheur existe, il est remplacé par celui que vous créez. Si le déclencheur n'existe pas, un nouveau est créé.

La valeur *nom_déclencheur* doit être unique dans le serveur ObjectServer et respecter les conventions de dénomination du serveur ObjectServer.

La valeur *nom_groupe* peut être tout groupe de déclencheurs déjà créé à l'aide de la commande CREATE TRIGGER GROUP.

Si DEBUG est défini sur TRUE, les informations de débogage sont envoyées au journal des messages du serveur ObjectServer si le niveau du message est debug.

Si ENABLED est défini sur TRUE, le déclencheur s'exécute lorsque le problème associé se produit. Sinon, le déclencheur ne s'exécute pas lorsque l'incident se produit.

La PRIORITE d'un déclencheur détermine l'ordre dans lequel le serveur ObjectServer exécute les déclencheurs lorsque plusieurs d'entre eux sont associés au même incident. Cette priorité est comprise dans un intervalle allant de 1 à 20. Plus le nombre est bas, plus la priorité est haute. Ainsi, un déclencheur de priorité de niveau 2 s'exécute avant un déclencheur de priorité de niveau 3. Si plusieurs déclencheurs de même priorité s'exécutent en raison d'un même incident, l'ordre dans lequel ils s'exécutent est indéterminé.

Utilisez le mot clé facultatif COMMENT pour ajouter un commentaire (*chaîne_commentaire*) au déclencheur.

A l'intérieur d'un déclencheur temporel, vous devez préciser la fréquence de déclenchement du déclencheur. Indiquez un *entier* exprimé en secondes (unité de temps par défaut), minutes ou heures.

Utilisez la clause facultative EVALUATE pour générer un ensemble de résultats temporaire défini à partir d'une seule instruction SELECT traitée dans *action_déclencheur*. L'instruction SELECT ne peut pas contenir de clause ORDER BY.

Remarque : La clause EVALUATE doit intégralement qualifier les tables (qui sont incluses dans l'instruction SELECT) avec un nom de base de données. Par exemple, la syntaxe `evaluate select Node from status...` est supposée non valide, la syntaxe correcte étant `evaluate select Node from alerts.status...`

Dans la plupart des cas, une clause EVALUATE peut être remplacée par une clause FOR EACH ROW. N'utilisez une clause EVALUATE que lorsqu'une clause GROUP BY est obligatoire.

Utilisez la clause facultative WHEN pour tester une *condition* particulière avant que l'action du déclencheur ne s'exécute. Si la condition n'est pas remplie, l'action du déclencheur ne s'exécute pas.

Vous pouvez, si vous le souhaitez, déclarer des variables locales de déclencheur à utiliser dans le corps du déclencheur. Ces variables sont déclarées et utilisées de la même manière que les variables de procédure. Cependant, les variables de déclencheur sont statiques, de sorte qu'elles conservent leur valeur entre les différentes exécutions du déclencheur.

Exemple

Le déclencheur temporel suivant permet de supprimer toutes les lignes en clair (`Severity = 0`) de la table alerts.status qui n'ont pas été modifiées dans les deux dernières minutes.

```
create trigger DeleteClears
group my_triggers
priority 1
every 60 seconds
begin
   delete from alerts.status where Severity = 0
      and StateChange < (getdate - 120);
end;
```

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

Référence associée:

«Exécution de commandes dans les actions du déclencheur», à la page 177
La variable *action_déclencheur* contient un ensemble de commandes permettant de manipuler les données dans le serveur ObjectServer.

«Valeurs recommandées lors de la création de déclencheurs», à la page 340
Lorsque vous créez ou modifiez des déclencheurs, vérifiez que ces derniers sont aussi efficaces que possible et qu'ils possèdent le plus faible temps d'exécution possible.

CREATE TRIGGER GROUP

La commande CREATE TRIGGER GROUP permet de créer un nouveau groupe de déclencheurs. Lorsque vous créez un déclencheur, vous devez l'attribuer à un groupe de déclencheurs.

Syntaxe

```
CREATE TRIGGER GROUP nom_groupe_déclencheurs;
```

La valeur *nom_groupe_déclencheurs* doit être unique dans le serveur ObjectServer et conforme aux conventions de dénomination du serveur ObjectServer.

Exemple

```
create trigger group update_database_triggers;
```

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

CREATE USER

La commande CREATE USER permet d'ajouter un utilisateur au serveur ObjectServer.

Syntaxe

```
CREATE USER 'nom_utilisateur'
 [ ID identificateur ]
 FULL NAME 'nom_utilisateur_complet'
 [ PASSWORD 'mot_de_passe' [ ENCRYPTED] ]
 [ PAM { TRUE | FALSE } ];
```

La variable *nom_utilisateur* est une chaîne de texte contenant le nom unique de l'utilisateur que vous ajoutez. Ce nom peut contenir 64 caractères au maximum. Si l'utilisateur doit être authentifié en externe (dans un référentiel LDAP ou à l'aide d'un module PAM, par exemple), indiquez le nom d'utilisateur enregistré dans le référentiel d'authentification externe.

Remarque : Les noms d'utilisateur sont sensibles à la casse et doivent être placés entre guillemets. Tous les espaces de gauche et de droite sont supprimés.

La variable *identificateur* est une valeur entière qui identifie l'utilisateur de manière unique. Si vous ne spécifiez aucun identificateur, un identificateur est automatiquement attribué. L'identificateur de l'utilisateur root est 0. Celui de l'utilisateur nobody est 65534. Une valeur comprise entre 1 et 2147483647 peut être attribuée aux identificateurs des autres utilisateurs.

La variable *nom_utilisateur_complet* est une chaîne de texte contenant le nom complet de l'utilisateur.

Vous pouvez indiquer le mot de passe de l'utilisateur utilisant le mot clé PASSWORD. La valeur par défaut est une chaîne vide. Si vous ajoutez le mot clé ENCRYPTED, le mot de passe est supposé codé. Aucun mot de passe n'est requis pour un utilisateur authentifié en externe.

Pour indiquer que l'utilisateur est authentifié en externe, attribuez la valeur TRUE à PAM. La valeur PAM ou LDAP doit également être attribuée à la propriété

ObjectServer **Sec.ExternalAuthentication**, selon le système d'authentification. Si la valeur FALSE est attribuée à PAM ou none à **Sec.ExternalAuthentication**, l'utilisateur ne peut pas être authentifié en externe. Si vous souhaitez enregistrer le nom d'utilisateur et le mot de passe associé dans le serveur ObjectServer, et que vous procédez à une authentification ObjectServer, attribuez la valeur FALSE à PAM. Pour plus d'informations relatives à PAM ou LDAP, voir *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

Exemple

```
create user 'joe' id 1 full name 'Joseph R. User';
```

Référence associée:

«Propriétés et options de ligne de commande du serveur ObjectServer», à la page 4
Utilisez les propriétés ou les options de ligne de commande ObjectServer pour configurer les paramètres du serveur ObjectServer. Pour éviter les erreurs, ajoutez autant de propriétés que possible au fichier de propriétés plutôt que des options de ligne de commande. D'autres utilitaires sont fournis que vous pouvez utiliser pour chiffrer les valeurs de propriété.

CREATE VIEW

La commande CREATE VIEW permet de créer une vue.

Syntaxe

```
CREATE [ OR REPLACE ] VIEW [nom_base_de_données.]nom_vue
[ (nom_colonne_vue,...) ]
[ TRANSIENT | PERSISTENT ]
AS commande_SELECT;
```

S'il est possible qu'une vue porte déjà le même nom que celle que vous souhaitez créer, utilisez les mots clés facultatifs OR REPLACE. Si la vue existe, elle est remplacée par celle que vous créez. Si la vue n'existe pas, une nouvelle est créée.

Le nom de la vue doit être unique dans la base de données et conforme aux conventions de dénomination ObjectServer. Les restrictions supplémentaires suivantes s'appliquent à la création des vues :

- Si vous n'indiquez pas de nom de base de données, la vue est créée dans la base de données des alertes.
- Vous ne pouvez créer une vue sur une autre vue.
- Vous ne pouvez pas créer une vue sur une table de la base de données de catalogues.

Vous pouvez spécifier un type de stockage TRANSIENT ou PERSISTENT, selon vos exigences en la matière. Une vue transitoire est détruite lorsque le client qui l'a créée se déconnecte. Une vue permanente est copiée sur le disque. Au démarrage du serveur ObjectServer, la vue est recréée.

La variable *commande_SELECT* est une commande SELECT (y compris les commandes SELECT agrégées), avec les restrictions suivantes :

- Vous devez spécifier tous les noms de colonne de manière explicite, plutôt que d'utiliser un caractère générique (*), dans la liste de sélection.
- Si vous intégrez des colonnes virtuelles, vous ne pouvez pas les mettre à jour.
- Si vous ne spécifiez pas de nom de base de données, la valeur par défaut est alerts.
- Vous ne pouvez pas spécifier de clause GROUP BY.

- Vous pouvez uniquement détenir une sous-requête contenant une clause WHERE dans une instruction SELECT agrégée.
- Vous ne pouvez pas utiliser de colonnes virtuelles dans une instruction SELECT agrégée.
- Si vous créez une vue des données globales, elle ne peut pas faire l'objet d'une instruction SELECT agrégée.

Exemple

```
create view alerts.myview persistent as select Severity, LastOccurrence, Summary
from alerts.status order by Severity, LastOccurrence;
```

Concepts associés:

«Conventions de dénomination SQL ObjectServer», à la page 213
Lorsque vous émettez des commandes SQL, vous devez vous conformer aux conventions de dénomination ObjectServer.

Référence associée:

«SELECT (scalaire)», à la page 401
La commande scalaire SELECT extrait les colonnes et lignes d'une table en fonction de critères spécifiques.

«SELECT (aggregate)», à la page 403
Une commande SELECT d'agrégation calcule le nombre de lignes et renvoie une valeur unique.

«SELECT (GROUP BY)», à la page 405
Vous pouvez utiliser une commande SELECT avec une clause GROUP BY pour regrouper en une seule ligne toutes les lignes ayant des valeurs identiques dans une colonne ou une combinaison de colonnes spécifiques. Vous pouvez également rechercher la valeur agrégée de chaque groupe de valeurs de colonne.

DELETE

La commande DELETE permet de supprimer une ou plusieurs lignes de données d'une table existante.

Syntaxe

```
DELETE FROM [nom_base_de_données.]nom_objet
 [ VIA valeur_colonne_clé_principale,... ]
 [ WHERE condition ];
```

Utilisation de la clause WHERE

Si vous incluez une clause WHERE, seules les lignes satisfaisant aux critères spécifiés dans la variable *condition* sont mises à jour. Si aucune condition n'est spécifiée dans la clause WHERE, toutes les lignes sont mises à jour.

La clause WHERE peut effectuer les mêmes fonctions que la clause VIA dans les versions antérieures de Tivoli Netcool/OMNIbus. Si la clé primaire est spécifiée dans la clause WHERE, le serveur ObjectServer n'effectue pas d'analyse complète de la table mais accède directement aux lignes en utilisant la clé primaire.

L'exemple suivant supprime toutes les lignes de la table alerts.status dans lesquelles la valeur de la zone Node est égale à Fred:

```
delete from alerts.status where Node = 'Fred';
```

Utilisation de la clause VIA

La clause VIA a été abandonnée avec Tivoli Netcool/OMNIbus 7.2 mais elle reste valide. Ses fonctions peuvent désormais être effectuées par la clause WHERE.

Si vous connaissez la valeur de la clé primaire de la ligne que vous souhaitez supprimer, vous pouvez la spécifier à l'aide de la clause VIA. En présence de plusieurs colonnes de clé principale, les valeurs doivent être spécifiées dans l'ordre et séparées par des virgules (,). Les valeurs de chaîne doivent être placées entre guillemets simples (').

Si plusieurs lignes sont mises à jour, la ou les clés principales de chaque ligne doivent être placées entre crochets ([ ]) et séparées par des virgules (,).

La clause VIA peut être utile pour documenter votre SQL. Par exemple, vous pouvez l'utiliser pour faire savoir au programme de maintenance des déclencheurs que la commande DELETE est exécutée en utilisant des clés primaires.

L'exemple suivant indique comment supprimer plusieurs clés primaires à l'aide de la commande DELETE VIA :

```
delete from alerts.status via ['722Identifier722'],['294Identifier294'];
```

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

DESCRIBE

La commande DESCRIBE permet d'afficher des informations sur les colonnes de la table ou de la vue spécifiée.

Syntaxe

```
DESCRIBE [nom_base_de_données.]nom_objet;
```

La sortie de cette commande indique le nom de colonne, le type de données (renvoyé en tant qu'ID ObjectServer), la longueur de la colonne et si la colonne fait partie d'une clé primaire (1 si c'est le cas, 0 dans le cas contraire).

Les colonnes masquées ne s'affichent pas car elles sont gérées par le système, un utilisateur classique n'ayant pas besoin de les afficher ou de les mettre à jour.

Exemple

Utilisez la commande suivante pour afficher les informations relatives aux colonnes de la table catalog.tables :

```
describe catalog.tables;
```

Voici un exemple de sortie de la commande précédente :

```
ColumnName          Type      Size     Key
------------------- --------- -------- ------------
TableName                   2      40           1
DatabaseName                2      40           1
Status                      0       4           0
NumDependents              12       4           0
TableID                     0       4           0
```

```
TableKind                   0       4           0
StorageKind                 0       4           0
ServerID                    0       4           0
```

Référence associée:

«Spécification des types de données des colonnes», à la page 171
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.

DROP COLUMN

Pour supprimer des colonnes d'une table existante, utilisez le paramètre `DROP COLUMN` avec la commande `ALTER TABLE`.

Dans cette commande, la syntaxe de suppression de colonnes se présente comme suit :

```
DROP [COLUMN] nom_colonne
```

Vous ne pouvez pas supprimer une colonne s'il s'agit d'une clé primaire.

Lors de la suppression d'une colonne, il est nécessaire de réaliser un certain nombre d'actions préliminaires afin d'identifier et de supprimer les dépendances externes dont fait l'objet la colonne. Vous devez rechercher des références à la colonne dans les déclencheurs, les procédures, les vues et les filtres de restriction en analysant les tables de base de données correspondantes. Vous devez également rechercher des références à la colonne dans vos fichiers de règles de sonde et fichiers de mappage de passerelle.

Avertissement : Si vous supprimez une colonne de laquelle dépendent des déclencheurs, des procédures, des vues, des filtres de restriction ou des index, ces objets dépendants sont également supprimés et un avertissement est écrit dans le fichier journal du serveur ObjectServer. Pour éviter la suppression accidentelle des déclencheurs, procédures, vues ou filtres de restriction, lisez les instructions ci-dessous relatives à la suppression des colonnes (étant donné que les index sont directement liés aux colonnes, ils sont toujours supprimés lors de la suppression de leurs colonnes associées).

Les instructions ci-après reposent sur un exemple de scénario dans lequel vous souhaitez supprimer la colonne `Country` d'une table du serveur ObjectServer :

1. Connectez-vous au serveur ObjectServer (OWL, par exemple) à l'aide de l'interface interactive SQL (voir le tableau ci-dessous). Votre nom d'utilisateur est supposé être celui par défaut, mais vous êtes invité à entrer votre mot de passe.

Tableau 103. Démarrage de l'interface interactive SQL

Option	Description
UNIX	Entrez : `$NCHOME/omnibus/bin/nco_sql -server OWL`
Windows	Entrez : `%NCHOME%\omnibus\bin\isql -S OWL`

2. Sauvegardez le serveur ObjectServer dans un emplacement temporaire (/tmp/mybackup, par exemple) en utilisant la commande ALTER SYSTEM BACKUP. Cette mesure de précaution garantit la restauration de votre système, le cas échéant.

   ```
   1> alter system backup '/tmp/mybackup';
   2> go
   ```

3. Répertoriez les détails de vos déclencheurs, tels qu'ils figurent dans la table catalog.triggers :

   ```
   1> describe catalog.triggers;
   2> go
   ```

   Le type de clé, le nom, le type de données et la longueur de chaque colonne de la table s'affichent à l'écran.

4. Extrayez les noms de tous les déclencheurs faisant référence à la colonne Country dans le corps ou la clause Evaluate du déclencheur :

   ```
   1> select TriggerName from catalog.triggers where CodeBlock like
   ' Country' or EvaluateBlock like ' Country';
   ```

5. Notez tous les déclencheurs répertoriés, puis supprimez les références Country en éditant chaque déclencheur. Pour ce faire, utilisez la fenêtre Trigger Details (Détails du déclencheur) (onglet **Action**) de Netcool/OMNIbus Administrator.

   Important : Avant d'effectuer les modifications décrites dans les étapes restantes de cette procédure, vérifiez que les conditions suivantes sont réunies :

   - Il peut arriver que plusieurs tables utilisent le même nom de colonne dans votre schéma. Si tel est le cas, identifiez les déclencheurs de la table que vous souhaitez supprimer avant d'effectuer les modifications.
   - Vérifiez qu'il n'existe aucune dépendance entre les déclencheurs qui pourrait provoquer des modifications non souhaitées.

6. Répétez les étapes 3 à 5 afin d'identifier tous les objets faisant référence à la colonne Country et pour supprimer toutes les instances de la référence. Le tableau ci-dessous répertorie les tables de base de données que vous devez rechercher, les instructions SELECT correspondantes et les fenêtres de Netcool/OMNIbus Administrator que vous pouvez utiliser pour éditer l'objet.

Tableau 104. Tables du catalogue système à rechercher, instructions SELECT et fenêtre de Netcool/OMNIbus Administrator

Type d'objet	Nom de la table	Instruction SELECT	Fenêtre de Netcool/OMNIbus Administrator
Procédures	catalog.sql_procedures	`select ProcedureName from catalog.sql_procedures where CodeBlock like 'Country';`	Fenêtre SQL Procedure Details (Détails de la procédure SQL)
Filtres de restriction	catalog.restrictions	`select RestrictionName from catalog.restrictions where ConditionText like 'Country';`	Fenêtre Restriction Filter Details (Détails du filtre de restriction)
Vues	catalog.views	`select ViewName from catalog.views where CreationText like 'Country';`	

7. Recherchez les fichiers de règles de sonde et supprimez les références à la colonne qui s'y trouvent.
8. Recherchez vos fichiers de mappage de passerelle $NCHOME/omnibus/gates/objserv_*type*/objserv_*type*.map, où *type* représente uni ou bi. Supprimez toutes les références à la colonne.

9. Une fois que toutes les références à la colonne ont été supprimées, supprimez la colonne `Country` en utilisant la syntaxe `ALTER TABLE ... DROP COLUMN`.

Référence associée:

«ALTER SYSTEM», à la page 356
La commande `ALTER SYSTEM` permet de modifier les paramètres par défaut et actuels du serveur ObjectServer en paramétrant les propriétés, d'arrêter le serveur ObjectServer, de supprimer des connexions utilisateur ou de sauvegarder le serveur ObjectServer.

«DESCRIBE», à la page 382
La commande `DESCRIBE` permet d'afficher des informations sur les colonnes de la table ou de la vue spécifiée.

«SELECT (scalaire)», à la page 401
La commande scalaire `SELECT` extrait les colonnes et lignes d'une table en fonction de critères spécifiques.

«SELECT (aggregate)», à la page 403
Une commande `SELECT` d'agrégation calcule le nombre de lignes et renvoie une valeur unique.

«SELECT (GROUP BY)», à la page 405
Vous pouvez utiliser une commande `SELECT` avec une clause `GROUP BY` pour regrouper en une seule ligne toutes les lignes ayant des valeurs identiques dans une colonne ou une combinaison de colonnes spécifiques. Vous pouvez également rechercher la valeur agrégée de chaque groupe de valeurs de colonne.

«ALTER TABLE», à la page 357
La commande `ALTER TABLE` permet de modifier les caractéristiques d'une table existante et de ses colonnes. Vous pouvez ajouter, supprimer et modifier des colonnes ; vous pouvez également définir manuellement la valeur des zones de type `INCR`.

DROP DATABASE

La commande `DROP DATABASE` permet de supprimer une base de données existante.

Vous ne pouvez pas supprimer une base de données contenant des objets. De même, vous ne pouvez pas supprimer les bases de données de sécurité ou de catalogue, qui sont des bases de données initialisées par le système.

Syntaxe

```
DROP DATABASE nom_base_de_données;
```

Exemple

```
drop database mydb;
```

Concepts associés:

«Bases de données initialisées par le système», à la page 169
Lorsque vous initialisez un serveur ObjectServer, plusieurs bases de données par défaut sont créées.

DROP FILE

La commande DROP FILE permet de supprimer un fichier ObjectServer existant.

Vous ne pouvez pas supprimer un fichier en cours d'utilisation (dans un déclencheur, par exemple).

Syntaxe

```
DROP FILE nom_fichier;
```

L'opération supprime le fichier ObjectServer, mais pas les fichiers physiques créés dans le système de fichiers.

Exemple

```
drop file logit;
```

DROP GROUP

La commande DROP GROUP permet de supprimer le groupe spécifié.

Syntaxe

```
DROP GROUP 'nom_groupe';
```

La variable *nom_groupe* est une chaîne de texte contenant le nom unique du groupe en cours de suppression.

Remarque : Les groupes par défaut Normal, Administrator et Super User assure la sécurité du groupe au niveau de la ligne dans la liste des événements. Ces groupes ne peuvent pas être supprimés ou renommés. Avec les autres groupes par défaut, ils sont créés par le script security.sql.

Exemple

```
drop group 'LondonAdmin';
```

DROP INDEX

La commande DROP INDEX permet de supprimer un index redondant dans une table de base de données.

Syntaxe

```
DROP INDEX nom_index;
```

La valeur *nom_index* est le nom unique de l'index en cours de suppression.

Remarque : Si une colonne indexée est supprimée, l'index est automatiquement supprimé.

Exemple

```
drop index SeverityIdx;
```

DROP PROCEDURE

La commande DROP PROCEDURE permet de supprimer une procédure existante.

Vous ne pouvez pas supprimer une procédure référencée par d'autres objets (des déclencheurs, par exemple).

Syntaxe

```
DROP PROCEDURE nom_procédure;
```

Exemple

```
drop procedure testproc;
```

DROP RESTRICTION FILTER

La commande DROP RESTRICTION FILTER permet de supprimer un filtre de restriction existant.

Vous ne pouvez supprimer un filtre de restriction attribué à des utilisateurs ou des groupes.

Syntaxe

```
DROP RESTRICTION FILTER nom_filtre;
```

Exemple

```
drop restriction filter myfilter;
```

DROP ROLE

La commande DROP ROLE permet de supprimer un rôle existant.

Syntaxe

```
DROP ROLE 'nom_rôle';
```

La variable *nom_rôle* est une chaîne de texte contenant le nom unique du rôle en cours de suppression. Lorsque vous supprimez un rôle, tous les droits d'accès qui lui sont associés sont supprimés.

Exemple

```
drop role 'AutoAdmin';
```

DROP SIGNAL

La commande DROP SIGNAL permet de supprimer un signal défini par l'utilisateur.

Vous ne pouvez pas supprimer un signal référencé par un déclencheur.

Syntaxe

```
DROP SIGNAL nom_signal;
```

DROP TABLE

La commande DROP TABLE permet de supprimer une table existante.

Vous ne pouvez pas supprimer une table référencée par d'autres objets (des déclencheurs, par exemple) ou contenant des données. Vous ne pouvez pas non plus supprimer des tables système, qui contiennent des métadonnées relatives aux objets ObjectServer.

Syntaxe

```
DROP TABLE [nom_base_de_données.]nom_table;
```

Exemple

Pour supprimer toutes les lignes d'une table, procédez comme suit :

```
delete from mytab;
```

Pour supprimer la table, procédez comme suit :

```
drop table mytab;
```

Concepts associés:

«Tables système», à la page 171
Les tables système sont des tables spéciales gérées par le serveur ObjectServer et contiennent des métadonnées sur les objets du serveur ObjectServer.

DROP TRIGGER

La commande DROP TRIGGER permet de supprimer un déclencheur existant.

Syntaxe

```
DROP TRIGGER nom_déclencheur;
```

Vous ne pouvez pas supprimer un déclencheur s'il comporte des objets dépendants.

Exemple

```
drop trigger mytrig;
```

DROP TRIGGER GROUP

La commande DROP TRIGGER GROUP permet de supprimer un groupe de déclencheurs existant.

Syntaxe

```
DROP TRIGGER GROUP nom_groupe_déclencheurs;
```

Vous ne pouvez pas supprimer un groupe de déclencheurs contenant des déclencheurs.

Exemple

```
drop trigger group update_database_triggers;
```

DROP USER

La commande DROP USER permet de supprimer l'utilisateur spécifié.

Syntaxe

```
DROP USER 'nom_utilisateur';
```

La variable *nom_utilisateur* est une chaîne de texte contenant le nom unique de l'utilisateur en cours de suppression.

Exemple

```
drop user 'joe';
```

DROP VIEW

La commande DROP VIEW permet de supprimer une vue existante.

Vous ne pouvez pas supprimer une vue référencée par d'autres objets.

Syntaxe

```
DROP VIEW [nom_base_de_données.]nom_vue;
```

Si vous n'indiquez pas de nom de base de données, la vue est supprimée de la base de données des alertes.

Exemple

```
drop view myview;
```

EXECUTE PROCEDURE

Après avoir créé une procédure, vous devez l'exécuter à l'aide de la commande EXECUTE PROCEDURE pour réaliser les actions de la procédure. Pour ce faire, utilisez l'interface interactive SQL (**nco_sql**) ou un déclencheur ou une procédure.

Syntaxe

```
{ EXECUTE | CALL } [ PROCEDURE ] nom_procédure
 [ ( expression,... ) | ( [ expression, expression,... ] ,... ) ];
```

Utilisez la variable *nom_procédure* pour indiquer la procédure à exécuter.

Chacune des expressions transmises comme paramètres effectifs doit donner lieu à une variable assignable correspondant au type du paramètre spécifié lors de la création de la procédure.

Remarque : Si vous transmettez un paramètre de tableau, les crochets qui encadrent la liste d'expressions (en gras dans la syntaxe précédente) ne sont pas facultatifs.

Exemple

Pour exécuter la procédure décrite dans «CREATE PROCEDURE (procédures externes)», à la page 365, utilisez l'appel suivant dans un déclencheur :

```
execute send_email( critical.Node, critical.Severity, 'Netcool E-mail',
'root@localhost', critical.Summary, 'localhost');
```

Concepts associés:

«Composants d'une procédure SQL», à la page 193
Les procédures SQL possèdent les composants principaux suivants : paramètres, déclarations de variable locale et corps de procédure.

Référence associée:

«CREATE PROCEDURE (procédures externes)», à la page 365
La commande CREATE PROCEDURE permet de créer des procédures externes.

FOR

Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser la boucle FOR pour réaliser des actions un certain nombre de fois, en fonction de la variable de comptage.

Syntaxe

```
FOR comptage = 1 to entier DO
 BEGIN
   liste_commandes_action;
END;
```

Il existe une commande BREAK provenant de la boucle en cours, et l'exécution de l'instruction suivante de la procédure commence.

Une commande CANCEL permet d'arrêter l'exécution d'une procédure.

Avertissement : N'utilisez pas la commande CANCEL avec un bureau ObjectServer en mode DualWrite.

Exemple

La procédure suivante permet de mettre à jour les lignes de la table alerts.status table et d'attribuer la valeur TRUE à l'indicateur reconnu :

```
CREATE PROCEDURE ACKNOWLEDGE_TOOL( ids ARRAY OF CHAR(255) )
 DECLARE
   k INTEGER;
 BEGIN
   FOR k = 1 TO array_len( ids ) DO
   BEGIN
    UPDATE alerts.status VIA ( ids[k] ) SET Acknowledged = TRUE;
   END;
END;
```

Référence associée:

«Construction d'une instruction de corps de procédure SQL», à la page 193
Le corps d'une procédure SQL contient un ensemble d'instructions SQL et de constructions de programmation qui gèrent les données du serveur ObjectServer.

FOR EACH ROW

Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser la boucle FOR EACH ROW pour réaliser des actions sur un ensemble de lignes répondant à une certaine condition.

Syntaxe

```
FOR EACH ROW nom_variable in nom_base_de_données.nom_table
 [ WHERE condition ]
BEGIN
   liste_commandes_action;
END;
```

Dans cette instruction, le nom de la variable est déclaré de manière implicite en tant que référence de ligne. Par conséquent, il n'est pas utile de déclarer la variable au début de la procédure. Cela signifie que toutes les modifications apportées aux colonnes référencées par la variable affectent directement les lignes référencées dans le serveur ObjectServer. Lorsque `END` est atteint, la variable déclarée de manière implicite est supprimée et ne peut plus être utilisée ailleurs dans la procédure.

Seules les tables de base (pas les vues) peuvent être mises à jour dans la boucle FOR EACH ROW. Vous ne pouvez pas insérer d'éléments dans la table en cours de traitement dans la boucle FOR EACH ROW.

Si une erreur est détectée pendant que la boucle FOR EACH ROW analyse les lignes, l'analyse est arrêtée. Il existe une exception à ce comportement : lorsqu'une erreur est détectée à partir d'une commande ALTER SYSTEM DROP CONNECTION, c'est-à-dire si le client est déconnecté. Dans ce cas, la commande ALTER SYSTEM DROP CONNECTION consigne un message d'erreur et la commande FOR EACH ROW poursuit son analyse.

Si vous incluez une clause WHERE, seules les lignes satisfaisant aux critères spécifiés dans la condition sont renvoyées.

Il existe une commande BREAK provenant de la boucle en cours, et l'exécution de l'instruction suivante de la procédure commence.

Une commande CANCEL permet d'arrêter l'exécution d'une procédure.

Avertissement : N'utilisez pas la commande CANCEL avec un bureau ObjectServer en mode `DualWrite`.

Exemple

L'exemple ci-dessous augmente le niveau de gravité de toutes les alertes dans la table alerts.status de 3 à 4.

```
FOR EACH ROW alert_row in alerts.status WHERE alert_row.Severity=3
 BEGIN
   SET alert_row.Severity = 4;
END;
```

Lors de l'exécution de cette instruction, le serveur ObjectServer lit chaque ligne de la table alerts.status et détermine si la valeur de la colonne de gravité est 3. Pour chaque ligne satisfaisant à cette condition, les instructions comprises entre BEGIN et END sont exécutées tant que toutes les lignes n'ont pas été traitées.

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

Référence associée:

«Construction d'une instruction de corps de procédure SQL», à la page 193
Le corps d'une procédure SQL contient un ensemble d'instructions SQL et de constructions de programmation qui gèrent les données du serveur ObjectServer.

GRANT

La commande GRANT permet d'affecter des autorisations système et objet aux rôles. Les *autorisations système* contrôlent les commandes qui peuvent être exécutées dans le serveur ObjectServer. Les *autorisations objet* contrôlent l'accès aux objets, tels que des tables.

Syntaxe d'attribution des autorisations système

```
GRANT autorisation_système,...
 TO ROLE 'nom_rôle',...
 [ WITH GRANT OPTION ];
```

La valeur d'*autorisation_système* peut être l'une des sous-commandes suivantes :

```
ISQL
ISQL WRITE
ALTER SYSTEM DROP CONNECTION
ALTER SYSTEM SHUTDOWN
ALTER SYSTEM BACKUP
ALTER SYSTEM SET PROPERTY
CREATE DATABASE
CREATE FILE
CREATE RESTRICTION FILTER
CREATE SQL PROCEDURE
CREATE EXTERNAL PROCEDURE
CREATE SIGNAL
CREATE TRIGGER GROUP
CREATE USER
CREATE GROUP
CREATE ROLE
ALTER USER
ALTER GROUP
ALTER ROLE
DROP USER
DROP GROUP
DROP ROLE
GRANT ROLE
REVOKE ROLE
```

La variable *nom_rôle* est une chaîne de texte contenant le nom unique du rôle ou des rôles au(x)quel(s) vous attribuez des autorisations.

L'option WITH GRANT OPTION autorise les rôles auxquels l'autorisation est accordée d'attribuer l'autorisation à d'autres rôles.

Conseil : Vous pouvez interroger la table catalog.security_permissions afin de visualiser les informations sur les autorisations. Par exemple, pour visualiser chaque autorisation système, utilisez la commande SQL suivante : `SELECT * FROM catalog.security_permissions WHERE Object = 'SYSTEM' ORDER BY Permission;`

Exemple d'attribution d'autorisations système

```
grant create database to role 'DDL_Admin';
```

Syntaxe d'attribution des autorisations objet

```
GRANT autorisation_objet,... ON autorisation_objet nom_objet
 TO ROLE 'nom_rôle',...
 [ WITH GRANT OPTION ];
```

Vous pouvez attribuer une ou plusieurs autorisations aux objets du serveur ObjectServer. Utilisez *autorisation_objet* pour définir les commandes SQL que les

utilisateurs autorisés peuvent exécuter sur un objet du serveur ObjectServer de type *objet_autorisation*. La variable *nom_objet* est une chaîne de texte contenant le nom unique de l'objet.

Le propriétaire de l'objet (son créateur) possède automatiquement les autorisations d'attribution et de révocation associées à cet objet et peut attribuer et révoquer ces autorisations dans d'autres rôles. La table suivante répertorie les autorisations que le propriétaire possède pour chaque type d'objet. Le propriétaire peut également attribuer ces autorisations à d'autres utilisateurs.

Tableau 105. Objets et droits associés

Objets (*objet_droit***)**	**Droits (***droit_objet***)**
DATABASE	DROP CREATE TABLE CREATE VIEW
TABLE	DROP ALTER SELECT INSERT UPDATE DELETE CREATE INDEX DROP INDEX
VIEW	DROP ALTER SELECT UPDATE DELETE
TRIGGER GROUP	DROP ALTER CREATE TRIGGER
TRIGGER	DROP ALTER
FILE	DROP ALTER WRITE
SQL PROCEDURE EXTERNAL PROCEDURE	DROP ALTER EXECUTE

Tableau 105. Objets et droits associés (suite)

Objets (*objet_droit***)**	**Droits (***droit_objet***)**
SIGNAL	DROP ALTER RAISE
RESTRICTION FILTER	DROP ALTER

La variable *nom_rôle* est une chaîne de texte contenant le nom unique du rôle ou des rôles au(x)quel(s) les autorisations sont attribuées.

L'option WITH GRANT OPTION autorise les rôles auxquels l'autorisation est accordée d'attribuer l'autorisation à d'autres rôles.

Conseil : Dans les commandes où vous remplacez un objet existant à l'aide de la syntaxe CREATE OR REPLACE, vous devez disposer de l'autorisation ALTER pour remplacer un objet existant. Des objets peuvent être modifiés uniquement à l'aide de la syntaxe CREATE OR REPLACE ; par exemple, il n'existe pas de commande ALTER VIEW, mais vous pouvez remplacer une vue existante si vous disposez d'une autorisation ALTER dans la vue.

Exemple d'attribution d'autorisations objet

```
grant drop on database testdb to role 'DDL_Admin';
```

Référence associée:

«REVOKE», à la page 398
La commande `REVOKE` permet de révoquer les droits système et objet des rôles.

GRANT ROLE

Une fois que vous avez créé les rôles en tant que collectes d'autorisations, vous pouvez attribuer les rôles aux groupes et révoquer les rôles des groupes. Utilisez la commande `GRANT ROLE` pour affecter des rôles aux groupes.

Les attributions de rôle prennent effet dans la session client suivante.

Syntaxe

```
GRANT ROLE 'nom_rôle',...
 TO GROUP 'nom_groupe',...;
```

Chaque variable *nom_rôle* est une chaîne de texte contenant le nom unique d'un rôle affecté.

Chaque variable *nom_groupe* est le nom d'un groupe auquel le(s) rôle(s) est/sont attribué(s).

Exemple

```
grant role 'AutoAdmin' to group 'LondonAdministrators';
```

Référence associée:

«REVOKE ROLE», à la page 400
La commande `REVOKE ROLE` permet de révoquer les rôles des groupes.

IF THEN ELSE

Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser l'instruction `IF THEN ELSE` pour réaliser une ou plusieurs actions en fonction des conditions spécifiées.

Syntaxe

```
IF condition THEN liste_commandes_action
 [ ELSEIF condition THEN liste_commandes_action ]
  ...
 [ ELSE liste_commandes_action ]
END IF;
```

Si la première condition est satisfaite (TRUE), les commandes qui suivent le mot clé THEN son exécutées en séquence tant que ELSEIF, ELSE ou END IF n'est pas atteint. Si la première condition n'est pas satisfaite et qu'il existe une instruction ELSEIF pour laquelle une condition est remplie, les commandes qui suivent cette instruction ELSEIF sont exécutées tant que le mot clé suivant n'est pas atteint. S'il existe une instruction ELSE, et qu'aucune des conditions précédentes n'a été satisfaite, les instructions qui suivent ELSE sont exécutées tant que l'instruction END IF n'est pas atteinte.

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

Référence associée:

«Construction d'une instruction de corps de procédure SQL», à la page 193
Le corps d'une procédure SQL contient un ensemble d'instructions SQL et de constructions de programmation qui gèrent les données du serveur ObjectServer.

IDUC EVTFT

La commande `IDUC EVTFT` permet d'activer les notificateurs contextuels des événements accélérés à envoyer aux clients, puis d'activer la fonction de la liste d'événements de bureau ou de la liste d'événements actifs de l'interface graphique Web.

Syntaxe

```
IDUC EVTFT destination, type_action, ligne
```

Les variables de cette commande peuvent avoir les valeurs suivantes :

- *destination* = *spid* | *canal_iduc*
- *spid* = *expression_entier* (ID de connexion client du littéral)
- *canal_iduc* = *expression_chaîne* (nom du canal)
- *type_action* = INSERT | UPDATE | DELETE
- *ligne* = *variable* (référence du nom de la variable d'une ligne dans l'automatisation)

Par exemple, si vous avez défini un indicateur d'événements accélérés dans vos règles de sonde et que vous avez ajouté une colonne pour cet indicateur dans la table alerts.status, vous pouvez ajouter une condition dans un déclencheur de post-insertion pour examiner la valeur dans cette colonne. Si la valeur est satisfaite pour la notification d'événements accélérés, l'événement est ensuite transmis sous

la forme d'une notification contextuelle dans les clients de notification d'événements accélérés spécifiques. Vous pouvez définir la condition dans le déclencheur à l'aide du format suivant :

```
begin
 if ( new.nom_colonne_événement_accéléré = 1 )
  then
   iduc evtft 'nom_canal' , insert , new ;
 end if;
end;
```

Dans cette syntaxe, *nom_colonne_événements_accélérés* est le nom de la colonne qui contient l'indicateur d'événements accélérés dans la table alerts.status et *nom_canal* est le nom d'un canal sur lequel les données d'événement sont diffusées. Notez que le nom du canal est sensible à la casse afin de vous permettre d'utiliser la syntaxe correcte dans la syntaxe.

Exemple

```
create or replace trigger evtft_insert
group channel_triggers
priority 1
comment 'Fast track critical events from alerts.status'
after insert on alerts.status
for each row
begin
 if ( new.FastTrack = 1 )
  then
   iduc evtft 'FastTrack' , insert , new ;
 end if;
end;
```

IDUC FLUSH

La commande `IDUC FLUSH` permet d'envoyer des notifications IDUC aux clients IDUC.

Syntaxe

```
IDUC FLUSH destination
```

Dans cette commande :

- *destination = spid*
- *spid = expression_entier* (ID de connexion client littéral)

Exemple

```
create or replace trigger exmple_trigger
group default_triggers
enabled true
priority 2
on signal iduc_data_fetch
begin
 for each row conn in iduc_system.iduc_stats
 begin
  IDUC FLUSH conn.connectionid;
 end;
end;
go
```

IDUC SNDMSG

La commande IDUC SNDMSG permet d'envoyer des messages d'informations à un client Notification d'événement accéléré.

Syntaxe

```
IDUC SNDMSG destination, expression_chaîne
```

Les variables de cette commande peuvent prendre les valeurs suivantes :

- *destination* = *spid* | *canal_iduc*
- *spid* = *expression_entier* (ID de connexion client littéral)
- *canal_iduc* = *expression_chaîne* (Nom du canal)
- *expression_chaîne* = texte descriptif à envoyer comme message

Exemple

```
create trigger notify_isqlconn
group default_triggers
priority 1
on signal connect
begin
 if( %signal.process = 'isql' )
  then
   iduc sndmsg 'notif_isql', 'ISQL Connection from ' +
    %signal.node + ' from user ' +
    %signal.username + ' at ' +
    to_char(%signal.at)
 end if;
end;
```

INSERT

La commande INSERT permet d'insérer une nouvelle ligne de données dans une table existante.

Syntaxe

```
INSERT INTO [nom_base_de_données.]nom_table
 [ (nom_colonne,...) ] VALUES (expression,...);
```

Vous devez indiquer une valeur pour chaque colonne de clé primaire de la table.

Si vous insérez des valeurs pour chaque colonne de la ligne, indiquez le mot clé VALUES suivi d'une liste séparée par des virgules de valeurs de colonne entre parenthèses. Entrez les valeurs dans l'ordre séquentiel des colonnes.

Si vous n'insérez *pas* de valeurs pour chaque colonne de la ligne, indiquez une liste séparée par des virgules de colonnes insérées entre parenthèses, suivie du mot clé VALUES, suivi d'une liste séparée par des virgules de valeurs de colonne entre parenthèses. Entrez les valeurs dans la même séquence que les colonnes spécifiées. Toutes les autres colonnes sont remplies de valeurs par défaut.

Conseil : Vous ne pouvez pas attribuer de valeurs à des colonnes gérées par le système (Serial, par exemple).

Exemple

Pour insérer une alerte dans la table mydb.mystatus en spécifiant les valeurs des colonnes indiquées, entrez :

```
insert into mydb.mystatus (Identifier, Severity, LastOccurrence)
values ('MasterMachineStats15', 5, getdate);
```

RAISE SIGNAL

La commande RAISE SIGNAL permet d'émettre un signal défini par l'utilisateur.

Syntaxe

```
RAISE SIGNAL nom_signal expression,...;
```

Les expressions doivent donner une valeur compatible avec le type de données de l'attribut associé, tel que défini dans la commande CREATE SIGNAL.

Exemple

```
RAISE SIGNAL illegal_delete %user.user_name, old.Summary;
```

REVOKE

La commande REVOKE permet de révoquer les droits système et objet des rôles.

Syntaxe de révocation des droits système

```
REVOKE droit_système,...
 FROM ROLE 'nom_rôle',... ;
```

La liste suivante présente toutes les variables *droit_système* qu'il est possible de révoquer pour un rôle :

```
ISQL
ISQL WRITE
ALTER SYSTEM DROP CONNECTION
ALTER SYSTEM SHUTDOWN
ALTER SYSTEM BACKUP
ALTER SYSTEM SET PROPERTY
CREATE DATABASE
CREATE FILE
CREATE RESTRICTION FILTER
CREATE SQL PROCEDURE
CREATE EXTERNAL PROCEDURE
CREATE SIGNAL
CREATE TRIGGER GROUP
CREATE USER
CREATE GROUP
CREATE ROLE
ALTER USER
ALTER GROUP
ALTER ROLE
DROP USER
DROP GROUP
DROP ROLE
GRANT ROLE
REVOKE ROLE
```

Chaque variable *nom_rôle* est une chaîne de texte contenant le nom unique d'un rôle dont le droit est en cours de révocation.

Exemple de révocation de droits système

```
revoke create table from role 'DDL_Admin';
```

Syntaxe de révocation des droits objet

```
REVOKE droit_objet,...
ON droit_objet nom_objet
FROM ROLE 'nom_rôle',... ;
```

Vous pouvez révoquer un ou plusieurs droits des objets ObjectServer. Utilisez la variable *droit_objet* pour spécifier les commandes SQL que vous souhaitez révoquer pour un objet ObjectServer de type *droit_objet*. La valeur *nom_objet* est une chaîne de texte contenant le nom unique de l'objet.

Les détails de chaque objet et droits associés que vous pouvez révoquer sont présentés dans le tableau suivant.

Tableau 106. Objets et droits associés

Objets (*objet_droit***)**	**Droits (***droit_objet***)**
DATABASE	DROP CREATE TABLE CREATE VIEW
TABLE	DROP ALTER SELECT INSERT UPDATE DELETE CREATE INDEX DROP INDEX
VIEW	DROP ALTER SELECT UPDATE DELETE
TRIGGER GROUP	DROP ALTER CREATE TRIGGER
TRIGGER	DROP ALTER
FILE	DROP ALTER WRITE

Tableau 106. Objets et droits associés (suite)

Objets (*objet_droit***)**	**Droits (***droit_objet***)**
SQL PROCEDURE EXTERNAL PROCEDURE	DROP ALTER EXECUTE
SIGNAL	DROP ALTER RAISE
RESTRICTION FILTER	DROP ALTER

Chaque variable *nom_rôle* est une chaîne de texte contenant le nom unique d'un rôle dont le droit est en cours de révocation.

Remarque : La commande REVOKE ne tombe pas en cascade pas dans la hiérarchie des droits. Par exemple, si vous révoquez le droit CREATE TABLE du rôle SuperAdmin après que `SuperAdmin` l'a accordé au rôle LondonAdministrators, ce dernier conserve le droit CREATE TABLE.

Exemple de révocation de droits objet

```
revoke drop on database testdb from role 'DDL_Admin';
```

Référence associée:

«GRANT», à la page 392
La commande `GRANT` permet d'affecter des autorisations système et objet aux rôles. Les *autorisations système* contrôlent les commandes qui peuvent être exécutées dans le serveur ObjectServer. Les *autorisations objet* contrôlent l'accès aux objets, tels que des tables.

REVOKE ROLE

La commande `REVOKE ROLE` permet de révoquer les rôles des groupes.

Syntaxe

```
REVOKE ROLE 'nom_rôle',...
 FROM GROUP 'nom_groupe',... ;
```

Chaque variable *nom_rôle* est une chaîne de texte contenant le nom unique d'un rôle en cours de révocation.

Chaque variable *nom_groupe* est le nom d'un groupe auquel le ou les rôles ne sont plus affectés.

Remarque : La commande REVOKE ROLE ne tombe pas en cascade pas dans la hiérarchie des rôles. Par exemple, si vous révoquez le rôle AutoAdmin du groupe SuperAdmin après que SuperAdmin l'a accordé au groupe LondonAdministrators, ce dernier conserve le rôle AutoAdmin.

Exemple

```
revoke role 'AutoAdmin' from group 'LondonAdministrators';
```

Référence associée:

«GRANT ROLE», à la page 394
Une fois que vous avez créé les rôles en tant que collectes d'autorisations, vous pouvez attribuer les rôles aux groupes et révoquer les rôles des groupes. Utilisez la commande GRANT ROLE pour affecter des rôles aux groupes.

SELECT (scalaire)

La commande scalaire SELECT extrait les colonnes et lignes d'une table en fonction de critères spécifiques.

Syntaxe

```
SELECT [SKIP nombre_de_lignes_à_passer] [TOP nombre_lignes]
 {*|expr_colonne_scalaire [AS nom_alias],...}
 FROM [ nom_base_de_données.]nom_objet
 [ WHERE condition ]
 [ORDER BY nom_ou_alias_colonne [ASC|DESC],...];
```

Utilisez la clause facultative SKIP pour spécifier que les *nombre_de_lignes_à_passer* premières lignes sont exclues de l'ensemble de résultats. Vous pouvez utiliser la clause ORDER BY pour exclure les valeurs du résultat les plus ou les moins élevées. Si vous ne spécifiez pas de clause ORDER BY, les lignes sont exclues de façon non déterministe. Vous pouvez utiliser la clause SKIP et la clause TOP ensemble pour paginer les résultats. Si vous utilisez une clause SKIP avec une valeur qui est supérieure au nombre de lignes de l'ensemble de résultats, aucune ligne n'est retournée. La clause SKIP n'est pas autorisée dans les instructions select avec agrégation, dans les instructions select sur des vues et dans les instructions select contenant des clauses de sous-sélection ou une clause d'évaluation.

Utilisez la clause facultative TOP pour n'afficher que les premières *nombre_lignes* lignes des résultats de requête correspondant aux critères de sélection. Si vous incluez une clause TOP, vous devez également inclure une clause ORDER BY pour ordonner (ou trier) les lignes sélectionnées.

Utilisez un astérisque (*) pour extraire les colonnes non masquées dans la table. Sinon, vous pouvez soit spécifier la liste de colonnes que vous souhaitez extraire séparées par des virgules, soit créer des *colonnes virtuelles* à l'aide des éléments suivants :

- Expressions simples (par exemple, `Severity`)
- Expressions complexes contenant des opérateurs mathématiques ou de chaîne (par exemple, `Severity + Tally`)
- Fonctions (par exemple, `getdate - 60`)

A la suite d'une colonne ou d'une colonne virtuelle, vous pouvez inclure le mot clé AS suivi d'un alias. Cet alias est un en-tête de remplacement pour le nom de la colonne ou de la colonne virtuelle et s'affiche dans les résultats de la requête. Si vous spécifiez un alias de colonne, utilisez cet alias dans les références de la clause ORDER BY. La longueur maximale d'un nom de colonne ou d'alias est 40 caractères.

Si vous incluez une clause WHERE, seules les lignes qui satisfont aux critères spécifiés dans la *condition* sont retournées.

Utilisez la clause facultative ORDER BY pour afficher les résultats dans un ordre séquentiel décroissant (DESC) ou croissant (ASC) selon les valeurs d'un ou de plusieurs noms de colonnes. Si la clause ORDER BY n'est pas spécifiée, aucune organisation n'est effectuée. Si vous avez spécifié un alias de colonne à l'aide du mot clé AS, utilisez cet alias dans la liste de colonnes au lieu du nom de colonne correspondant.

Exemples

L'exemple suivant sélectionne toutes les lignes de la table alerts.status où Severity est égal à 4 :

```
select * from alerts.status where Severity = 4;
```

L'exemple suivant sélectionne toutes les lignes de la table alerts.status où le noeud contient la chaîne `terminal` suivie de tout autre caractère. Dans cet exemple, la syntaxe de l'expression standard est utilisée dans la comparaison LIKE. Pour plus d'informations sur la syntaxe d'expression standard utilisée dans la comparaison LIKE, voir l'annexe sur les expressions standard dans *IBM Tivoli Netcool/OMNIbus User's Guide*.

```
select * from alerts.status where Node like 'terminal.*';
```

Dans l'exemple suivant, la colonne virtuelle `Severity + Tally` est renseignée en additionnant les valeurs des deux colonnes :

```
select Severity, Severity + Tally from alerts.status;
```

L'exemple suivant est identique à l'exemple précédent, excepté que la colonne virtuelle `Severity + Tally` est renommée `Real_Severity` :

```
select Severity, Severity + Tally as Real_Severity from alerts.status;
```

Dans l'exemple suivant, la colonne Node est classée par ordre ascendant et les deux premières lignes sont exclues de l'ensemble de résultats :

```
select skip 2 Node from alerts.status order by Node asc;
```

Vous pouvez utiliser les clauses SKIP et TOP ensemble pour paginer les résultats. Dans l'exemple suivant, la colonne Node est classée par ordre ascendant, les deux premières lignes sont exclues de l'ensemble de résultats et les cinq premiers résultats sont sélectionnés :

```
select skip 2 top 5 Node from alerts.status order by Node asc;
```

Utilisation de variables ROW dans des clauses SUBSELECT

Une référence à une variable ROW dans une clause subselect est autorisée uniquement si la variable ROW est à un niveau d'imbrication différent.

Par exemple, l'instruction suivante n'est pas autorisée :

```
create or replace procedure test1()
begin
    for each row status_row in alerts.status where status_row.Identifier not in
            (select status_row.Identifier from alerts.test_tab )
    begin
        ....
    end;
end;
```

L'instruction suivante est un exemple de ce qui est autorisé :

```
create or replace procedure test1()
begin
    for each row t_group in catalog.trigger_groups
    begin
        for each row trig in catalog.trigger_stats where
            trig.TriggerName in ( select TriggerName from catalog.triggers
       where GroupName = t_group.GroupName )
        begin
            ...
        end;
    end;
end;
```

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

«Expressions», à la page 227
Une expression est une combinaison syntaxique de valeurs et d'opérations associées pour calculer de nouvelles valeurs. Les expressions peuvent être simples ou complexes.

«Fonctions», à la page 221
Une fonction traite un élément de données ou des éléments dans une commande SQL, puis retourne une valeur.

«Opérateurs mathématiques et de chaîne», à la page 215
Utilisez les opérateurs mathématiques pour ajouter, soustraire, multiplier et diviser les opérandes numériques dans les expressions. Utilisez les opérateurs de chaîne pour manipuler les chaînes de caractères (types de données VARCHAR et CHAR).

Référence associée:

«SELECT (GROUP BY)», à la page 405
Vous pouvez utiliser une commande SELECT avec une clause GROUP BY pour regrouper en une seule ligne toutes les lignes ayant des valeurs identiques dans une colonne ou une combinaison de colonnes spécifiques. Vous pouvez également rechercher la valeur agrégée de chaque groupe de valeurs de colonne.

«SELECT (aggregate)»
Une commande SELECT d'agrégation calcule le nombre de lignes et renvoie une valeur unique.

SELECT (aggregate)

Une commande SELECT d'agrégation calcule le nombre de lignes et renvoie une valeur unique.

Syntaxe

```
SELECT expression_agr [ AS nom_alias ],...
 FROM [nom_base_de_données.]nom_table
  [ WHERE condition ];
```

Les fonctions d'agrégation suivantes (illustrées par *expression_agr* dans la syntaxe) sont prises en charge.

Tableau 107. Fonctions d'agrégation

Fonction	Résultat retourné
max(*expr_colonne_scalaire*)	Retourne la valeur numérique maximale de l'expression de colonne à partir des lignes qui satisfont à la condition SELECT.
min(*expr_colonne_scalaire*)	Retourne la valeur numérique minimale de l'expression de colonne à partir des lignes qui satisfont à la condition SELECT.
avg(*expr_colonne_scalaire*)	Retourne la valeur numérique moyenne de l'expression de colonne à partir des lignes qui satisfont à la condition SELECT.
sum(*expr_colonne_scalaire*)	Retourne la somme (total) des valeurs numériques de l'expression de colonne à partir des lignes qui satisfont à la condition SELECT.
count(*expr_colonne_scalaire*) count(*)	Retourne le nombre total de lignes qui satisfont à la condition SELECT.
dist(*expr_colonne_scalaire*, *valeur*)	Retourne le nombre total de lignes dans lesquelles la colonne est égale à la valeur spécifiée. Le résultat de : dist(*expr_colonne_scalaire*, *valeur*) est équivalent à ce qui suit : SELECT count(*expr_colonne_scalaire*) FROM *nom_table* WHERE *expr_colonne_scalaire* = *valeur*;

En suivant une fonction d'agrégation, vous pouvez inclure le mot clé AS suivi d'un alias. Cet alias est un en-tête de remplacement pour l'expression d'agrégation et s'affiche dans les résultats de la requête.

La longueur maximale d'un nom de colonne ou d'alias est 40 caractères.

Si vous incluez une clause WHERE, seules les lignes qui satisfont aux critères spécifiés dans la *condition* sont retournées.

Exemples

L'exemple suivant retourne la valeur Severity la plus élevée, la valeur Severity moyenne et le nombre de lignes dans lesquelles la valeur Severity est égale à 4 :

```
select MAX(Severity), AVG(Severity), DIST(Severity, 4) from alerts.status;
```

L'exemple suivant retourne le nombre de lignes dans lesquelles la valeur de Node est myhost :

```
select DIST(Node, 'myhost') from alerts.status;
```

Les exemples suivants effectuent des comparaisons à l'aide de la fonction getdate, qui retourne l'heure actuelle :

```
select MAX(getdate-LastOccurrence) from alerts.status;

select AVG((getdate-LastOccurrence)/60) as ResponseTime from alerts.status
where OwnerUID=34;
```

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

«Fonctions», à la page 221
Une fonction traite un élément de données ou des éléments dans une commande SQL, puis retourne une valeur.

Référence associée:

«SELECT (GROUP BY)»
Vous pouvez utiliser une commande SELECT avec une clause GROUP BY pour regrouper en une seule ligne toutes les lignes ayant des valeurs identiques dans une colonne ou une combinaison de colonnes spécifiques. Vous pouvez également rechercher la valeur agrégée de chaque groupe de valeurs de colonne.

«SELECT (scalaire)», à la page 401
La commande scalaire SELECT extrait les colonnes et lignes d'une table en fonction de critères spécifiques.

SELECT (GROUP BY)

Vous pouvez utiliser une commande SELECT avec une clause GROUP BY pour regrouper en une seule ligne toutes les lignes ayant des valeurs identiques dans une colonne ou une combinaison de colonnes spécifiques. Vous pouvez également rechercher la valeur agrégée de chaque groupe de valeurs de colonne.

Syntaxe

```
SELECT [ TOP num_lignes ] expr_colonne_scalaire_et_expr_colonne_aggr [ AS nom_alias ] ,...
 FROM [nom_base_de_données.]nom_table
 [ WHERE condition ]
 GROUP BY expr_ou_alias_colonne_scalaire ,... [ HAVING condition ]
 [ ORDER BY expr_ou_alias_aggr [ { ASC | DESC } ] ,... ];
```

La syntaxe GROUP BY combine les expressions de colonne scalaire et les expressions d'agrégation. L'utilisation d'un astérisque (*) est admise uniquement dans la fonction d'agrégation COUNT(*).

A la suite d'une expression scalaire ou d'agrégation, vous pouvez inclure le mot clé AS suivi d'un alias. Cet alias est un en-tête de remplacement pour l'expression de colonne scalaire ou l'expression d'agrégation et s'affiche dans les résultats de la requête. Vous devez indiquer un alias pour chaque colonne virtuelle. Cela permet de le référencer dans la clause GROUP BY. Si vous n'indiquez aucun alias pour une expression d'agrégation, vous ne pouvez pas le référencer dans l'expression d'agrégation de la clause ORDER BY.

La longueur maximale d'un nom de colonne ou d'un alias est de 40 caractères.

La clause GROUP BY regroupe toutes les lignes contenant des données dans les colonnes spécifiées et permet d'exécuter des fonctions d'agrégation sur ces colonnes en fonction de leurs valeurs. Si vous avez indiqué un alias de colonne à l'aide du mot clé AS, utilisez cet alias dans la liste de la colonne GROUP BY plutôt que dans le nom de colonne ou l'expression correspondant.

Remarque : La liste de colonnes de la clause GROUP BY doit correspondre à celle sélectionnée et ne doit contenir aucune des expressions d'agrégation.

La variable *condition* qui suit le mot clé facultatif HAVING est une expression qui renvoie un sous-ensemble de lignes de la table. A l'inverse des autres conditions du serveur ObjectServer SQL, celles de la clause HAVING peuvent contenir des fonctions d'agrégation.

Utilisez la clause facultative ORDER BY pour afficher les résultats dans l'ordre séquentiel en fonction des valeurs d'un ou de plusieurs expressions d'agrégation, dans l'ordre croissant (ASC) ou décroissant (DESC). Si la clause ORDER BY n'est pas spécifiée, les résultats ne sont pas ordonnés. Vous devez utiliser l'alias de l'expression d'agrégation dans la clause ORDER BY plutôt que l'expression d'agrégation correspondante.

Exemples

L'exemple suivant renvoie la valeur de gravité la plus élevée pour chaque noeud :

```
select Node, max(Severity) from alerts.status group by Node;
```

L'exemple suivant renvoie la valeur de gravité la plus élevée trouvée pour chaque noeud, à l'exception du noeud Sun1, du niveau de gravité le plus bas au niveau le plus élevé :

```
select Node, max(Severity) as MAX_Sev from alerts.status
where Node <> 'Sun1' group by Node order by MAX_Sev;
```

L'alias de colonne de `max(Severity)` (`MAX_Sev`) s'affiche comme en-tête dans les résultats de la requête.

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

Référence associée:

«SELECT (scalaire)», à la page 401
La commande scalaire `SELECT` extrait les colonnes et lignes d'une table en fonction de critères spécifiques.

«SELECT (aggregate)», à la page 403
Une commande `SELECT` d'agrégation calcule le nombre de lignes et renvoie une valeur unique.

SET

Lors de la construction du corps d'une procédure SQL, vous pouvez utiliser une instruction d'affectation `SET` pour écrire la valeur d'une expression dans une variable ou un paramètre.

Syntaxe

```
SET { nom_paramètre | nom_variable } = expression
```

Vous pouvez attribuer une valeur à une référence de paramètre, de variable ou de ligne dans une boucle FOR EACH ROW.

La variable *expression* est une expression valide.

Remarque : Le type de la valeur renvoyée par l'expression doit être compatible avec la variable dans laquelle vous écrivez la valeur.

Concepts associés:

«Expressions», à la page 227
Une expression est une combinaison syntaxique de valeurs et d'opérations associées pour calculer de nouvelles valeurs. Les expressions peuvent être simples ou complexes.

Référence associée:

«Construction d'une instruction de corps de procédure SQL», à la page 193
Le corps d'une procédure SQL contient un ensemble d'instructions SQL et de constructions de programmation qui gèrent les données du serveur ObjectServer.

SET DATABASE et USE DATABASE

Les commandes SET DATABASE et USE DATABASE permettent de définir une base de données en tant que base de données par défaut pour une session d'interface interactive SQL **nco_sql**. Ces deux commandes réalisent la même fonction.

Restriction : Vous ne pouvez pas utiliser cette commande dans des déclencheurs ou des procédures.

Après avoir configuré la base de données par défaut avec la commande SET DATABASE ou USE DATABASE, vous pouvez spécifier un nom d'objet sans le faire précéder du nom de base de données. Les paramètres de la base de données par défaut durent tout le long de la session dans laquelle ils sont définis.

Syntaxe

```
{ SET | USE } DATABASE nom_base_données;
```

Remarque : La base de données par défaut n'est pas appliquée dans les commandes CREATE VIEW et DROP VIEW. Si aucun nom de base de données n'est spécifié dans ces commandes, la vue est toujours créée ou supprimée dans la base de données des alertes.

Exemples

```
use database newthings;

set database mydb;
```

SVC

La commande SVC permet d'ajouter ou de mettre à jour l'état d'une alerte de statut de service dans la table service.status pour IBM Tivoli Composite Application Manager for Internet Service Monitoring.

Syntaxe

```
SVC UPDATE 'nom' entier;
```

Dans cette commande, *nom* est le nom de l'élément de profil générant l'alerte et *entier* est son statut en cours. Les valeurs valides du statut du service sont présentées dans la table suivante. Si vous entrez une autre valeur, le niveau de service est défini sur 3 (inconnu).

Tableau 108. Niveaux de statut de service

Integer	Niveau de statut de service
0	Correct.
1	Marginal.

Tableau 108. Niveaux de statut de service (suite)

Integer	Niveau de statut de service
2	Incorrect.
3	Le niveau de service est inconnu.

Exemple

```
svc update 'newservice' 2;
```

UPDATE

La commande UPDATE permet de mettre à jour une ou plusieurs colonnes dans une ligne de données existante d'une table.

Syntaxe

```
UPDATE [nom_base_de_données.]nom_objet
 [ VIA valeur_de_colonne_clé_primaire,... ]
 SET nom_colonne = expression,...
 [ WHERE condition ];
```

Vous pouvez utiliser des valeurs de colonne dans les calculs. Dans l'exemple ci-dessous, la valeur 0 est attribuée à Severity lorsqu'une alerte a été reconnue :

```
update status set Severity=(1-Acknowledged)*Severity;
```

Vous ne pouvez pas mettre à jour des colonnes gérées par le système (Serial, par exemple) ni les colonnes dans lesquelles la valeur TRUE a été attribuée à la propriété NOMODIFY. Si la valeur TRUE a été attribuée à la propriété NOMODIFY, la valeur de la colonne ne peut pas être modifiée après la commande INSERT initiale.

Pour les autres colonnes ou lorsque la propriété NOMODIFY a pour valeur FALSE, vous pouvez mettre à jour les lignes d'une table à l'aide de la clause WHERE ou VIA.

Utilisation de la clause WHERE

Si vous incluez une clause WHERE, seules les lignes satisfaisant aux critères spécifiés dans la variable *condition* sont mises à jour. Si aucune condition n'est spécifiée dans la clause WHERE, toutes les lignes sont mises à jour.

La clause WHERE peut exécuter les même fonctions que la clause VIA dans les précédentes versions de Tivoli Netcool/OMNIbus. Si la clé primaire est spécifiée dans la clause WHERE, le serveur ObjectServer n'effectue pas d'analyse complète de la table mais accède directement aux lignes en utilisant la clé primaire. Les exemples suivants présentent des utilisations de la clause WHERE.

Pour définir la gravité (Severity) par 0 pour les lignes de la table alerts.status dans lesquelles Node est égal à Fred, entrez :

```
update alerts.status set Severity = 0 where Node = 'Fred';
```

Pour rechercher les lignes pour lesquelles Severity est égal à 1 et Node est égal à Fred, puis définir Severity par 0 et remplacer la zone Summary par la chaîne «Discarded», entrez :

```
update alerts.status set Severity = 0, Summary = 'Discarded'
where Severity = 1 and Node = 'Fred';
```

Utilisation de la clause VIA

La clause VIA est devenue obsolète avec la version 7.2 de Tivoli Netcool/OMNIbus mais elle reste valide. Ses fonctions peuvent à présent être exécutées par la clause WHERE.

Si vous connaissez la valeur de la clé primaire de la ligne que vous souhaitez mettre à jour, vous pouvez la spécifier à l'aide de la clause VIA. En présence de plusieurs colonnes de clé primaire, les valeurs doivent être spécifiées dans l'ordre et séparées par des virgules (,). Les valeurs de chaîne doivent être placées entre guillemets simples (').

Si plusieurs lignes sont mises à jour, la ou les clés primaires de chaque ligne doivent être placées entre crochets ([ ]) et séparées par des virgules (,).

La clause VIA peut être utile pour la documentation de votre code SQL. Par exemple, vous pouvez l'utiliser pour faire savoir au programme de maintenance des déclencheurs que la commande UPDATE est exécutée en utilisant des clés primaires.

L'exemple suivant indique comment mettre à jour plusieurs clés primaires à l'aide de la commande UPDATE VIA :

```
update alerts.status via ['722Identifier722'],['294Identifier294'] set Severity = 3;
```

Soit une table avec le schéma ci-après. L'exemple 1 montre comment mettre à jour une ligne dans une table dotée de plusieurs clés primaires. L'exemple 2 montre comment mettre à jour plusieurs lignes (trois dans ce cas) dans une table dotée de plusieurs clés primaires.

Exemple de schéma :

```
create table alerts.example persistent ( col1 int primary key, col2 int primary key,
col3 int primary key, col4 int );
```

Exemple 1 :

```
update alerts.example via 1,1,1 SET col4 = 10;
```

Exemple 2 :

```
update alerts.example via [1,1,1],[2,2,2],[3,3,3] SET col4 = 33;
```

Concepts associés:

«Conditions», à la page 228
Une condition est une combinaison d'expressions et d'opérateurs définis sur TRUE ou FALSE.

Référence associée:

«INSERT», à la page 397
La commande INSERT permet d'insérer une nouvelle ligne de données dans une table existante.

WRITE INTO

La commande WRITE INTO permet de consigner les informations de connexion dans les fichiers ObjectServer. Par exemple, vous pouvez utiliser cette commande pour écrire la sortie d'un déclencheur dans un fichier journal. Un fichier ObjectServer est un fichier logique, qui détient un fichier ou un ensemble de fichiers correspondant sur le système de fichiers physique.

Les fichiers possèdent une taille maximale définie. Lorsque cette taille est atteinte, un nouveau fichier est créé. Il y a un nombre maximal de fichiers. Lorsque la taille maximale et le nombre maximal de fichiers sont dépassés, les fichiers pivotent. Au cours de cette rotation de fichiers, n'ouvrez pas les fichiers journaux exigeant un accès en écriture exclusif (par exemple à l'aide de Microsoft Excel), car le serveur ObjectServer est incapable de faire pivoter les fichiers.

Syntaxe

```
WRITE INTO nom_fichier [ VALUES ] (expression, ...);
```

Un retour chariot suit chaque message.

Exemple

La commande suivante ajoute un message au fichier physique associé au fichier ObjectServer file1 à chaque connexion d'un utilisateur à une base de données.

```
WRITE INTO file1 VALUES
('User', %user.nom_utilisateur, 'connected to the system at', getdate );
```

Dans cet exemple, la variable utilisateur %user.nom_utilisateur est uniquement disponible dans les procédures et les déclencheurs.

Annexe C. Commandes SQL, expressions variables et boutons d'aide dans les outils, automatisations et listes d'événements transitoires

Vous pouvez utiliser un certain nombre de commandes SQL, d'expressions variables et de boutons d'aide pour récupérer des informations d'une liste d'événements en cours, de l'événement actuel ou de l'environnement du système d'exploitation. Vous pouvez utiliser ces expressions lors de la création d'un outil, d'un déclencheur ou d'une procédure SQL, ainsi que dans les paramètres transmis à une liste d'événements transitoires.

Le tableau suivant répertorie les commandes SQL, les expressions variables et les boutons d'aide.

Tableau 109. Commandes SQL, expressions variables et boutons d'aide dans les outils, déclencheurs, procédures et listes d'événements transitoires

Commande/ Expression variable	**Bouton**	**Syntaxe**
select_*commande* insert_*commande* update_*commande* delete_*commande* use_*commande* service_*commande*		Cliquez sur ce bouton pour sélectionner une commande SQL dans le menu contextuel. Selon la commande que vous avez sélectionnée, renseignez la fenêtre comme suit : • **Select** (SELECT) : sélectionnez la base de données et la table sur laquelle exécuter la commande SELECT. Ensuite, choisissez les colonnes de table à sélectionner. • **Insert** (INSERT) : sélectionnez la base de données et la table sur laquelle exécuter la commande INSERT. Ensuite, sélectionnez les colonnes de table dans lesquelles insérer les valeurs. Pour chaque colonne sélectionnée, entrez la valeur à insérer. Pour les instructions INSERT, vous devez inclure la clé primaire. Les clés primaires sont indiquées par un astérisque (*). • **Update** (UPDATE) : sélectionnez la base de données et la table sur lesquelles exécuter la commande. Ensuite, choisissez les colonnes de table à mettre à jour. Pour chaque colonne sélectionnée, entrez la nouvelle valeur. Pour les instructions UPDATE, vous devez exclure la clé primaire. Les clés primaires sont indiquées par un astérisque (*). **Remarque :** Pour les insertions et mises à jour de la table alerts.status, toutes les conversions existantes apparaissent dans les listes déroulantes. • **Delete** (Supprimer) : sélectionnez la table à supprimer. • **Use** (Utiliser) : sélectionnez la table à utiliser. • **Service** : sélectionnez un nom de service et une valeur. Les valeurs peuvent être `Good` (Correct), `Marginal` ou `Bad` (Incorrect).
nom_colonne *@nom_colonne*		Cliquez sur ce bouton pour sélectionner un nom de colonne de table à ajouter à la commande. Le nom de colonne est remplacé par la valeur de la ligne correspondante de la liste d'événements lors de l'exécution de l'outil. Lorsque le nom de colonne est précédé du symbole @, il est remplacé par la valeur de ligne correspondante de la liste d'événements lors de l'exécution. Cela peut être utilisé dans une requête SQL ou un fichier de restriction (`RemoteNodeAlias = '@LocalNodeAlias'`, par exemple)
nom_conversion		Cliquez sur ce bouton pour procéder à une sélection dans une liste de conversions disponibles.

© Copyright IBM Corp. 1994, 2014

Tableau 109. Commandes SQL, expressions variables et boutons d'aide dans les outils, déclencheurs, procédures et listes d'événements transitoires (suite)

Commande/ Expression variable	Bouton	Syntaxe
Non applicable		Cliquez sur ce bouton pour concevoir une liste de mots clés qui complète l'instruction SQL entrée.
Non applicable		Cliquez sur ce bouton pour vérifier la validité de la syntaxe SQL entrée.
%valeur_interne		Cliquez sur ce bouton pour procéder à une sélection dans une liste des valeurs internes connues de l'instance en cours de la liste d'événements. Par exemple, pour exécuter la liste d'événements transitoires et indiquer le serveur ObjectServer auquel se connecter à l'aide de l'option de ligne de commande `-server`, spécifiez : `-server "%server"` Les valeurs internes suivantes sont à la disposition des outils et se présentent sous la forme de paramètres dans la liste d'événements transitoires : **%display** : affichage en cours exécutant l'application (UNIX uniquement). **%password** : mot de passe de l'utilisateur qui exécute l'application. **%encrypted_password** : mot de passe codé de l'utilisateur qui exécute l'application (UNIX uniquement). En mode FIPS 140–2, le mot de passe est transmis sous la forme d'un texte en clair lorsqu'il est utilisé dans les outils, mais il est masqué lorsqu'il est spécifié sous forme de paramètre dans la ligne de commande. **%server** : nom du serveur ObjectServer auquel l'outil est connecté. **%desktopserver** : nom du serveur ObjectServer du bureau auquel l'outil est connecté. **%uid** : identifiant d'utilisateur ObjectServer de l'utilisateur qui exécute l'application. **%username** : nom d'utilisateur ObjectServer de l'utilisateur qui exécute l'application. La valeur interne suivante est disponible pour les procédures et déclencheurs : **%user** : permet de spécifier les variables utilisateur et les informations d'accès relatives aux utilisateurs connectés. La valeur interne suivante est disponible pour les déclencheurs uniquement : **%trigger** : permet de spécifier les variables de déclencheur et les informations d'accès relatives aux exécutions en cours et exécutions précédentes des déclencheurs. La valeur interne suivante est disponible pour les déclencheurs de signal uniquement : **%signal** : permet de spécifier les variables de signal et les informations d'accès relatives aux signaux émis.

Tableau 109. Commandes SQL, expressions variables et boutons d'aide dans les outils, déclencheurs, procédures et listes d'événements transitoires (suite)

Commande/ Expression variable	Bouton	Syntaxe
$prompt. *nom_invite*		Cliquez sur ce bouton pour sélectionner le nom de l'invite à utiliser lors de la requête de l'utilisateur. Par exemple, pour exécuter la liste d'événements transitoires et inviter l'utilisateur à entrer son mot de passe à l'aide de l'invite Mot de passe, indiquez : `-password $prompt.Password` Vous pouvez utiliser les invites dans les outils et sous forme de paramètres dans la liste des événements transitoires.
$selected_rows. *nom_colonne*	Non applicable	Liste de valeurs de *nom_colonne* pour toutes les alertes sélectionnées. Par exemple : `update alerts.status set TaskList = 0 where Serial in ($selected_rows.serial)` N'utilisez pas cette syntaxe si vous cochez la case **Exécuter pour chaque ligne sélectionnée**. Cochez plutôt la case si la modification est différente pour chaque alerte.
$(*variable_ environnement*)	Non applicable	Indique une variable d'environnement. Par exemple, lorsque vous exécutez une liste d'événements transitoires, vous pouvez indiquer le fichier de filtre à l'aide de l'option de ligne de commande `-elf` : `-elf "$(NCHOME)/omnibus/ini/tool.elf` Pour exécuter l'outil sous Windows, placez la variable d'environnement, comme $(NCHOME), entre guillemets. S'il existe un espace entre le nom de chemin, il ne sera pas interprété correctement.

Conseil :

- Lorsque vous entrez des commandes SQL dans les panneaux d'édition SQL de Tivoli Netcool/OMNIbus, vous pouvez saisir un ou plusieurs caractères et appuyer sur Ctrl+F1 pour ouvrir une boîte de dialogue contenant une liste de mots clés susceptibles de correspondre à votre entrée. Sélectionnez le mot clé requis, puis cliquez sur **OK** pour compléter votre entrée. Si un seul mot clé correspond aux caractères que vous avez entrés, il est automatiquement inséré. Si vous appuyez sur Ctrl+F1 après avoir entré un mot clé associé à une base de données, la boîte de dialogue propose une liste de bases de données ObjectServer possibles dans laquelle vous pouvez faire votre sélection. Si vous appuyez sur Ctrl+F1 après avoir entré un nom de base de données suivi d'un point (`alerts.`, par exemple), vous pouvez de nouveau appuyer sur Ctrl+F1 pour afficher une liste de tables dans la base de données et faire votre sélection.
- Vous pouvez cliquer sur le bouton **To Clipboard (Vers le presse-papiers)** pour copier la commande à un format texte vers le presse-papiers.

Tâches associées:

«Création et édition d'outils», à la page 108
Lors de sa création, un outil est ajouté à la base de données d'outils. Les outils qui s'affichent dans le panneau Tools (Outils) sont des liens vers des entrées de cette base de données.

«Création et édition de déclencheurs de base de données», à la page 117
Un déclencheur de base de données se met en application en cas de modification d'une base de données de déclenchement. Par exemple, vous pouvez créer un déclencheur pour réaliser une action à chaque insertion dans la table alerts.status.

«Création et édition de déclencheurs de signal», à la page 121
Les déclencheurs de signal sont activés lorsqu'un système ou un signal défini par l'utilisateur est mis au premier plan. Les signaux *système* sont émis de manière spontanée par le serveur ObjectServer lorsqu'il détecte des modifications apportées au système. Les signaux *définis par l'utilisateur* sont créés, émis et supprimés de manière explicite.

«Création et édition de déclencheurs temporels», à la page 124
Les déclencheurs temporels sont mis en application de manière répétée selon une fréquence spécifiée.

«Création et édition de procédures SQL», à la page 130
Les procédures SQL se composent essentiellement de paramètres, de déclarations de variable locale et du corps de la procédure.

Référence associée:

«Variables utilisateur implicites dans les procédures et les déclencheurs», à la page 195
Vous pouvez utiliser des variables utilisateur pour accéder aux informations relatives aux utilisateurs connectés dans une expression SQL dans le corps d'un déclencheur ou d'une procédure.

«Utilisation des variables de déclencheur dans les conditions et actions du déclencheur», à la page 178
Vous pouvez utiliser des variables de déclencheur pour accéder aux informations relatives aux exécutions en cours et précédentes du déclencheur. Utilisez la notation `%trigger` pour spécifier les variables de variables de déclencheur. Le symbole `%` indique que vous faites référence à une variable implicite. Le mot clé `trigger` fait référence au déclencheur en cours.

«Signaux système et leurs attributs», à la page 181
Si un signal système est émis, les attributs qui identifient la cause sont définis. Ces attributs sont transmis sous forme de variables implicites dans le déclencheur de signal associé.

Annexe D. Tables du serveur ObjectServer

La base de données du serveur ObjectServer contient les tables suivantes : les tables d'alertes, les tables de services, les tables de catalogues système, les tables de statistiques, les tables de prise en charge d'outils client, les tables d'outils de bureau, les tables du serveur ObjectServer de bureau, les tables de sécurité, les tables de canaux IDUC et les tables d'événements affectés par un service.

Les tables de la base de données du serveur ObjectServer sont stockées dans le fichier `$NCHOME/omnibus/db` sur les systèmes UNIX et dans le fichier `%NCHOME%\omnibus\db` sur les systèmes Windows.

Tables d'alertes

Les informations d'alertes sont réacheminées vers le serveur ObjectServer à partir de programmes externes, comme des sondes ou des passerelles. Ces informations sont stockées et gérées dans les tables de base de données et affichées dans la liste d'événements.

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 110. Colonnes de la table alerts.status

Nom de colonne	Type de données	Obligatoire	Description

Identifier	varchar(255)	Oui	Contrôle le dédoublonnage du serveur ObjectServer. La zone Identificateur contrôle la fonction de dédoublonnage 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 dédoublonnés avec leurs événements de problème respectifs.
			
			L'identificateur ci-dessous identifie correctement les événement répétés dans un environnement type :
			`@Identifier=@Node+" "+@AlertKey+" "+@AlertGroup+" "+@Type+" "+@Agent+" "+@Manager`
			
			Il est possible que l'ajout d'informations supplémentaires à la zone Identificateur soit nécessaire pour s'assurer que le dédoublonnage 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 :
			`@Identifier=@Node +" "+ @AlertKey+" "+@AlertGroup+" "+@Type+" "+@Agent+" "+@Manager+" "+$specific-trap+" "+$2`

Serial	incr	Oui	Numéro de série Tivoli Netcool/OMNIbus de la ligne.

© Copyright IBM Corp. 1994, 2014

Tableau 110. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
Node	varchar(64)	Oui	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. 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. 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.
NodeAlias	varchar(64)	Non	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. 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.
Manager	varchar(64)	Oui	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. Par exemple, la syntaxe suivante peut être utilisée pour définir la zone Manager : `@Manager="MTTrapd Probe on" + hostname()`
Agent	varchar(64)	Non	Nom descriptif du sous-gestionnaire qui a généré l'alerte. 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` Par exemple : `Cisco-Accounting Control, Cisco-Health Monitor, IETFBRIDGEMIB, ATMF-ATM-FORUM-MIB` 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. 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. 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`. 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.
AlertGroup	varchar(255)	Non	Nom descriptif du type de panne indiqué par l'alerte. Par exemple : `Interface Status` ou `CPU Utilization`). La zone AlertGroup doit contenir la même valeur pour les événements de problème connexe et de résolution. Par exemple, l'interruption SNMP 2 (linkDown) et l'interruption SNMP 3 (linkUp) doivent toutes deux contenir la même valeur AlertGroup de Link Status. La zone AlertGroup d'un message TL-1 sera définie sur la valeur du type d'alarme du message.

Tableau 110. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description

AlertKey	varchar(255)	Oui	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.
			
			Les sondes traitant les interruptions SNMP devraient définir la zone AlertKey sur l'une des valeurs suivantes (par ordre de préférence) :
			• 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 échouée. AlertKey peut être définie de deux manières :
			– `@AlertKey = extract($OID1, "\.([0-9]+)$")`
			– `@AlertKey = $1`
			• Description textuelle de l'instance dérivée du nom ou de la description de l'interruption. Par exemple, une unité ayant deux alimentations électriques (A et B) peut envoyer deux interruptions caractéristiques séparées, sans liaisons variables, pour indiquer le statut en échec de l'une de ses alimentations électriques. L'instance d'alimentation électrique appropriée aurait besoin d'être dérivée des définitions d'interruption de la base d'informations de gestion, puis encodée dans le fichier de règles :
			`switch($specific-trap)`
			`{`
			`case "1":`
			`@AlertKey = "A"`
			`case "2":`
			`@AlertKey = "B"`
			`default:`
			`}`
			• 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.
			
			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 :
			`Nov 20 13:12:57 device.customer.net`
			`195.180.208.193 19986: 37w0d: %LINK-3-UPDOWN:`
			`Interface FastEthernet0/13,`
			`changed state to down`
			
			Dans cet exemple, AlertKey serait défini sur `FastEthernet0/13` à l'aide de la syntaxe suivante :
			`@AlertKey = extract($Details, "Interface`
			`(.*),`
			`changed")`
			
			En règle générale, la zone AlertKey d'un message TL-1 est définie sur la valeur de l'emplacement de l'alarme du message.

Tableau 110. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
Severity	integer	Oui	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 : 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). 1 : Indéterminé. La 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. 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. 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. 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. 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.

Tableau 110. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description

Summary	varchar(255)	Oui	Contient du texte qui décrit la condition de l'alarme et l'instance de l'objet géré affectée.
			• 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`ifDescr` dans la seconde liaison variable. Le texte récapitulatif d'un tel événement serait similaire à :
			`"Link Down ( FastEthernet0/13 )"`
			• 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 :
			– Aucune valeur fournie :
			`"Link Utilization High ( BRI2/0:1 )"`
			– Nom de valeur comparée fourni :
			`"Link Utilization High: inOctets Exceeded Threshold ( BRI2/0:1)"`
			– Nom de valeur comparée et valeur fournies :
			`"Link Utilization High: inOctets, 7100, Exceeded Threshold (BRI2/0:1 )"`
			– Nom de seuil fourni :
			`"Link Utilization High: inOctetsMax Exceeded ( BRI2/0:1 )"`
			– Valeur de seuil fournie :
			`"Link Utilization High: inOctetsMax, 7000, Exceeded ( BRI2/0:1)"`
			– Valeur comparée et valeur de seuil fournies :
			`"Link Utilization High: 7100 Exceeded 7000 ( BRI2/0:1 )"`
			– Noms et valeurs fournis :
			`"Link Utilization High: inOctets, 7100, Exceeded inOctetsMax,7000 ( BRI2/0:1 )"`

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 110. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description

Type	integer	Non	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 :
			
			0 : Type non défini
			
			1 : Incident
			
			2 : Résolution
			
			3 : Incident Netcool/Visionary
			
			4 : Résolution Netcool/Visionary
			
			7 : Nouvelle alarme Netcool/ISMs
			
			8 : Ancienne alarme Netcool/ISMs
			
			11 : Plus grave
			
			12 : Moins grave
			
			13 : Information
			
			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.
			
			Par exemple, la logique de fichier de règles suivante est utilisée pour gérer des interruptions associées à l'état de connexion homologue BGP :
			`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`
			`}`

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 dédoublonnage.
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	Indique l'état d'escalade de l'alerte : 0 : Non escaladée 1 : Escaladée
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 110. 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 (LastOccurence) 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 110. 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 : • 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 110. 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 110. Colonnes de la table alerts.status (suite)

Nom de colonne	Type de données	Obligatoire	Description
OldRow	integer	Non	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. 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`. La valeur par défaut est `0`.
ProbeSub SecondId	integer	Non	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`.
MasterSerial	integer	Non	Identifie le serveur ObjectServer maître si cette alerte est traitée dans un environnement du serveur ObjectServer. Cette colonne est ajoutée lorsque vous exécutez l'utilitaire d'initialisation de la base de données **nco_dbinit** avec l'option `-desktopserver`. **Remarque :** MasterSerial doit être la dernière colonne de la table **alerts.status** si vous utilisez un environnement du serveur ObjectServer.
BSM_ Identity	varchar(1024)	Non	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).

Concepts associés:

«Fonctions», à la page 221
Une fonction traite un élément de données ou des éléments dans une commande SQL, puis retourne une valeur.

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 111. Colonnes de la table alerts.details

Nom de colonne	Type de données	Description
KeyField	varchar(255)	Chaîne d'organisation interne permettant une unicité. La valeur Keyfield est composée d'une valeur Identifier, de quatre chiffres et d'un numéro de séquence commençant par 1, par exemple : *Identifier####1* où *Identifier* est un type de données de varchar(255), utilisé pour relier les détails aux entrées de la table alerts.status. Si la valeur Identifier 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. **Conseil :** Pour éviter un débordement de KeyField (et assurer une unicité), la longueur de la valeur Identifier doit être suffisamment inférieure à 255 pour permettre l'ajout des quatre chiffres et d'un numéro de séquence (d'un ou plusieurs chiffres).

Tableau 111. Colonnes de la table alerts.details (suite)

Nom de colonne	Type de données	Description
Identifier	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 Identifier, 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 112. 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 alerts.iduc_messages

La table alerts.iduc_messages est obligatoire pour le support multiculturel et permet d'envoyer des messages au client insert, delete, update ou control (IDUC). Cette table garantit que les caractères transférés avec des encodages différents sont convertis en formats reconnaissables.

Le tableau suivant décrit les colonnes de la table alerts.iduc_messages.

Tableau 113. Colonnes de la table alerts.iduc_messages

Nom de colonne	Type de données	Description
MsgID	integer	Clé primaire de la table.
MsgText	varchar(4096)	Texte du message envoyé à une liste d'événements par l'utilitaire **nco_elct**. Cet utilitaire vous permet d'ouvrir une liste d'événements transitoire personnalisée.
MsgTime	time	Heure d'envoi du message.

Table alerts.application_types

La table alerts.application_types contient des détails sur les types d'application qui provoquent la génération de messages d'observation de la connexion lorsque les applications se connectent et se déconnectent. Cette table permet de configurer la gravité des événements de connexion et de déconnexion par type d'application.

Par exemple, une connexion de passerelle est traitée comme une résolution (effacement d'une déconnexion), alors qu'une connexion de liste d'événements est un événement de type 1, qui est résolu par une déconnexion. Une déconnexion de passerelle est traitée comme un incident alors qu'une déconnexion de liste d'événements est une résolution. La table alerts.application_types permet de configurer une passerelle pour générer un événement de type 1 (avertissement) lors de la déconnexion et un événement de type 2 (déconnexion effacée) lors de la connexion. Elle permet également de configurer une liste d'événements pour générer un événement de type 1 lors de la connexion et un événement de type 2 (effacer) lors de la déconnexion.

La table alerts.application_types est lue par les déclencheurs connection_watch_connect et connection_watch_disconnect.

Vous pouvez ajouter un nouveau type d'application à cette table en ajoutant une ligne, le cas échéant. Vous pouvez également modifier le comportement d'une application en mettant à jour sa ligne.

Le tableau suivant décrit les colonnes de la table alerts.application_types.

Tableau 114. Colonnes de la table alerts.application_types

Nom de colonne	Type de données	Description
application	varchar(64)	Clé primaire de la table. Il s'agit du nom d'application interne indiqué en tant qu'expression régulière pour la mise en correspondance efficace de chaînes.
description	varchar(64)	Nom descriptif de l'événement.
connect_type	int	Type d'événement de connexion.
connect_severity	int	Gravité de l'événement de connexion.

Tableau 114. Colonnes de la table alerts.application_types (suite)

Nom de colonne	Type de données	Description
disconnect_type	int	Type d'événement de déconnexion.
disconnect_severity	int	Gravité de l'événement de déconnexion.
expire_time	int	Nombre de secondes avant l'effacement automatique de l'alerte. Utilisé par l'automatisation Temps d'expiration.
discard	Boolean	Défini sur `TRUE` si l'événement doit être effacé.

Table master.class_membership

La table master.class_membership prend en charge le mappage des classes Tivoli Enterprise Console aux classes Tivoli Netcool/OMNIbus, et stocke les informations d'appartenance de classe. Cette table est utilisée avec la fonction SQL instance_of().

Le tableau suivant décrit les colonnes de la table master.class_membership.

Tableau 115. Colonnes de la table master.class_membership

Nom de colonne	Type de données	Description
Class	integer	Clé primaire de la table. Numéro de classe tel que stocké dans la table alert.status.
ClassName	varchar(255)	Nom de classe.
Parent	integer	Clé principale de la table. ID parent de la classe. La classe racine a un ID parent de -1. Si une classe a plusieurs parents, une ligne existe pour chaque parent.

La table master.class_membership n'autorise pas les mappages en double des noms de classe aux numéros de classe. Elle n'autorise également pas plusieurs entrées avec le même nom de classe ou le même numéro de classe. Si une entrée est insérée en double dans la table,un message d'avertissement au format suivant est ajouté dans le journal du serveur ObjectServer :

```
Warning: W-OBJ-103-002: Class name 'ClassName' and number 'class_number'
must be unique, row discarded
```

Concepts associés:

«Fonctions», à la page 221
Une fonction traite un élément de données ou des éléments dans une commande SQL, puis retourne une valeur.

Tables de services

La table de services contient des informations sur IBM Tivoli Composite Application Manager for Internet Service Monitoring.

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 116. 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.

Tables de catalogues système

La base de données de catalogue contient les tables système créées et gérées par le serveur ObjectServer. Les tables système contiennent des métadonnées sur les objets du serveur ObjectServer.

Vous pouvez afficher les informations des tables système à l'aide des commandes SELECT et DESCRIBE, mais vous ne pouvez pas modifier ces tables.

Table catalog.memstores

La table catalog.memstores stocke des informations sur les magasins de mémoire, y compris les limites matérielles et logicielles de taille du magasin, et le nombre d'octets actuellement utilisés.

Le tableau suivant décrit les colonnes de la table catalog.memstores.

Tableau 117. Colonnes de la table catalog.memstores

Nom de colonne	Type de données	Description
StoreName	varchar(40)	Nom du magasin de mémoire.
HardLimit	unsigned	Taille maximale du magasin en mémoire.
SoftLimit	unsigned	Taille maximale actuelle du magasin. Elle peut être augmentée jusqu'à la limite matérielle.
UsedBytes	unsigned	Quantité de mémoire (en octets) utilisée par le magasin de mémoire.
IsProtected	Boolean	Utilisation interne.

Table catalog.databases

La table catalog.databases stocke des informations sur chaque base de données, y compris le nombre d'objets dans la base de données et le type de base de données (système ou utilisateur).

Le tableau suivant décrit les colonnes de la table catalog.databases.

Tableau 118. Colonnes de la table catalog.databases

Nom de colonne	Type de données	Description
DatabaseName	varchar(40)	Nom de la base de données.
NumTables	unsigned	Nombre de tables de base et de vues dans la base de données.
IsSystem	Boolean	`TRUE` s'il s'agit d'une base de données système.

Table catalog.tables

La table catalog.tables stocke des informations sur tous les types de tables, y compris les tables système et utilisateur, les vues et les tables de transition.

Le tableau suivant décrit les colonnes de la table catalog.tables.

Tableau 119. Colonnes de la table catalog.tables

Nom de colonne	Type de données	Description
TableName	varchar(40)	Nom de la table.
DatabaseName	varchar(40)	Nom de la base de données parent.
Status	integer	Statut de la table : `0` : Valide `1` : Non valide `2` : Echec de la compilation
NumDependents	unsigned	Nombre d'unités dépendantes.
TableID	integer	Identificateur de la table.
TableKind	integer	Type de table : `0` : Table de base `1` : Table de transition `2` : Vue
StorageKind	integer	Type de stockage : `1` : Permanent `2` : Virtuel `4` : Transitoire

Table catalog.base_tables

La table catalog.base_tables stocke des informations sur les tables utilisateur et système.

Le tableau suivant décrit les colonnes de la table catalog.base_tables.

Tableau 120. Colonnes de la table catalog.base_tables

Nom de colonne	Type de données	Description
TableName	varchar(40)	Nom de la table.
DatabaseName	varchar(40)	Nom de la base de données parent.
StoreName	varchar(40)	Nom du magasin parent.
NumColumns	integer	Nombre de colonnes dans la table.
CreationTime	time	Heure de création de la table.
StorageKind	integer	Type de stockage : 1 : Permanent 2 : Virtuel
IsSystem	Boolean	TRUE s'il s'agit d'une table système.
IsNoModify	Boolean	TRUE s'il s'agit d'une table qui ne peut être actuellement modifiée.
UsedBytes	unsigned	Taille (en octets) de la table. Mise à jour lors de la création, de la modification et à un intervalle par défaut de 60 secondes.

Table catalog.views

La table catalog.views stocke des informations sur les vues. La colonne CreationText contient les commandes SQL utilisées pour créer la vue.

Le tableau suivant décrit les colonnes de la table catalog.views.

Tableau 121. Colonnes de la table catalog.views

Nom de colonne	Type de données	Description
ViewName	varchar(40)	Nom de la vue.
DatabaseName	varchar(40)	Nom de la base de données parent.
CreationText	varchar(16384)	Texte de la commande CREATE VIEW utilisée pour créer la vue.
StorageKind	integer	Type de stockage : 1 : Permanent 4 : Transitoire
IsRecovered	Boolean	TRUE s'il s'agit d'une vue récupérée avec succès après le redémarrage.
IsDmlEnabled	Boolean	TRUE si toutes les clés primaires de la table se trouvent dans la vue et si des actions DML peuvent par conséquent être exécutées dans la vue.
IsAggregate	Boolean	TRUE si cette colonne est créée à partir d'une instruction SELECT agrégée.

Table catalog.files

La table catalog.files stocke des informations sur les fichiers du serveur ObjectServer, y compris le chemin d'accès au fichier du système d'exploitation associé à chaque fichier du serveur ObjectServer.

Le tableau suivant décrit les colonnes de la table catalog.files.

Tableau 122. Colonnes de la table catalog.files

Nom de colonne	Type de données	Description
FileName	varchar(40)	Nom du fichier du serveur ObjectServer.
FilePath	varchar(1028)	Chemin d'accès complet au fichier sur le système de fichiers.
MaximumFiles	unsigned	Nombre maximum de fichiers.
MaximumSize	unsigned	Taille maximale des fichiers.
IsEnabled	Boolean	`TRUE` si les informations ont été consignées dans ce fichier.
Status	integer	Etat actuel du fichier : `0` : Valide `1` : Non valide `2` : Echec de la compilation

Table catalog.restrictions

La table catalog.restrictions stocke des informations sur les filtres de restriction. La colonne ConditionText contient la condition SQL.

Le tableau suivant décrit les colonnes de la table catalog.restrictions.

Tableau 123. Colonnes de la table catalog.restrictions

Nom de colonne	Type de données	Description
RestrictionName	varchar(40)	Nom du filtre de restriction.
TableName	varchar(40)	Nom de la table dans laquelle le filtre de restriction a été créé.
DatabaseName	varchar(40)	Nom de la base de données parent.
ConditionText	varchar(16384)	Texte de condition du filtre de restriction.
CreationText	varchar(16384)	Texte de CREATE RESTRICTION utiliser pour créer le filtre de restriction.

Table catalog.columns

La table catalog.columns stocke des informations sur les colonnes de table, y compris leurs types de données.

Le tableau suivant décrit les colonnes de la table catalog.columns.

Tableau 124. Colonnes de la table catalog.columns

Nom de colonne	Type de données	Description
ColumnName	varchar(40)	Nom de la colonne.
TableName	varchar(40))	Nom de la table.
DatabaseName	varchar(40)	Nom de la base de données parent.
DataType	entier	Type de colonne de données.

Tableau 124. Colonnes de la table catalog.columns (suite)

Nom de colonne	Type de données	Description
Length	unsigned	Nombre de caractères dans la colonne.
IsPrimaryKey	Boolean	TRUE si la colonne est une clé primaire.
OrdinalPosition	unsigned	Position dans la liste de colonnes.
IsHidden	Boolean	TRUE s'il s'agit d'une colonne masquée.
IsNoModify	Boolean	TRUE si la colonne ne peut pas être actuellement modifiée.
IsNoDefault	Boolean	TRUE si la valeur de cette colonne doit être indiquée dans la commande INSERT initiale.
IsSystem	Boolean	TRUE s'il s'agit d'une colonne système.

Table catalog.primitive_signals

La table catalog.primitive_signals stocke des informations sur les signaux utilisateur et système.

Le tableau suivant décrit les colonnes de la table catalog.primitive_signals.

Tableau 125. Colonnes de la table catalog.primitive_signals

Nom de colonne	Type de données	Description
SignalName	varchar(40)	Nom du signal.
IsSystem	Boolean	TRUE s'il s'agit d'un signal système.
CommentBlock	varchar(1024)	Chaîne de commentaire indiquée dans la commande CREATE SIGNAL.

Table catalog.primitive_signal_parameters

La table catalog.primitive_signal_parameters stocke des informations sur les paramètres des signaux système ou définis par l'utilisateur.

Le tableau suivant décrit les colonnes de la table catalog.primitive_signal_parameters.

Tableau 126. Colonnes de la table catalog.primitive_signal_parameters

Nom de colonne	Type de données	Description
ParameterName	varchar(40)	Nom du paramètre.
SignalName	varchar(40)	Nom du signal avec ce paramètre.
DataType	integer	Type de données de paramètres.
Length	unsigned	Nombre de caractères dans le paramètre.
OrdinalPosition	integer	Position dans la liste de paramètres.

Table catalog.trigger_groups

La table catalog.trigger_groups stocke des informations sur les groupes de déclencheurs, y compris si le groupe de déclencheurs est activé.

Le tableau suivant décrit les colonnes de la table catalog.trigger_groups.

Tableau 127. Colonnes de la table catalog.trigger_groups

Nom de colonne	Type de données	Description
GroupName	varchar(40)	Nom du groupe de déclencheurs.
IsEnabled	Boolean	TRUE si le groupe de déclencheurs est actuellement activé.

Table catalog.triggers

La table catalog.triggers stocke des informations sur les déclencheurs, y compris le type de déclencheur, sa priorité et le groupe dans lequel il se trouve.

Le tableau suivant décrit les colonnes de la table catalog.triggers.

Tableau 128. Colonnes de la table catalog.triggers

Nom de colonne	Type de données	Description
TriggerName	varchar(40)	Nom du déclencheur.
GroupName	varchar(40)	Nom du groupe de déclencheurs.
TriggerKind	integer	Type de déclencheur : 0 : Base de données 1 : Signal 2 : Temporel
DebugEnabled	Boolean	TRUE si le débogage est activé pour le déclencheur.
IsEnabled	Boolean	TRUE si le déclencheur est activé.
TriggerPriority	integer	Priorité du déclencheur : 1 est le niveau le plus élevé, 20 est le niveau le plus faible de priorité.
CommentBlock	varchar(1024)	Chaîne de commentaire indiquée dans la commande CREATE TRIGGER.
EvaluateBlock	varchar(2048)	Clause d'évaluation indiquée dans la commande CREATE TRIGGER.
BindName	varchar(40)	Nom de liaison indiqué dans la clause d'évaluation de la commande CREATE TRIGGER.
ConditionBlock	varchar(1024)	Condition WHEN indiquée dans la commande CREATE TRIGGER.
DeclareBlock	varchar(1024)	Déclaration de variable indiquée dans la commande CREATE TRIGGER.
CodeBlock	varchar(8192)	Corps du déclencheur.

Table catalog.database_triggers

La table catalog.database_triggers stocke des informations sur les déclencheurs de base de données, y compris le type d'opération de base de données qui provoque la mise en application du déclencheur.

Le tableau suivant décrit les colonnes de la table catalog.database_triggers.

Tableau 129. Colonnes de la table catalog.database_triggers

Nom de colonne	Type de données	Description
TriggerName	varchar(40)	Nom du déclencheur.
EventOrder	integer	Ordre des événements : 0 : Avant 1 : Après
EventOp	integer	Opération d'événement : 0 : Insérer 1 : Réinsérer 2 : Mettre à jour 3 : Supprimer
EventLevel	integer	Niveau de déclenchement : 0 : Déclencheur au niveau de la ligne 1 : Déclencheur au niveau de l'instruction
DatabaseName	varchar(40)	Nom de la base de données.
TableName	varchar(40)	Nom de la table.

Table catalog.signal_triggers

La table catalog.signal_triggers stocke des informations sur les déclencheurs de signaux, y compris le nom du signal qui provoque la mise en application du déclencheur.

Le tableau suivant décrit les colonnes de la table catalog.signal_triggers.

Tableau 130. Colonnes de la table catalog.signal_triggers

Nom de colonne	Type de données	Description
TriggerName	varchar(40)	Nom du déclencheur.
SignalName	varchar(40)	Nom du signal.

Table catalog.temporal_triggers

La table catalog.temporal_triggers stocke des informations sur les déclencheurs temporels, y compris la fréquence de mise en application.

Le tableau suivant décrit les colonnes de la table catalog.temporal_triggers.

Tableau 131. Colonnes de la table catalog.temporal_triggers

Nom de colonne	Type de données	Description
TriggerName	varchar(40)	Nom du déclencheur.
Frequency	integer	Fréquence du déclencheur en secondes.

Table catalog.procedures

La table catalog.procedures stocke des informations sur les procédures, y compris si la procédure est une procédure SQL ou une procédure externe.

Le tableau suivant décrit les colonnes de la table catalog.procedures.

Tableau 132. Colonnes de la table catalog.procedures

Nom de colonne	Type de données	Description
ProcedureName	varchar(40)	Nom de la procédure.
Kind	unsigned	Type de procédure : 0 : SQL 1 : Externe

Table catalog.sql_procedures

La table catalog.sql_procedures stocke des informations sur les procédures SQL, y compris le code de la procédure.

Le tableau suivant décrit les colonnes de la table catalog.sql_procedures.

Tableau 133. Colonnes de la table catalog.sql_procedures

Nom de colonne	Type de données	Description
ProcedureName	varchar(40)	Nom de la procédure.
DeclareBlock	varchar(16384)	Déclaration de variable indiquée dans la commande CREATE PROCEDURE.
CodeBlock	varchar(32768)	Corps de la procédure.

Table catalog.external_procedures

La table catalog.external_procedures stocke des informations sur les procédures externes, y compris le nom de l'exécutable de la procédure et l'hôte sur lequel il s'exécute.

Le tableau suivant décrit les colonnes de la table catalog.external_procedures.

Tableau 134. Colonnes de la table catalog.external_procedures

Nom de colonne	Type de données	Description
ProcedureName	varchar(40)	Nom de la procédure.

Tableau 134. Colonnes de la table catalog.external_procedures (suite)

Nom de colonne	Type de données	Description
ExecutableName	varchar(1024)	Nom de l'exécutable.
HostName	varchar(1024)	Nom de l'hôte.
UserId	varchar(1024)	ID utilisateur.
GroupId	varchar(1024)	ID groupe.
ArgumentsSpec	varchar(32768)	Arguments indiqués dans le texte CREATE PROCEDURE.

Table catalog.procedure_parameters

La table catalog.procedure_parameters stocke des informations sur les paramètres de procédure, y compris les types de paramètres.

Le tableau suivant décrit les colonnes de la table catalog.procedure_parameters.

Tableau 135. Colonnes de la table catalog.procedure_parameters

Nom de colonne	Type de données	Description
ParameterName	varchar(40)	Nom du paramètre.
ProcedureName	varchar(40)	Nom de la procédure.
ParameterKind	integer	Type de paramètre : `0` : Base `1` : Ligne `2` : Tableau
DataType	integer	Type de données du paramètre.
OrdinalPosition	integer	Position dans la liste d'arguments.
Length	integer	Nombre de caractères dans le paramètre.
TableName	varchar(40)	Pour un paramètre de ligne, il s'agit de la table parent de cette ligne. Sinon, il s'agit d'une chaîne vide.
DatabaseName	varchar(40)	Pour un paramètre de ligne, il s'agit de la base de données parent de la table parent de cette ligne. Sinon, il s'agit d'une chaîne vide.
ParameterMode	integer	Mode du paramètre : `1` : Entrée `2` : Sortie `3` : Entrée/Sortie

Table catalog.connections

La table catalog.connections contient des informations sur les connexions au serveur ObjectServer.

Le tableau suivant décrit les colonnes de la table catalog.connections.

Tableau 136. Colonnes de la table catalog.connections

Nom de colonne	Type de données	Description
ConnectionID	integer	ID connexion.
LogName	varchar(40)	Nom du fichier journal de l'application connectée.
HostName	varchar(40)	Nom de l'hôte connecté.
AppName	varchar(40)	Nom de l'application connectée.
AppDescription	varchar(40)	Description de l'application connectée.
IsRealTime	Boolean	`TRUE` si le client utilise IDUC. Les bureaux et passerelles utilisent IDUC et sont des connexions en temps réel.
ConnectTime	time	Durée de connexion du client.

Table catalog.properties

La table catalog.properties contient des informations sur les propriétés du serveur ObjectServer.

Le tableau suivant décrit les colonnes de la table catalog.properties.

Tableau 137. Colonnes de la table catalog.properties

Nom de colonne	Type de données	Description
PropName	varchar(40)	Nom de la propriété.
PropGroup	varchar(40)	Groupe de la propriété, tel que **`Auto`** ou **`Store`**. Toutes les propriétés n'appartiennent pas à un groupe.
Description	varchar(255)	Description de la propriété.
Type	integer	Type de données de la propriété.
Valeur	varchar(255)	Valeur actuelle de la propriété.
IsModifyable	Boolean	`TRUE` si la propriété est modifiable.
IsImmediate	Boolean	`TRUE` si lors de modifications de la propriété, l'effet est immédiat. Sinon, le serveur ObjectServer doit être redémarré.
IsAdvanced	Boolean	`TRUE` si la propriété est avancée. Les propriétés avancées ne doivent pas être modifiées sans l'assistance du service de support logiciel IBM.

Table catalog.security_permissions

La table catalog.security_permissions contient des informations de droit relatives aux objets du serveur ObjectServer. Cette table est uniquement utilisée par Netcool/OMNIbus Administrator.

Le tableau suivant décrit les colonnes de la table catalog.security_permissions.

Tableau 138. Colonnes de la table catalog.security_permissions

Nom de colonne	Type de données	Description
ApplicationID	integer	Identificateur d'application.
Object	varchar(40)	Nom de l'objet.
ObjectType	integer	Type d'objet, par exemple, une table.
ActionID	integer64	Identificateur de l'action de droit. Cette colonne est uniquement utilisée par Netcool/OMNIbus Administrator.
Permission	varchar(40)	Type de droit.

Table catalog.profiles

La table catalog.profiles contient des informations de temporisation pour les commandes SQL en cours d'exécution à partir des connexions client.

Les statistiques du profil SQL sont également consignées dans le fichier `$NCHOME/omnibus/log/`*`nom de serveur`*`_profiler_report.log`*`n`*, à l'intervalle indiqué dans la propriété **`ProfileStatsInterval`** ou dans l'option de ligne de commande `-profilestatsinterval`.

Le tableau suivant décrit les colonnes de la table catalog.profiles.

Tableau 139. Colonnes de la table catalog.profiles

Nom de colonne	Type de données	Description
ConnectionID	integer	ID connexion.
UID	integer	ID utilisateur.
AppName	varchar(40)	Nom de l'application connectée.
HostName	varchar(40)	Nom de l'hôte connecté.
ProfiledFrom	time	Heure de début du profilage.
LastSQLTime	real	Durée, en secondes, de la dernière commande SQL.
MinSQLTime	real	Temps d'exécution le plus court, en secondes, pour ce client.
MaxSQLTime	real	Temps d'exécution le plus long, en secondes, pour ce client.
PeriodSQLTime	real	Durée, en secondes, pendant laquelle l'application a exécuté des instructions SQL depuis le dernier rapport de profil.
TotalSQLTime	real	Durée totale, en secondes, pour l'exécution de toutes les commandes SQL pour ce client.
LastTimingAt	time	Heure de dernière utilisation d'un profil SQL pour ce client.
NumSubmits	integer	Nombre de soumissions pour ce client. Une seule soumission peut contenir plusieurs commandes SQL, exécutées avec la commande **`go`**.
TotalParseTime	real	Enregistre la durée totale passée à exécuter une analyse syntaxique des commandes pour ce client.

Tableau 139. Colonnes de la table catalog.profiles (suite)

Nom de colonne	Type de données	Description
TotalResolveTime	real	Enregistre la durée totale passée à résoudre des commandes pour ce client.
TotalExecTime	real	Enregistre la durée totale passée à exécuter des commandes pour ce client.

Table catalog.indexes

La table catalog.indexes stocke des informations sur les index, y compris la colonne et la table de base de données sur lesquelles l'index se base, et le type d'index.

Le tableau suivant décrit les colonnes de la table catalog.indexes.

Tableau 140. Colonnes de la table catalog.indexes

Nom de colonne	Type de données	Description
IndexName	varchar(40)	Nom de l'index.
DatabaseName	varchar(40)	Nom de la base de données dans laquelle se trouve la colonne indexée.
TableName	varchar(40)	Nom de la table dans laquelle se trouve la colonne indexée.
ColumnName	varchar(40)	Nom de la colonne indexée.
IndexKind	integer	Type d'index : 1 : Hachage 2 : Arborescence
IsValid	Boolean	TRUE si l'index est valide. FALSE si l'index est non valide, en raison d'un incident d'allocation de mémoire. Pour rendre un index à nouveau valide, vous pouvez régénérer l'index en redémarrant le serveur ObjectServer.

Tables de statistiques

Les tables de statistiques contiennent des informations de temporisation.

La table catalog.profiles contient des informations de temporisation pour l'exécution de commandes SQL à partir de connexions client.

La table master.stats stocke des informations de temporisation sur les tables alerts.status, alerts.details et alerts.journal.

La table catalog.trigger_stats stocke des informations de temporisation sur les déclencheurs.

Table catalog.profiles

La table catalog.profiles contient des informations de temporisation pour les commandes SQL en cours d'exécution à partir des connexions client. Le profilage SQL est activé à l'aide de la propriété **Profile** ou de l'option de ligne de commande `-profile`.

Les statistiques du profil SQL sont également consignées dans le fichier `$NCHOME/omnibus/log/`*`nom de serveur`*`_profiler_report.log`*`n`*, à l'intervalle indiqué dans la propriété **ProfileStatsInterval** ou dans l'option de ligne de commande `-profilestatsinterval`.

Le tableau suivant décrit les colonnes de la table catalog.profiles.

Tableau 141. Colonnes de la table catalog.profiles

Nom de colonne	Type de données	Description
ConnectionID	integer	ID connexion.
UID	integer	ID utilisateur.
AppName	varchar(40)	Nom de l'application connectée.
HostName	varchar(40)	Nom de l'hôte connecté.
ProfiledFrom	time	Heure de début du profilage.
LastSQLTime	real	Durée, en secondes, de la dernière commande SQL.
MinSQLTime	real	Temps d'exécution le plus court, en secondes, pour ce client.
MaxSQLTime	real	Temps d'exécution le plus long, en secondes, pour ce client.
PeriodSQLTime	real	Durée, en secondes, pendant laquelle l'application a exécuté des instructions SQL depuis le dernier rapport de profil.
TotalSQLTime	real	Durée totale, en secondes, pour l'exécution de toutes les commandes SQL pour ce client.
LastTimingAt	time	Heure de dernière utilisation d'un profil SQL pour ce client.
NumSubmits	integer	Nombre de soumissions pour ce client. Une seule soumission peut contenir plusieurs commandes SQL, exécutées avec la commande **go**.
TotalParseTime	real	Enregistre la durée totale passée à exécuter une analyse syntaxique des commandes pour ce client.
TotalResolveTime	real	Enregistre la durée totale passée à résoudre des commandes pour ce client.
TotalExecTime	real	Enregistre la durée totale passée à exécuter des commandes pour ce client.

Table master.stats

La table master.stats stocke des informations de temporisation sur les tables alerts.status, alerts.details et alerts.journal. Ces informations de temporisation sont recueillies si le groupe de déclencheurs stats_triggers est activé. Le groupe de déclencheurs stats_triggers est désactivé par défaut dans le fichier `automation.sql`.

Le tableau suivant décrit les colonnes de la table master.stats.

Tableau 142. Colonnes de la table master.stats

Nom de colonne	Type de données	Description
StatTime	time	Heure à laquelle les statistiques sont collectées.

Tableau 142. Colonnes de la table master.stats (suite)

Nom de colonne	Type de données	Description
NumClients	integer	Nombre total de clients (par exemple, les bureaux) connectés au serveur ObjectServer.
NumRealtime	integer	Nombre de clients en temps réel connectés au serveur ObjectServer. Les bureaux et passerelles utilisent IDUC et sont des connexions en temps réel.
NumProbes	integer	Nombre de sondes connectées au serveur ObjectServer.
NumGateways	integer	Nombre de passerelles connectées au serveur ObjectServer.
NumMonitors	integer	Nombre de moniteurs connectés au serveur ObjectServer.
NumProxys	integer	Nombre de serveurs proxy connectés au serveur ObjectServer.
EventCount	integer	Nombre actuel d'entrées dans la table alerts.status.
JournalCount	integer	Nombre actuel d'entrées dans la table alerts.journal.
DetailCount	integer	Nombre actuel d'entrées dans la table alerts.details.
StatusInserts	integer	Nombre total d'insertions dans la table alerts.status.
StatusNewInserts	integer	Nombre de nouvelles insertions dans la table alerts.status.
StatusDedups	integer	Nombre de réinsertions dans la table alerts.status.
JournalInserts	integer	Nombre d'insertions dans la table alerts.journal.
DetailsInserts	integer	Nombre d'insertions dans la table alerts.details.
StatusUpdates	integer	Nombre de mises à jour de la table alerts.status.

Table catalog.trigger_stats

La table catalog.trigger_stats stocke des informations de temporisation sur les déclencheurs, y compris le nombre de fois où le déclencheur a été mis au premier plan et le nombre de fois où il a été mis en application. Ces statistiques sont recueillies sauf si le système d'automatisation est désactivé, c'est-à-dire, sauf si l'option de ligne de commande `-autoenabled` est définie sur `FALSE`.

Les statistiques des déclencheurs sont également consignées dans le fichier `$NCHOME/omnibus/log/`*`nom de serveur`*`_trigger_stats.log`*`n`*.

Le tableau suivant décrit les colonnes de la table catalog.trigger_stats.

Tableau 143. Colonnes de la table catalog.trigger_stats

Nom de colonne	Type de données	Description
TriggerName	varchar(40)	Nom du déclencheur.
PreviousCondition	Boolean	Valeur de la condition lors de la dernière mise au premier plan du déclencheur.
PreviousRowcount	unsigned	Nombre de lignes renvoyées par la clause EVALUATE lors de la dernière mise au premier plan du déclencheur.
NumZeroRowcount	unsigned	Nombre de fois consécutives où l'évaluation n'a renvoyé aucune ligne.
NumPositiveRowcount	unsigned	Nombre de fois consécutives où l'évaluation a renvoyé une ou plusieurs lignes.
PeriodNumRaises	unsigned	Nombre de fois où le déclencheur a été mis au premier plan depuis le dernier rapport.

Tableau 143. Colonnes de la table catalog.trigger_stats (suite)

Nom de colonne	Type de données	Description
PeriodNumFires	unsigned	Nombre de fois où le déclencheur a été mis en application depuis le dernier rapport.
PeriodTime	real	Durée de fonctionnement du déclencheur depuis le dernier rapport.
NumRaises	unsigned	Nombre de fois où le déclencheur a été mis au premier plan.
NumFires	unsigned	Nombre de fois où le déclencheur a été mis en application.
MaxTime	real	Durée maximale d'exécution du déclencheur.
TotalTime	real	Durée de fonctionnement du déclencheur depuis le démarrage.

Remarque : La table catalog.trigger_stats est mise à jour de manière périodique, en fonction du paramètre indiqué pour la propriété **Auto.StatsInterval** ou l'option de ligne de commande `-autostatsinterval`. La valeur par défaut est toutes les 10 secondes.

Table catalog.channel_stats

La table catalog.channel_stats contient une entrée pour chaque canal actuellement connu dans le serveur ObjectServer. Chaque entrée de la table présente des détails sur le canal en deux parties : les statistiques de la période de collecte de statistiques actuelle et le total statistique depuis le démarrage du serveur ObjectServer.

Le tableau suivant décrit les colonnes de la table catalog.channel_stats.

Tableau 144. Colonnes de la table catalog.channel_stats

Nom de colonne	Type de données	Description
ChannelName	varchar(64)	Nom du canal.
LastTimingAt	time	Heure à laquelle la dernière mise à jour des statistiques a été effectuée . Elle est stockée sous forme de temps universel coordonné.
PeriodNumMsgs	int	Nombre de fois où ce canal a été utilisé dans la dernière fenêtre de profil.
PeriodTime	real	Délai pour envoyer des messages à tous les clients dans la dernière fenêtre de période de profil.
PeriodMaxTime	real	Délai maximum pour envoyer des messages à tous les clients dans la dernière fenêtre de période de profil.
PeriodClientNum	int	Nombre de clients auxquels des messages ont été envoyés lors du dernier appel.
PeriodMaxClientNum	int	Nombre maximum de clients auxquels des messages ont été envoyés dans la dernière fenêtre de période de profil.
NumMsg	int	Nombre de fois où ce canal a été utilisé depuis le démarrage du serveur.
TotalTime	real	Durée totale nécessaire pour envoyer des messages à tous les clients depuis le démarrage du serveur.
MaxTime	real	Durée maximale nécessaire pour envoyer des messages à tous les clients depuis le démarrage du serveur.
MaxClientNum	int	Nombre maximum de clients auxquels des messages ont été envoyés depuis le démarrage du serveur.

Tableau 144. Colonnes de la table catalog.channel_stats (suite)

Nom de colonne	Type de données	Description
TotalClientNum	int	Nombre total de clients auxquels des messages ont été envoyés depuis le démarrage du serveur.

Tables de prise en charge d'outils client

Les tables de prise en charge d'outils client sont utilisées par les interfaces graphiques pour afficher des informations d'alerte.

Table alerts.resolutions

La table alerts.resolutions permet de conserver l'option **Resolutions (Résolutions)** dans la liste d'événements.

Le tableau suivant décrit les colonnes de la table alerts.resolutions.

Tableau 145. Colonnes de la table alerts.resolutions

Nom de colonne	Type de données	Description
KeyField	varchar(255)	Clé principale de la table.
Tag	integer	Valeur de classe de cette résolution.
Sequence	integer	Numéro de séquence qui définit le tri à l'heure d'affichage.
Title	varchar(64)	Titre de la résolution.
Resolution1	varchar(255)	Première ligne de texte de la résolution.
Resolution2	varchar(255)	Deuxième ligne de texte de la résolution.
Resolution3	varchar(255)	Troisième ligne de texte de la résolution.
Resolution4	varchar(255)	Quatrième ligne de texte de la résolution.

Table alerts.conversions

La table alerts.conversions permet de convertir facilement une valeur numérique en chaîne dans n'importe quelle colonne.

Le tableau suivant décrit les colonnes de la table alerts.conversions.

Tableau 146. Colonnes de la table alerts.conversions

Nom de colonne	Type de données	Description
KeyField	varchar(255)	Clé primaire de la table. Chaîne d'organisation interne (composée de Colname et de Value).
Colname	varchar(255)	Nom de la colonne pour laquelle cette conversion est appropriée.
Value	integer	Valeur numérique de la conversion.
Conversion	varchar(255)	Valeur de chaîne de la conversion.

Table alerts.col_visuals

La table alerts.col_visuals permet de fournir des visuels par défaut pour les colonnes lorsqu'elles sont affichées dans les outils de bureau.

Le tableau suivant décrit les colonnes de la table alerts.col_visuals.

Tableau 147. Colonnes de la table alerts.col_visuals

Nom de colonne	Type de données	Description
Colname	varchar(255)	Nom de la colonne des paramètres visuels.
Title	varchar(255)	Titre de la colonne lorsqu'elle est affichée.
DefWidth	integer	Largeur par défaut de la colonne lorsqu'elle est affichée.
MaxWidth	integer	Largeur maximale de la colonne lorsqu'elle est affichée.
TitleJustify	integer	Justification du titre de colonne : 0 : gauche 1 : centré 2 : droite
DataJustify	integer	Justification des données de colonne : 0 : gauche 1 : centré 2 : droite

Table alerts.colors

La table alerts.colors permet de créer les couleurs requises par le bureau Windows.

Le tableau suivant décrit les colonnes de la table alerts.colors.

Tableau 148. Colonnes de la table alerts.colors

Nom de colonne	Type de données	Description
Severity	integer	Gravité du problème : 0 : Effacer I1 : Indéterminé 2 : Avertissement 3 : Mineur 4 : Majeur 5 : Critique
AckedRed	integer	Composant rouge de la couleur RVB des événements avec accusé réception. Doit être compris entre 0 et 255.
AckedGreen	integer	Composant vert de la couleur RVB des événements avec accusé réception. Doit être compris entre 0 et 255.
AckedBlue	integer	Composant bleu de la couleur RVB des événements avec accusé réception. Doit être compris entre 0 et 255.
UnackedRed	integer	Composant rouge de la couleur RVB des événements sans accusé réception. Doit être compris entre 0 et 255.

Tableau 148. Colonnes de la table alerts.colors (suite)

Nom de colonne	Type de données	Description
UnackedGreen	integer	Composant vert de la couleur RVB des événements sans accusé réception. Doit être compris entre 0 et 255.
UnackedBlue	integer	Composant bleu de la couleur RVB des événements sans accusé réception. Doit être compris entre 0 et 255.

Tables d'outils de bureau

Les tables d'outils de bureau contiennent des informations permettant de configurer des outils de listes d'événements.

Table tools.actions

La table tools.actions permet de contrôler les outils de bureau.

Le tableau suivant décrit les colonnes de la table tools.actions.

Tableau 149. Colonnes de la table tools.actions

Nom de colonne	Type de données	Description
ActionID	integer	Identificateur de l'outil.
Name	varchar(64)	Nom de l'outil.
Owner	integer	Indique si l'outil a un propriétaire ou non.
Enabled	integer	Indique si l'outil est activé ou non.
Description1	varchar(255)	Première ligne de la description.
Description2	varchar(255)	Deuxième ligne la description.
Description3	varchar(255)	Troisième ligne la description.
Description4	varchar(255)	Quatrième ligne la description.
HasInternal	integer	Indique si l'outil a un effet interne ou non.
InternalEffect1	varchar(255)	Première ligne de l'effet interne.
InternalEffect2	varchar(255)	Deuxième ligne de l'effet interne.
InternalEffect3	varchar(255)	Troisième ligne de l'effet interne.
InternalEffect4	varchar(255)	Quatrième ligne de l'effet interne.
InternalForEach	integer	Lorsque cette colonne est définie, l'effet interne est démarré pour chaque ligne sélectionnée.
HasExternal	integer	Indique si l'outil a une procédure externe.
ExternalEffect1	varchar(255)	Première ligne de la procédure externe.
ExternalEffect2	varchar(255)	Deuxième ligne de la procédure externe.
ExternalEffect3	varchar(255)	Troisième ligne de la procédure externe.
ExternalEffect4	varchar(255)	Quatrième ligne de la procédure externe.
ExternalForEach	integer	Lorsque cette colonne est définie, la procédure externe est démarrée pour chaque ligne sélectionnée.
RedirectOut	integer	Lorsque cette colonne est sélectionnée, la sortie est répercutée via une fenêtre en lecture seule dans le même affichage que la liste d'événements qui a exécuté l'outil.

Tableau 149. Colonnes de la table tools.actions (suite)

Nom de colonne	Type de données	Description
RedirectErr	integer	Lorsque cette colonne est sélectionnée, les erreurs sont répercutées via une fenêtre en lecture seule dans le même affichage que la liste d'événements qui a exécuté l'outil.
Platform	varchar(255)	Indique le type de système d'exploitation sur lequel la procédure externe s'exécute : `NT` (plateformes Windows) `UNIX` (plateformes UNIX) `UNIX, NT` (plateformes UNIX et Windows)
JournalText1	varchar(255)	Première ligne de l'entrée de journal.
JournalText2	varchar(255)	Deuxième ligne de l'entrée de journal.
JournalText3	varchar(255)	Troisième ligne de l'entrée de journal.
JournalText4	varchar(255)	Quatrième ligne de l'entrée de journal.
JournalForEach	integer	Lorsque cette colonne est définie, l'entrée de journal est ajoutée pour chaque ligne sélectionnée.
HasForcedJournal	integer	Force l'ouverture d'une fenêtre d'entrée de journal lors de l'exécution de l'outil.

Table tools.action_access

La table tools.action_access permet de contrôler l'accès aux outils de bureau.

Le tableau suivant décrit les colonnes de la table tools.action_access.

Tableau 150. Colonnes de la table tools.action_access

Nom de colonne	Type de données	Description
ActionID	integer	Identificateur unique de l'outil, issu de la table d'actions.
GID	integer	Indique le groupe pour lequel l'outil est disponible.
ClassID	integer	Indique pour quelle classe l'outil est disponible.
ActionAccessID	integer	Zone de clé primaire de la table.

Table tools.menus

La table tools.menus permet de contrôler les menus des outils de bureau.

Le tableau suivant décrit les colonnes de la table tools.menus.

Tableau 151. Colonnes de la table tools.menus

Nom de colonne	Type de données	Description
MenuID	integer	Zone de clé primaire de la table de menus.
Name	varchar(64)	Nom du menu.
Owner	integer	Indique si le menu a un propriétaire.
Enabled	integer	Indique si le menu est activé ou désactivé.

Table tools.menu_items

La table tools.menu_items permet de contrôler les éléments de menu des outils de bureau.

Le tableau suivant décrit les colonnes de la table tools.menu_items.

Tableau 152. Colonnes de la table tools.menu_items

Nom de colonne	Type de données	Description
KeyField	varchar(32)	Zone clé de cet élément de menu. Créé à partir de la colonne `menu_id` (de la table tools.menus) et de la colonne `menu_item_id`.
MenuID	integer	Identificateur unique du menu issu de la table tools.menus.
MenuItemID	integer	Identificateur de clé primaire de cet élément de menu.
Title	varchar(64)	Nom qui s'affiche sur le menu.
Description	varchar(255)	Description de l'élément de menu.
Enabled	integer	Indique si l'élément de menu est activé ou désactivé.
InvokeType	integer	Indique le type d'élément de menu : `0` : outil `1` : séparateur de ligne `2` : sous-menu
InvokeID	integer	Indique l'identificateur d'action de l'action définie dans la colonne InvokeType.
Position	integer	Indique la position (ordre) de cet élément dans le menu.
Accelerator	varchar(32)	Indique le raccourci-clavier de cet élément de menu.

Table tools.prompt_defs

La table tools.prompt_defs permet de stocker toutes les définitions d'invites.

Le tableau suivant décrit les colonnes de la table tools.prompt_defs.

Tableau 153. Colonnes de la table tools.prompt_defs

Nom de colonne	Type de données	Description
Name	varchar(64)	Nom de l'invite.
Prompt	varchar(64)	Titre de l'invite qui s'affiche en haut de la fenêtre d'invite.
Default	varchar(64)	Valeur par défaut à entrer si aucune valeur n'a été saisie par l'utilisateur.
Value	varchar(255)	Liste des valeurs disponibles.
Type	integer	Type d'invite. Le type 7 est un lien vers une définition de menu portant le même nom que l'invite.

Table tools.menu_defs

La table tools.menu_defs permet de contrôler les éléments du menu des outils de bureau.

Le tableau suivant décrit les colonnes de la table tools.menu_defs.

Tableau 154. Colonnes de la table tools.menu_defs

Nom de colonne	Type de données	Description
Name	varchar(64)	Nom du menu. Il doit correspondre au nom de la définition d'invite de type 7.
DatabaseName	varchar(64)	Base de données utilisée pour générer les éléments de menu.
TableName	varchar(64)	Table utilisée pour générer les éléments de menu.
ShowField	varchar(64)	Zone de la table à afficher comme liste déroulante de menu.
AssignField	varchar(64)	Zone actuelle utilisée pour entrer une valeur dans l'invite.
OrderByField	varchar(64)	Zone utilisée pour trier le menu.
WhereClause	varchar(255)	Filtre (condition) permettant d'afficher un sous-ensemble des éléments de menu.
Direction	integer	Ordre des éléments de menu : 0 : Croissant 1 : Décroissant

Tables du serveur ObjectServer de bureau

La table master.national est utilisée dans une architecture du serveur ObjectServer de bureau. Cette table est créée lors de l'exécution de la commande **nco_dbinit** à l'aide de l'option -desktopserver. La table master.servergroups permet d'équilibrer les charges des connexions de bureau.

Table master.national

La table master.national identifie le serveur ObjectServer maître et le mode d'écriture double dans une architecture du serveur ObjectServer de bureau.

Le tableau suivant décrit les colonnes de la table master.national.

Tableau 155. Colonnes de la table master.national

Nom de colonne	Type de données	Description
KeyField	incr	Colonne de clé primaire de la table.
MasterServer	varchar(29)	Nom du serveur ObjectServer maître dans une architecture de serveur ObjectServer de bureau.
DualWrite	integer	Si le mode d'écriture double est activé. Ce mode permet aux opérateurs de consulter rapidement les résultats des actions d'outils (par exemple, les accusés réception et la définition d'une priorité) sur leurs bureaux à deux serveurs. Pour ce faire, toutes les actions d'outils sont envoyées au serveur ObjectServer de bureau et au serveur ObjectServer maître. 1 : activé 0 : désactivé

Table master.servergroups

La table master.servergroups permet d'équilibrer les charges des connexions de bureau.

Le tableau suivant décrit les colonnes de la table master.servergroups.

Tableau 156. Colonnes de la table master.servergroups

Nom de colonne	Type de données	Description
ServerName	varchar(64)	Nom d'un serveur ObjectServer de bureau.
Group ID	integer	Identificateur du groupe auquel le serveur ObjectServer de bureau appartient. Les données de connexion des utilisateurs de listes d'événements sont uniquement distribuées aux serveurs ObjectServer de bureau ayant le même ID groupe.
Weight	integer	Priorité de chaque ObjectServer de bureau. Les valeurs plus élevées attirent proportionnellement plus de connexions. Par exemple, un serveur ObjectServer avec une pondération de 2 attire deux fois plus de connexions qu'un serveur avec une pondération de 1. Les connexions à charges équilibrées ne sont pas réacheminées vers un serveur ObjectServer d'une pondération de 0.

Tables de sécurité pour la compatibilité amont

Dans Netcool/OMNIbus V3.6, la base de données `maître` contenait des tables d'authentification d'utilisateur pour stocker des informations de sécurité relatives à Netcool/OMNIbus. Ces informations sont maintenant stockées dans la base de données de sécurité et la table catalog.security_permissions.

Les tables master.names, master.members et master.groups proposaient des fonctionnalités d'identification et d'autorisation des groupes et des utilisateurs. La table master.profiles principale fournissait des informations de restriction des utilisateurs. Ces tables doivent uniquement être compatibles avec les versions précédentes du bureau.

Tables IDUC

La base de données iduc_system stocke toutes les tables de prise en charge d'applications IDUC requises pour accélérer la notification des événements, l'envoi de messages d'information et les heures de mise à jour IDUC.

Table iduc_system.channel

La table iduc_system.channel définit l'ensemble de canaux IDUC connus au sein du serveur ObjectServer.

Le tableau suivant décrit les colonnes de la table iduc_system.channel.

Tableau 157. Colonnes de la table iduc_system.channel

Nom de colonne	Type de données	Description
Name	varchar(64)	Nom de canal.
ChannelID	int	Clé ajoutée pour optimiser les références aux détails sur le canal stockés dans les tables associées.
Description	varchar(2048)	Description du canal.

Table iduc_system.channel_interest

La table iduc_system.channel_interest stocke les entrées de canal d'intérêt d'un canal donné. Il peut y avoir plusieurs entrées d'intérêt par canal.

Le tableau suivant décrit les colonnes de la table iduc_system.channel_interest.

Tableau 158. Colonnes de la table iduc_system.channel_interest

Nom de colonne	Type de données	Description
Element Name	varchar(64)	Nom des destinataires du canal. Il s'agit d'un nom d'utilisateur ou de groupe.
IsGroup	int	Indique si les destinataires sont un groupe d'utilisateurs.
Hostname	varchar(255)	Nom d'hôte du client IDUC.
AppName	varchar(255)	Nom de l'application du client IDUC.
AppDescription	varchar(255)	Description de l'application du client IDUC.
ChannelID	int	ID de canal.

Table iduc_system.channel_summary

La table iduc_system.channel_summary est uniquement utilisée pour une commande du client IDUC de suivi rapide d'un événement (ou d'événement accéléré). Les lignes de n'importe quelle table du serveur ObjectServer peuvent être réacheminées sous forme d'événements accélérés. Cette table permet l'association d'un canal à plusieurs tables à partir desquelles les événements peuvent être accélérés.

Le tableau suivant décrit les colonnes de la table iduc_system.channel_summary.

Tableau 159. Colonnes de la table iduc_system.channel_summary

Nom de colonne	Type de données	Description
DatabaseName	varchar(64)	Base de données à laquelle la table appartient.
TableName	varchar(64)	Nom de la table.
SummaryID	int	Clé entière ajoutée pour optimiser les références à la table de colonnes récapitulative.
ChannelID	int	ID de canal.

Table iduc_system.channel_summary_cols

La table iduc_system.channel_summary_cols stocke des détails sur les colonnes exactes qui constituent le récapitulatif réel d'une table donnée.

Le tableau suivant décrit les colonnes de la table iduc_system.channel_summary_cols.

Tableau 160. Colonnes de la table iduc_system.channel_summary_cols

Nom de colonne	Type de données	Description
ColumnName	varchar(64)	Nom d'une colonne faisant partie d'une définition récapitulative donnée.
Position	int	Position de la colonne dans l'ordre récapitulatif.
SummaryID	int	ID de récapitulatif.
ChannelID	int	ID de canal.

Table iduc_system.iduc_stats

La table iduc_system.iduc_stats stocke des détails sur la dernière heure à laquelle des modifications IDUC ont été transmises à un client IDUC.

Le tableau suivant décrit les colonnes de la table iduc_system.iduc_stats.

Tableau 161. Colonne de la table iduc_system.iduc_stats

Nom de colonne	Type de données	Description
ServerName	varchar(40)	Nom du serveur ObjectServer auquel le client IDUC se connecte. Clé primaire.
AppName	varchar(40)	Nom de l'application du client IDUC.
AppDesc	varchar(128)	Description de l'application du client IDUC. Clé primaire.
ConnectionId	int	Identifie la connexion de manière unique. Clé primaire.
LastIducTime	UTC	Heure à laquelle les modifications IDUC ont été transmises pour la dernière fois au client IDUC. Cette colonne est mise à jour lorsque la passerelle extrait des données du serveur ObjectServer.

Tables d'événements affectés par un service

Les tables d'événements affectés par un service fournissent une prise en charge des événements affectés par un service générés dans Network Manager IP Edition. Un événement affecté par un service est une alerte qui avertit les opérateurs qu'un service client critique a été affecté par un ou plusieurs événements réseau.

Table precision.service_affecting_event

La table precision.service_affecting_event stocke l'identificateur d'un événement affecté par un service généré dans Network Manager IP Edition.

Le tableau suivant décrit les colonnes de la table precision.service_affecting_event.

Tableau 162. Colonne de la table precision.service_affecting_event

Nom de colonne	Type de données	Description
ServiceEntityId	integer	Colonne de clé primaire de la table. ID d'un événement affecté par un service.

Table precision.service_details

La table precision.service_details stocke les détails d'un événement affecté par un service généré dans Network Manager IP Edition.

Le tableau suivant décrit les colonnes de la table precision.service_details.

Tableau 163. Colonnes de la table precision.service_details

Nom de colonne	Type de données	Description
ServiceEntityId	integer	Colonne de clé principale de la table. ID d'un événement affecté par un service.
Type	varchar(255)	Type de service.
Name	varchar(255)	Nom de service.
Customer	varchar(255)	Client associé.

Table precision.entity_service

La table precision.entity_service mappe les identificateurs des événements affectés par un service aux ID numériques qui identifient de manière unique les entités de la topologie de Network Manager IP Edition auxquelles les événements sont associés.

Le tableau suivant décrit les colonnes de la table precision.entity_service.

Tableau 164. Colonnes de la table precision.entity_service

Nom de colonne	Type de données	Description
NmosEntityId	integer	Colonne de clé principale de la table. ID numérique qui identifie l'entité de topologie de Network Manager IP Edition à laquelle un événement affecté par un service est associé.
ServiceEntityId	integer	Colonne de clé principale de la table. ID d'un événement affecté par un service.

Tables de registre

Les tables de registre contiennent des informations sur les services qui sont connectés au serveur ObjectServer.

Table registry.oslc

Cette table est utilisée pour configurer et gérer des inscriptions de fournisseurs de services OSLC dans les registres de services IBM JazzSM.

Tableau 165. Table d'inscription de fournisseurs de services OSLC registry.oslcsp.

Colonne	Type	Description
Name	VARCHAR(64)	Nom défini par l'utilisateur de l'entrée de table d'inscription.
RegistryURI	VARCHAR(1024)	Enregistrement des services du fournisseur de services OSLC du service de registre. RegistryURI est la clé principale de la table.
RegistryUsername	VARCHAR(64)	Utilisateur permettant de s'authentifier sur le registre de fournisseurs de services JazzSM.
RegistryPassword	VARCHAR(64)	Mot de passe permettant de s'authentifier sur le registre de fournisseurs de services JazzSM.

Tableau 165. Table d'inscription de fournisseurs de services OSLC registry.oslcsp. (suite)

Colonne	Type	Description
Registered	entier	Indique si l'entrée possède un enregistrement d'inscription auprès du service de registre JazzSM. Les valeurs possibles sont les suivantes : • `0` : l'entrée n'a pas d'inscription. • `1` : l'entrée a une inscription.
RegistrationURI	VARCHAR(1024)	URI de l'enregistrement d'inscription dans le registre de fournisseurs de services JazzSM pour cet ObjectServer.
LastRegistered	time	Date et heure de la dernière inscription réussie dans le registre de fournisseurs de services JazzSM.

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 deviennent persistantes lorsque vous appliquez la mise à jour de schéma ObjectServer `update74to81.sql`.

Remarque : Toutes les données de la table virtuelle existante registry.probes sont perdues durant la mise à jour du schéma.

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.

Le tableau suivant décrit les colonnes de la table registry.probes.

Tableau 166. Colonnes de la table registry.probes

Nom de colonne	Type de données	Description
ApiReleaseID	VARCHAR(20)	Identificateur d'édition de la bibliothèque d'API de sonde.

Tableau 166. Colonnes de la table registry.probes (suite)

Nom de colonne	Type de données	Description
ApiVersion	VARCHAR(20)	Version de la bibliothèque d'API de sonde.
ConnectionID	entier	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. Lorsque l'ID de connexion est 0, la sonde n'est pas connectée au serveur ObjectServer. Dans les déploiements multiniveau et de reprise en ligne, l'ID de connexion est uniquement défini sur les serveurs ObjectServer principal et de collecte. Dans une configuration de reprise en ligne, l'ID de connexion sur le serveur de sauvegarde ObjectServer est toujours 0. Dans un déploiement multiniveau, l'ID de connexion sur les serveurs ObjectServer d'affichage et d'agrégation est toujours 0. **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.
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.SSLListeningPort** 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	heure	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	entier	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 166. 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 inclut des consultations (lookup) ou des inclusions (include), le total de contrôle est une combinaison des totaux de contrôle des règles individuelles, des fichiers de consultation et des fichiers d'inclusion. Cette valeur change lorsque le contenu du fichier de règles change. **Remarque :** Si un fichier de règles est mis en cache, la valeur du total de contrôle est conservée dans un commentaire de métadonnées dans le fichier mis en cache. Ne modifiez pas cette valeur dans le fichier mis en cache. Voici un exemple de commentaire de métadonnées de total de contrôle : `#METADATA CheckSum=f967b187bf66e08d4277ab688f87f696bea3bb22`
StartTime	heure	Heure de démarrage de la sonde. Cette information vous permet de déterminer si la sonde est en cours de démarrage ou de reconnexion.
Status	entier	Indique le statut de la sonde : `0` : La sonde est arrêtée. `1` : La sonde est en cours d'exécution.

Référence associée:

«Automatisations Tivoli Netcool/OMNIbus standard», à la page 196
Un ensemble d'automatisations standard est inclus à Tivoli Netcool/OMNIbus. Ces automatisations sont créées lors de l'initialisation de la base de données.

Tables de surveillance automatique

Les tables de surveillance automatique contiennent les données requises pour l'exécution de la fonction de surveillance automatique de Tivoli Netcool/OMNIbus.

Table master.sm_activity

La table master.sm_activity stocke la date/heure à laquelle la dernière entrée dans la table master.stats a été traitée par chaque déclencheur lors de sa dernière exécution. Ces données sont utilisées par la fonction de surveillance automatique de Tivoli Netcool/OMNIbus.

Tableau 167. Colonnes de la table master.sm_activity

Nom de colonne	Type de données	Description
ReportTrigger	char(255)	Nom du déclencheur. Il s'agit de la colonne de clé principale.
ProbeStatsLastSeqNos	Entier	Cette colonne n'est actuellement pas utilisée.
MasterStatsLast	Heure	Date/heure à laquelle la dernière entrée dans la table master.stats a été traitée.

Concepts associés:

«Surveillance automatique de Tivoli Netcool/OMNIbus», à la page 326
Vous pouvez surveiller une plage d'indicateurs clés de performances (KPI) au sein de l'infrastructure du serveur ObjectServer en activant l'extension de surveillance automatique de Tivoli Netcool/OMNIbus. Les données de performances sont signalées sous forme d'événements synthétiques et peuvent être affichées dans la liste d'événements ou dans un tableau de bord de l'interface graphique Web Netcool Health. Les données d'événement sont également consignées dans un fichier journal de surveillance automatique.

Table master.sm_thresholds

La table master.sm_thresholds contient les valeurs de seuil qui définissent la génération d'événements de surveillance automatique Tivoli Netcool/OMNIbus.

Tableau 168. Colonnes de la table master.sm_thresholds

Nom de colonne	Type de données	Description
ThresholdName	char(64)	Nom du seuil. Il s'agit de la colonne de clé principale.
DisplayName	char(64)	Version à affichage convivial du nom de seuil.
Sev1	Entier	Le seuil à partir duquel les alertes de gravité 1 sont générés. **Remarque :** Cette colonne n'est actuellement pas utilisée.
Sev2	Entier	Le seuil à partir duquel les alertes de gravité 2 sont générés. **Remarque :** Cette colonne n'est actuellement pas utilisée.
Sev3	Entier	Le seuil à partir duquel les alertes de gravité 3 sont générés.
Sev4	Entier	Le seuil à partir duquel les alertes de gravité 4 sont générés.
Sev5	Entier	Le seuil à partir duquel les alertes de gravité 5 sont générés.
EnableInfo	Entier	Indique si le seuil génère des événements d'information en sus des événements d'alerte : • `0` : les événements d'information ne sont pas générés • `1` : les événements d'information sont générés La valeur par défaut est 1.
DeduplicateInfo	Entier	Indique si le seuil dédouble les événements d'information : • `0` : les événements d'information ne sont pas dédoublonnés • `1` : les événements d'information sont dédoublonnés La valeur par défaut est 1. Les événements d'information sont dédoublonnés par défaut pour empêcher la génération d'un nombre excessif d'événements individuels. Vous pouvez désactiver le dédoublonnage des événements d'information si, par exemple, les mesures du seuil doivent être représentées sous forme graphique à partir de Tivoli Data Warehouse sur une période de temps. Vous pouvez désactiver le dédoublonnage pour cette période puis réactiver le dédoublonnage par la suite.

Concepts associés:

«Surveillance automatique de Tivoli Netcool/OMNIbus», à la page 326
Vous pouvez surveiller une plage d'indicateurs clés de performances (KPI) au sein de l'infrastructure du serveur ObjectServer en activant l'extension de surveillance automatique de Tivoli Netcool/OMNIbus. Les données de performances sont signalées sous forme d'événements synthétiques et peuvent être affichées dans la liste d'événements ou dans un tableau de bord de l'interface graphique Web Netcool Health. Les données d'événement sont également consignées dans un fichier journal de surveillance automatique.

Annexe E. Conseils pour les messages d'erreur

Cette annexe décrit les messages d'erreur qui apparaissent le plus fréquemment dans les fichiers journaux pour les composants du serveur et l'interface graphique Web et qui sont consignés dans les rapports PMR (Problem Management Reports). Des exemples de messages d'erreur sont joints avec des conseils quant aux mesures à prendre suite à ces messages. Les messages sont consignés dans différents fichiers journaux de composants Tivoli Netcool/OMNIbus, ce qui inclut les fichiers journaux d'ObjectServer, de passerelle et de sonde.

Le tableau suivant répertorie les types de message qui sont consignés par les applications Tivoli Netcool/OMNIbus.

Tableau 169. Types de message

Type de message	**Exemple de code message**
Debug (Débogage)	`D-ETC-004-051`
Error (Erreur)	`E-IVM-005-001`
Fatal	`F-IVM-005-001`
Information	`I-GOB-104-027`
Warning (Avertissement)	`W-STO-103-001`

En fonction de l'application, le niveau de consignation des messages est spécifié par une propriété **`MessageLevel`** ou une option de ligne de commande ou `-messagelevel` (ou les deux). Le niveau de message que vous spécifiez détermine quels types de messages sont consignés. Le niveau `debug` est le plus prolixe et peut générer des fichiers journaux volumineux. Le niveau par défaut standard est `warn`.

Codes message

Les codes message Tivoli Netcool/OMNIbus sont séparés en catégories de composants.

Le tableau suivant répertorie les codes message côté serveur en cours.

Tableau 170. Codes message Tivoli Netcool/OMNIbus

Code message	**Description**
ALD	Messages relatifs à l'authentification LDAP.
APM	Messages relatifs à l'authentification PAM (Pluggable Authentication Modules).
AUT	Messages relatifs aux automatisations et aux déclencheurs.
BRI	Messages relatifs à l'utilitaire de serveur pare-feu (**`nco_bridgeserv`**).
CHE	Messages relatifs à l'utilitaire de vérification de point de contrôle de magasin de mémoire (**`nco_check_store`**).
CMD	Messages relatifs au traitement de commandes SQL.

© Copyright IBM Corp. 1994, 2014

Tableau 170. Codes message Tivoli Netcool/OMNIbus (suite)

Code message	Description
DBI	Messages relatifs à l'utilitaire d'initialisation de la base de données (**nco_dbinit**).
ETC	Messages relatifs à la bibliothèque Netcool/OMNIbus centrale `libnetcool`.
GOB	Messages relatifs aux passerelles ObjectServer.
GTK	Messages relatifs au kit d'outils NGtk de la passerelle.
HTT	Messages relatifs à l'interface HTTP/HTTPS.
IDC	Messages relatifs aux clients IDUC.
IDS	Messages relatifs aux serveurs IDUC.
IPC	Messages relatifs à la communication interprocessus (IPC).
IVM	Messages relatifs à la communication interprocessus (IPC).
MEM	Messages relatifs au magasin de mémoire.
NHT	Messages relatifs à l'interface HTTP/HTTPS.
OAM	Messages relatifs à l'authentification.
OBJ	Messages relatifs à l'ObjectServer.
OBX	Messages relatifs à l'application ObjectServer (**nco_objserv**).
POS	Messages relatifs à l'utilitaire **nco_postmsg**.
PRO	Messages relatifs à l'agent de processus.
PRX	Messages relatifs à l'utilitaire de serveur proxy (**nco_proxyserv**).
RCL	Messages relatifs aux clients HTTP and OSLC.
RCO	Messages relatifs aux interfaces HTTP and OSLC.
REG	Messages relatifs au stockage de base de données d'ObjectServer.
REP	Messages relatifs à l'utilitaire Probe Rules Utility (**nco_rules_util**).
RSE	Messages relatifs aux serveurs HTTP and OSLC.
SEC	Messages relatifs à la sécurité.
STB	Messages relatifs à la sécurité.
STK	Messages relatifs aux serveurs.
STO	Messages relatifs au stockage de base de données d'ObjectServer.
STR	Messages relatifs à l'utilitaire **nco_store_resize**.
UNK	Messages relatifs aux problèmes non classifiés.

Le tableau suivant répertorie les codes message de l'interface graphique Web en cours.

Tableau 171. Codes message interface graphique Web

Code message	Description
HEMAW	Messages relatifs au composant HEMAW.
HEMBT	Messages relatifs au composant HEMBT.
HEMDA	Messages relatifs au composant HEMDA.
HEMEC	Messages relatifs au composant HEMEC.
HEMEE	Messages relatifs au composant HEMEE.

Tableau 171. Codes message interface graphique Web (suite)

Code message	Description
HEMEM	Messages relatifs au composant HEMEM.
HEMEN	Messages relatifs au composant HEMEN.
HEMEO	Messages relatifs au composant HEMEO.
HEMEP	Messages relatifs au composant HEMEP.
HEMES	Messages relatifs au composant HEMES.
HEMEU	Messages relatifs au composant HEMEU.
HEMFV	Messages relatifs au composant HEMFV.
HEMHA	Messages relatifs au composant HEMHA.
HEMHB	Messages relatifs au composant HEMHB.
HEMIM	Messages relatifs au composant HEMIM.
HEMIN	Messages relatifs au composant HEMIN.
HEMIP	Messages relatifs au composant HEMIP.
HEMJS	Messages relatifs au composant HEMJS.
HEMUP	Messages relatifs au composant HEMUP.

Messages d'erreur générés par les composants côté serveur

Les rubriques suivantes décrivent les réponses de l'administrateur pour un sous-ensemble de messages d'erreur générés par les composants côté serveur de Tivoli Netcool/OMNIbus.

Le catalogue des messages d'erreur complet est disponible sur le site de téléchargement Tivoli au format CSV :

ftp://ftp.software.ibm.com/software/tivoli/Netcool/NetcoolOmnibus/messages/

Remarque : Le délimiteur de zone dans le fichier CSV est le caractère # (connu sous le nom de signe «dièse» ou «numéro»).

ALD

Messages relatifs à l'authentification LDAP.

E-ALD-102-026 Echec de l'exécution de la recherche sur le serveur LDAP avec le nom distinctif de base *%0s* **et le filtre** *%1s* **:** *%2d* **:** *%3s*

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal.

Exemple 1 :

```
E-ALD-102-026 : Echec de l'exécution de la
recherche sur le serveur LDAP avec le nom
distinctif de base 'ou=bluepages,o=ibm.com'
et le filtre '(cn=Utilisateur test)':
81:Impossible de contacter le serveur LDAP
```

Exemple 2 :

```
E-ALD-102-026 : Echec de l'exécution de la
recherche sur le serveur LDAP avec le nom
distinctif de base 'ou="Tivoli",ou=SWG,o=ibm'
et filtre '(&(cn=Utilisateur test)
(|(ou=Tivoli)(ou=Webtop))':
87:Filtre de recherche incorrect
```

Le message d'exemple 1 indique qu'une recherche LDAP a échoué car l'ObjectServer n'a pas pu contacter le serveur LDAP. Le message d'exemple 2 indique qu'une recherche LDAP a échoué car un filtre de recherche non valide a été spécifié.

Réaction du système : L'utilisateur n'est pas authentifié.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour répondre à l'exemple de message 1, recherchez les problèmes de connectivité du réseau. Ce problème peut également se produire lorsque l'ObjectServer est configuré pour la liaison anonyme, mais que le serveur LDAP nécessite la sécurité de liaison. Si le problème n'est pas lié à la connexion réseau, vérifiez la configuration LDAP.

Pour résoudre l'exemple de message 2, vérifiez votre configuration LDAP. Contactez votre administrateur LDAP pour connaître les exigences de filtre de recherche adaptées à votre installation LDAP.

Cette erreur peut également être provoquée par plusieurs autres conditions. La chaîne d'erreur à la fin du message (`%2d:%3s`) donne une indication de la cause du problème.

Pour plus d'informations sur la configuration de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

E-ALD-102-027 Aucune entrée d'utilisateur LDAP trouvée avec le nom distinctif de base *%0s* **et le filtre** *%1s*

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-ALD-102-027 : Aucun utilisateur LDAP trouvé avec
le nom distinctif de base ou=Tivoli,ou=SWG,o=ibm
et le filtre (cn=User One)
```

L'exemple de message indique qu'une recherche LDAP basée sur les paramètres décrits n'a pas trouvé les détails de l'utilisateur dans LDAP.

Réaction du système : L'utilisateur n'est pas authentifié.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre ce problème, effectuez les opérations suivantes :

- Vérifiez que le nom d'utilisateur utilisé pour se connecter à l'ObjectServer est correct.
- Vérifiez que le nom distinctif de base et le filtre de recherche sont correctement spécifiés dans le fichier de propriétés LDAP. La propriété **`LDAPSearchBase`** spécifie le nom distinctif de base et la propriété **`LDAPSearchFilter`** spécifie le filtre de recherche.
- Contactez votre administrateur LDAP et vérifiez que l'utilisateur existe dans LDAP.

Pour plus d'informations sur la configuration et l'identification et la résolution des problèmes de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

E-ALD-102-028 Echec de l'authentification car plusieurs utilisateurs LDAP ont été trouvés avec le nom distinctif de base *%0s* **et le filtre** *%1s*

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-ALD-102-028 : Plusieurs utilisateurs LDAP avec
nom distinctif de base 'ou=Tivoli,ou=SWG,o=ibm'
et le filtre '(cn=User Two)'
```

L'exemple de message indique qu'un nom d'utilisateur n'est pas unique et existe dans plusieurs annuaires LDAP.

Réaction du système : L'utilisateur n'est pas authentifié.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre ce problème, modifiez la recherche LDAP afin qu'elle ne renvoie qu'un seul utilisateur. Si nécessaire, contactez votre administrateur LDAP pour la configuration requise de votre installation LDAP.

Pour plus d'informations sur la configuration de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

E-ALD-102-029 Echec de ldap_get_dn() avec le numéro d'erreur LDAP *%0d %1s*

Explication : Ce message est consigné lorsqu'un problème interne se produit lors de la conversion d'un résultat de recherche en nom distinctif. La chaîne de code d'erreur (`%0d %1s`) contient davantage de détails.

Réaction du système : L'utilisateur n'est pas authentifié.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Si ce problème persiste, contactez service de support logiciel IBM.

Pour plus d'informations sur la configuration et l'identification et la résolution des problèmes de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

E-ALD-102-030 La chaîne LDAPSearchFilter ne peut pas contenir plus d'une instance de `%s`

Explication : Ce message est consigné lorsque la valeur de la propriété **`LDAPSearchFilter`** contient plusieurs instances de la variable `%s`.

Lorsqu'il authentifie un utilisateur, ObjectServer remplace la variable `%s` dans le filtre de recherche par le nom d'utilisateur. Un filtre de recherche doit contenir une seule instance de la variable `%s`.

Réaction du système : ObjectServer ne démarre pas.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre ce problème, modifiez le filtre de recherche afin qu'il ne contienne qu'une seule instance de la variable `%s`.

Pour plus d'informations sur la configuration de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

E-ALD-102-031 La chaîne LDAPSearchFilter doit contenir `%s` (remplacé par le nom d'utilisateur en phase d'exécution)

Explication : Ce message est consigné lorsque la valeur de la propriété **`LDAPSearchFilter`** ne contient pas la variable `%s`.

Lorsqu'il authentifie un utilisateur, ObjectServer remplace la variable `%s` dans le filtre de recherche par le nom d'utilisateur. La variable `%s` est requise dans la chaîne de recherche LDAP.

Réaction du système : ObjectServer ne démarre pas.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre ce problème, modifiez le filtre de recherche afin qu'il contienne la variable `%s`. Une seule instance de la variable `%s` est autorisée.

Pour plus d'informations sur la configuration de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

E-ALD-102-032 Echec de l'analyse de LDAPSearchBase (*%0s***)**

Explication : Ce message est consigné lorsque l'ObjectServer ne peut pas analyser la valeur de la propriété **`LDAPSearchBase`**.

Réaction du système : ObjectServer ne démarre pas.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Vérifiez que la propriété **`LDAPSearchBase`** est correctement spécifiée dans le fichier de propriétés LDAP. Le fichier de propriétés LDAP `ldap.props` est dans le répertoire `$NCHOME/omnibus/etc/`.

Pour plus d'informations sur la configuration de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

E-ALD-102-033 La chaîne LDAPSearchBase ne doit pas contenir %s.

Explication : Ce message est consigné lorsque la valeur de la propriété **LDAPSearchBase** contient la variable %s.

Réaction du système : ObjectServer ne démarre pas.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Modifiez la propriété **LDAPSearchBase** pour supprimer toute instance de la variable %s. La variable %s doit être utilisée uniquement dans la propriété **LDAPSearchFilter**.

Pour plus d'informations sur la configuration de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

E-ALD-102-034 LDAPSearch a renvoyé NO_SUCH_OBJECT. Vérifiez que LDAPSearchBase est correctement spécifié et que l'objet de base de recherche *%0s* **existe sur le serveur LDAP.**

Explication : Ce message est consigné lorsqu'une recherche LDAP renvoie un code d'erreur NO_SUCH_OBJECT. Cette erreur se produit généralement lorsque le nom distinctif de base spécifié dans la requête de recherche est introuvable.

Réaction du système : Si plusieurs noms distinctifs sont spécifiés par la propriété **LDAPSearchBase**, l'ObjectServer continue la recherche du nom distinctif de base suivant. S'il n'y a plus de base de recherche, l'ObjectServer ne parvient pas à authentifier l'utilisateur.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Vérifiez que les noms distinctifs spécifiés par la propriété **LDAPSearchBase** correspondent aux objets qui existent dans le serveur LDAP.

Pour plus d'informations sur la configuration de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

W-ALD-103-002 LDAPSearchBase et DistinguishedName ont été spécifiés. Le paramètre DistinguishedName sera ignoré.

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
W-ALD-103-002 : LDAPSearchBase et
 DistinguishedName ont été spécifiées.
 Le paramètre DistinguishedName sera ignoré.
```

Le message d'exemple indique que les propriétés **LDAPSearchBase** et **DistinguishedName** sont activées dans le fichier de propriétés LDAP. L'ObjectServer ignore la propriété **DistinguishedName**.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Déterminez si vous nécessitez une liaison directe à LDAP ou si vous nécessitez une recherche LDAP pour authentifier les utilisateurs. Contactez votre administrateur LDAP pour déterminer la meilleure option.

Une authentification de liaison directe requiert la définition de la propriété **DistinguishedName**. Une recherche LDAP requiert la définition des propriétés **LDAPSearchBase** et **LDAPSearchFilter**. Le fichier de propriétés LDAP `ldap.props` est dans le répertoire `$NCHOME/omnibus/etc/`.

Pour plus d'informations sur la configuration de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

W-ALD-103-003 LDAPSearchFilter a été spécifié sans LDAPSearchBase. LDAPSearchFilter est ignoré et l'authentification de la liaison directe utilisée.

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
W-ALD-103-003 : LDAPSearchFilter a été spécifié
avec LDAPSearchBase. LDAPSearchFilter sera ignoré
et l'authentification de la liaison directe
utilisée.
```

Cet exemple de message indique que **LDAPSearchFilter** est activé dans le fichier de propriétés LDAP mais la propriété **LDAPSearchBase** n'est pas activée. L'ObjectServer ignore la propriété **LDAPSearchFilter** et utilise une liaison directe avec LDAP.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Si vous souhaitez utiliser une recherche LDAP pour authentifier un utilisateur, indiquez une valeur pour la propriété **LDAPSearchBase**. Si vous nécessitez une liaison directe à LDAP, laissez cette propriété **LDAPSearchFilter** vide. Le fichier de propriétés LDAP `ldap.props` est dans le répertoire `$NCHOME/omnibus/etc/`.

Pour plus d'informations sur la configuration de l'authentification LDAP, voir *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

APM

Messages relatifs à l'authentification PAM (Pluggable Authentication Modules).

E-APM-102-005 User authentication of user *%0s* **, has failed. [***%1d***][***%2s***]**

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-APM-102-005 : L'authentification de
l'utilisateur 'plp40289' a échoué.
[7][Authentication failure]
```

L'exemple de message indique que l'authentification d'un compte utilisateur sur le module Pluggable Authentication Module (PAM) a échoué.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Cette erreur peut être provoquée par différentes conditions. Les chaînes d'erreur indiquent le numéro de l'erreur PAM et la cause de l'erreur.

Examinez les messages relatifs dans le journal pour déterminer si l'échec d'authentification affecte un utilisateur unique ou tous les utilisateurs. Déterminez également si l'échec est spécifique de l'ObjectServer ou du référentiel d'utilisateurs.

Si un ObjectServer, un agent de processus ou un processus de passerelle ne s'exécute pas en tant que root, vous pouvez rencontrer des problèmes d'authentification sur les systèmes PAM externes. Pour plus d'informations, voir la section d'identification et résolution des problèmes du *Guide d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus*.

AUT

Messages relatifs aux automatisations et aux déclencheurs.

E-AUT-002-006 Trigger *%0s***\: action execution failed:** *%1s*

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal.

Exemple 1 :

```
E-AUT-002-006: Trigger "deleteitm":
 action execution failed:
 Attempt to insert duplicate row
```

Exemple 2 :

```
E-AUT-002-006:
 Trigger "connection_watch_disconnect":
 action execution failed:
 Not enough space
```

Exemple 3 :

```
E-AUT-002-006: Trigger "log_all_event":
action execution failed:
There is not enough memory available now.
The external procedure (log_event) was not
executed.
```

Les exemples de messages indiquent que les déclencheurs ObjectServer ne sont pas exécutés, peut-être parce que la mémoire est insuffisante pour traiter l'opération.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Dans l'exemple de message 1, un déclencheur de dédoublonnage n'existe pas pour la table dans laquelle l'événement est inséré. Dans ce cas, vous pouvez ajouter un déclencheur de dédoublonnage pour traiter les lignes en double dans la table concernée.

L'exemple de déclencheur suivant supprime les données d'événement en double entrantes et n'effectue pas de mises à jour :

```
create or replace trigger custom_dedup
 group default_triggers
 debug false
 enabled true
 priority 1
comment 'Deduplication processing for
 custom.status'
 before reinsert on custom.status
 for each row
 begin

 cancel;

end;
```

Dans les exemples de messages 2 et 3, un problème lié à la quantité de mémoire disponible a lieu. Surveillez l'utilisation de la mémoire du serveur ObjectServer et du système hôte sur une longue période pour déterminer si le système nécessite plus de mémoire physique.

Pour plus d'informations sur les déclencheurs et les automatisations, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Concepts associés:

«Déclencheurs», à la page 176
Vous pouvez utiliser les déclencheurs pour détecter les modifications du serveur ObjectServer et exécuter des réponses automatiques à ces modifications. Vous permettez ainsi au serveur ObjectServer de traiter des alertes sans qu'un opérateur n'aie besoin d'intervenir. Vous pouvez également utiliser l'automatisation pour gérer le dédoublonnage, qui permet de réduire la quantité de données traitées dans le serveur ObjectServer en éliminant les événements en double.

E-AUT-003-004 Trigger *%0s*\: recursive raise detected. Aborting execution

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
2012-04-13T21:12:28 : Erreur : E-AUT-003-004 :
 Trigger "test_correlation": recursive raise
 detected. Aborting execution
```

Cette erreur indique que le déclencheur concerné exécute une action qui peut entraîner l'exécution récursive du déclencheur, détériorant ainsi les performances de l'ObjectServer. Lorsque ce dernier détecte qu'il appelle un déclencheur de manière récursive, il empêche l'exécution du déclencheur et renvoie ce message d'erreur.

Dans l'exemple de message, un déclencheur de base de données personnalisée, `test_correlation`, est déclenché lors d'une mise à jour de la table alerts.status. Cependant, le déclencheur proprement dit applique une mise à jour à alerts.status, ce qui démarre une boucle récursive.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : La réponse à cette erreur est fonction de votre configuration système. Pour résoudre le problème indiqué dans l'exemple, modifiez le déclencheur afin d'empêcher la récursivité ou utilisez un déclencheur temporel à la place d'un déclencheur de base de données.

E-AUT-003-011 *%0s* raising signal *%1s*. Counterpart ObjectServer is not available for failover or failback.

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-AUT-003-011 : signal émis par la passerelle
gw_counterpart_down. L'ObjectServer de
contrepartie n'est pas disponible pour la
reprise en ligne ou la reprise par
restauration
```

L'exemple de message a été écrit dans un fichier journal ObjectServer. Il indique qu'une passerelle ObjectServer bidirectionnelle ne peut pas se connecter à un ObjectServer dans une paire de reprise en ligne ou de reprise par restauration. La passerelle émet le signal `gw_counterpart_down` dans l'ObjectServer auquel il est encore connecté.

Le signal `gw_counterpart_down` peut être configuré pour entraîner l'ObjectServer restant à reprendre les fonctions de l'ObjectServer principal.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Déterminez la cause de l'indisponibilité de l'ObjectServer. Pour plus d'informations, examinez les messages qui l'accompagnent dans le journal. Si l'ObjectServer qui a consigné le message était l'ObjectServer de secours, vérifiez qu'il traite à présent les alarmes.

ETC

Messages relatifs à la bibliothèque Netcool/OMNIbus centrale `libnetcool`.

E-ETC-102-036 Fail to parse date string: *%0s* **using format:** *%1s*. **Error code:** *%2s*

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
Erreur : E-ETC-102-036 :
Fail to parse date string:
 Tue DEC 19 18:33:11 JST 2010
 using format: %a %b %e %T %Y.
 Error code: U_PARSE_ERROR
```

Ce message d'erreur indique qu'une sonde n'a pas réussi à analyser une chaîne de date.

L'exemple de message d'erreur a été consigné dans un fichier journal de sonde. Dans cet exemple, l'erreur se produit car le modèle de date et d'heure `%a %b %e %T %Y` ne correspond pas à la chaîne de date, qui inclut les informations de fuseau horaire (JST).

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre le problème indiqué par l'exemple de message d'erreur, ajoutez les informations de fuseau horaire au modèle. Par exemple, utilisez le modèle `%a %b %e %T %Z %Y` pour définir la chaîne d'heure `Tue DEC 19 18:33:11 JST 2010`.

Cependant, étant donné que JST n'est pas un identificateur de fuseau horaire standard pris en charge par ICU, la chaîne d'heure `Tue DEC 19 18:33:11 JST 2010` continue de générer une erreur d'analyse. Pour résoudre ce problème, utilisez un identificateur de fuseau horaire sans équivoque égal à JST, par exemple `Tue DEC 19 18:33:11 GMT+9 2010`.

Cette erreur ne se produit pas dans les versions de Tivoli Netcool/OMNIbus antérieures à la V7.3 car ces versions ignorent les informations de fuseau horaire. Dans Tivoli Netcool/OMNIbus version 7.3, les modèles de date et d'heure POSIX sont obsolètes en faveur des modèles de date et d'heure LDML (Locale Data Markup Language) qui sont fournis par les bibliothèques ICU (International Components for Unicode). Dans la mesure du possible, remplacez les modèles POSIX dans les fichiers de règles de sonde par des modèles LDML.

Pour plus d'informations sur le remplacement des modèles POSIX par des formats de date et d'heure LDML, voir *IBM Tivoli Netcool/OMNIbus - Guide d'installation et de déploiement*.

Pour plus d'informations sur l'utilisation des fonctions de date et d'heure dans les fichiers de règles, voir *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

W-ETC-004-066 Failed to load properties file *%0s*: *%1s*

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal :

```
W-ETC-004-066: Failed to load properties file
 /opt/netcool/omnibus/etc/NCOMS.props:
 No such file or directory
W-ETC-004-066: Failed to load properties file
 /nco731/tivoli/netcool/omnibus/etc/NCOMS.props:
 A file or directory in the path name
 does not exist.
```

Les exemples de messages indiquent que le fichier de propriétés ObjectServer n'existe pas ou qu'une partie de son chemin d'accès est incorrecte.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre le problème indiqué par l'exemple de message, vérifiez que le fichier de propriétés ObjectServer existe. L'emplacement par défaut est `$NCHOME/omnibus/etc`.

Vous pouvez utiliser l'option ObjectServer `-propsfile` pour définir le fichier de propriétés :

```
nco_objserv -propsfile $NCHOME/omnibus/etc
/NCOMS.props
```

Cette erreur peut se produire lorsque vous tentez de démarrer un serveur ObjectServer sans d'abord exécuter l'utilitaire d'initialisation de la base de données (**nco_dbinit**) pour créer les fichiers base de données et le fichier de propriétés par défaut.

Pour plus d'informations sur la création et l'exécution de serveurs ObjectServer, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

GOB

Messages relatifs aux passerelles ObjectServer.

E-GOB-102-041 [ngobjserv]: *%0s*: **Reader has failed to connect to the ObjectServer** *%1s* **. (***%2d***:***%3s***)**

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal.

Exemple :

```
E-GOB-102-041: [ngobjserv]: RdrWtr : le lecteur
n'a pas pu se connecter à l'ObjectServer
'DEVPNCOMS'.
(-19:General failure)
```

Exemple :

```
E-GOB-102-041: [ngobjserv]: ObjectServerA :
le lecteur n'a pas pu se connecter à
l'ObjectServer 'NCOMS_TMPR'.
(-27:Connection closed)
```

Exemple :

```
E-GOB-102-041: [ngobjserv]: ObjectServerB :
le lecteur n'a pas pu se connecter à
l'ObjectServer 'NCOMSBAK'.
(-2302:Session connection failure)
```

Exemple :

```
E-GOB-102-041: [ngobjserv]: ObjectServerB :
le lecteur n'a pas pu se connecter à
l'ObjectServer 'NGSVOIP_OBJ1'.
(1000008 : nom de noeud ou nom de
service inconnu)
```

Les exemples de message ont été écrits sur la passerelle ou des fichiers journaux de la passerelle de l'ObjectServer. Les messages indiquent qu'un lecteur de passerelle n'a pas pu se connecter à un ObjectServer nommé.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Le code d'erreur (*%2d*:*%3s*) indique la cause du problème. Pour plus d'informations, examinez les messages qui l'accompagnent dans le journal.

E-GOB-102-057 [ngobjserv]: Failed to send the command to the ObjectServer. (*%0d***:***%1s***)**

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
Error: E-GOB-102-057: [ngobjserv]:
 Failed to send the command to the ObjectServer.
 (-19:General failure)
```

L'exemple de message d'erreur a été consigné dans un fichier journal de passerelle ObjectServer bidirectionnelle et indique que la passerelle ne peut pas envoyer une commande au serveur ObjectServer.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Cette erreur peut être provoquée par différentes conditions. Vérifiez que le serveur ObjectServer s'exécute correctement et qu'il gère la charge en cours.

Pour surveiller les performances du serveur ObjectServer, assurez-vous que sa propriété **Profile** est activée et vérifiez les journaux de statistiques de profil et de déclencheur dans les fichiers `$NCHOME/omnibus/logs/`*`nom_ObjectServer`*`_profiler_report.log1` et `$NCHOME/omnibus/logs/`*`nom_ObjectServer`*`_trigger_stats.log1`.

Assurez-vous que le temps total des automatisations et

demandes client dans les journaux de statistiques de profil et de déclencheur est inférieur à la valeur de la propriété **Granularity** de l'ObjectServer (valeur par défaut : 60 secondes).

Si le temps total est régulièrement supérieur à la valeur de la propriété **Granularity**, les performances du serveur ObjectServer peuvent se détériorer. Déterminez les incidents liés aux clients et déclencheurs identifiés dans les fichiers journaux comme consommant la plupart des cycles IDUC. Le temps client peut être supérieur à la période de granularité en raison du traitement multitâche du serveur ObjectServer.

Pour plus d'informations sur la surveillance des performances du serveur ObjectServer, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Référence associée:

«Indicateurs clés de performances du serveur ObjectServer», à la page 310
Surveillez ces indicateurs clés de performances (KPI) pour déterminer l'efficacité du serveur ObjectServer.

«Exécution du serveur ObjectServer avec le profilage activé», à la page 316
Utilisez le profilage pour mesurer la durée de l'exécution des requêtes SQL sur le serveur ObjectServer et identifier les connexions client qui utilisent des ressources excessives.

E-GOB-102-061 [ngobjserv]: Failed to connect to the server *%0s* **. (***%1d***:***%2s***)**

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-GOB-102-061: [ngobjserv] : Echec de la connexion
 au serveur 'AGG_B'. (-2302:Session connection
 failure)
```

L'exemple de message indique qu'une passerelle de l'ObjectServer n'a pas pu se connecter à l'ObjectServer AGG_B.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Cette erreur peut être provoquée par différentes conditions. La chaîne d'erreur indique la cause de l'erreur.

Les actions suivantes vous aident à déterminer la cause de l'erreur.

- Examinez le fichier journal pour consulter les messages associés qui peuvent indiquer la cause première de l'erreur.
- Vérifiez que l'ObjectServer indiqué s'exécute correctement. Vous pouvez utiliser l'interface graphique d'administrateur de l'Tivoli Netcool/OMNIbus (**nco_config**) pour vérifier l'état de l'ObjectServer. Vous pouvez également utiliser l'interface interactive SQL pour vérifier la disponibilité de l'ObjectServer ou l'utilitaire **nco_ping** pour vérifier sa disponibilité et le temps de réponse.
- Si l'ObjectServer ne répond pas, vérifiez qu'il est correctement configuré dans l'éditeur de serveur (ou le fichier de données de connexions $NCHOME/etc/omni.dat).
- Si l'ObjectServer est occupé, consultez le profil suivant et déclenchez les fichiers journaux de statistiques pour déterminer s'il existe des problèmes de performances :
  - $NCHOME/omnibus/logs/*nom_ObjectServer*_profiler_report.log1
  - $NCHOME/omnibus/logs/*nom_ObjectServer*_trigger_stats.log1

  Vous devez activer la propriété **Profile** de l'ObjectServer pour générer ces fichiers journaux.
- Si l'ObjectServer est correctement configuré et s'exécute correctement, l'erreur peut être causée par une situation de dépassement du délai d'attente. Dans ce cas, la valeur de la propriété **Ipc.Timeout** dans le fichier de propriétés de la passerelle de l'ObjectServer est peut-être trop faible.

Pour plus d'informations sur la surveillance des performances du serveur ObjectServer, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Pour des informations sur les propriétés de la passerelle de l'ObjectServer, voir *IBM Tivoli Netcool/OMNIbus ObjectServer Gateway Reference Guide*.

Référence associée:

«Indicateurs clés de performances du serveur ObjectServer», à la page 310
Surveillez ces indicateurs clés de performances (KPI) pour déterminer l'efficacité du serveur ObjectServer.

«Exécution du serveur ObjectServer avec le profilage activé», à la page 316
Utilisez le profilage pour mesurer la durée de l'exécution des requêtes SQL sur le serveur ObjectServer et identifier les connexions client qui utilisent des ressources excessives.

E-GOB-102-128 [ngobjserv]: *%0s***: Failed to send the constructed SQL command to the server. (***%1d***:***%2s***)**

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal :

Exemple 1 :

```
E-GOB-102-128: [ngobjserv] : ObjectServerA :
Echec de l'envoi de la commande SQL
construite au serveur. (-19:General failure)
```

Exemple 2 :

```
E-GOB-102-128: [ngobjserv]:
Programme d'écriture : Echec de l'envoi
de la commande SQL construite au serveur.
(-27:Connection closed)
```

Les exemples de message indiquent qu'une passerelle ObjectServer n'a pas pu envoyer une commande SQL à un ObjectServer.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Cette erreur peut être provoquée par différentes conditions. La chaîne d'erreur indique la cause de l'erreur.

La cause la plus probable de ces exemples de message d'erreur est un problème lié à la connexion entre la passerelle et l'ObjectServer. L'ObjectServer n'est pas en cours d'exécution ou il existe des problèmes sur le réseau. Examinez le fichier journal de l'ObjectServer pour consulter les messages associés.

GTK

Messages relatifs au kit d'outils NGtk de la passerelle.

E-GTK-102-073 [ngtk]: *%0s*: **Reader connection handler function has failed to complete successfully. (***%1d***:***%2s***)**

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal.

Exemple 1 :

```
E-GTK-102-073: [ngtk]: ObjectServerB : la fonction
de gestionnaire de connexion au lecteur n'a pas pu
se terminer correctement.
(-22:Object not found)
```

Exemple 2 :

```
E-GTK-102-073: [ngtk]: ObjectServerA : la fonction
de gestionnaire de connexion au lecteur n'a pas pu
se terminer correctement.
(-27:Connection closed)
```

Exemple 3 :

```
E-GTK-102-073: [ngtk]: ObjectServerB : la fonction
de gestionnaire de connexion au lecteur n'a pas pu
se terminer correctement.
(-2302:Session connection failure)
```

Les exemples de message ont été écrits dans les fichiers journaux de la passerelle de l'ObjectServer. Les messages indiquent qu'un lecteur de passerelle n'a pas pu se connecter à un ObjectServer.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Le code d'erreur (*%1d*:*%2s*) indique la cause du problème. Pour plus d'informations, examinez les messages qui l'accompagnent dans le journal.

E-GTK-102-113 [ngtk]: *%0s*: Reader component has failed to connect to the source system. (*%1d*:*%2s*)

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal.

Exemple 1 :

```
E-GTK-102-113: [ngtk]: G_JDBC : Le composant
lecteur n'a pas pu se connecter au système
source (-19:General failure)
```

Exemple 2 :

```
E-GTK-102-113: [ngtk]: ObjectServerA: Le
composant lecteur n'a pas pu se connecter
au système source.
(-22:Object not found)
```

Exemple 3 :

```
E-GTK-102-113: [ngtk]: ObjectServerA: Le
composant lecteur n'a pas pu se connecter
au système source.
(-27:Connection closed)
```

Exemple 4 :

```
E-GTK-102-113: [ngtk]: ObjectServerB : Le
composant lecteur n'a pas pu se connecter
au système source.
(-2302:Session connection failure)
```

Les exemples de message ont été écrits sur la passerelle ou des fichiers journaux de la passerelle de l'ObjectServer. Les messages indiquent qu'un composant lecteur de passerelle n'a pas pu se connecter à une application source ou une base de données.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Le code d'erreur (*%1d*:*%2s*) indique la cause du problème. Pour plus d'informations, examinez les messages qui l'accompagnent dans le journal.

IVM

Messages relatifs à la communication interprocessus (IPC).

E-IVM-005-001 OpenServer - Error: *%0d*:*%1s*

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-IVM-005-001: OpenServer - Error: 16133/10/0:
 Configuration of 100 connections has been
 exceeded, connection rejected
```

L'exemple de message indique qu'un seuil de connexions a été dépassé. Le message E-IVM-005-001 peut indiquer n'importe laquelle des différentes erreurs et n'est pas limité aux problèmes de connexion.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Les erreurs OpenServer peuvent avoir de nombreuses causes. Pour plus d'informations, examinez le texte du message et les messages qui l'accompagnent dans le journal.

E-IVM-005-002 OpenServer - Error: *%0d*:*%1s*

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal.

Exemple 1 :

```
E-IVM-005-002: OpenServer - Error: 16240/10/0:
 Net-Library routine sybnet_listen() failed in
 srv__start_net
 Network error: status = 22 - Protocol driver call
 to determine properties failed
 Error -1:SSL: Unable to open keyfile. Either the
 file is missing, or the file permissions do not
 grant access, or the file format is incorrect.
```

Exemple 2 :

```
E-IVM-005-002: OpenServer - Error: 16240/10/0:
 Net-Library routine sybnet_listen() failed in
 srv__start_net
 Network error: status = 22 - Protocol driver call
 to determine properties failed
 Error -1:SSL: Password incorrect. Possibly the
 stash-file is missing or the password does not
 meet security requirements.
```

Exemple 3 :

```
E-IVM-005-002: OpenServer - Error: 16240/10/0:
 Net-Library routine  failed in
 srv__io_check
 Network error: status = 32 - Net-Lib protocol
 driver call to read data failed
 Error -1:SSL: The label for this server cannot
 be found in the key file.
```

Exemple 4 :

```
E-IVM-005-002: OpenServer - Error: 16240/10/0:
 Net-Library routine sybnet_listen() failed in
 srv__start_net
 Network error: status = 23 - Net-Lib protocol
 driver call to register a listener failed
 Error -1:Socket bind failed - errno 13
 Permission denied
```

Exemple 5 :

```
E-IVM-005-002: OpenServer - Error: 16240/10/0:
 Net-Library routine sybnet_listen() failed in
 srv__start_net
```

```
Network error: status = 23 - Net-Lib protocol
driver call to register a listener failed
Error -1:Socket bind failed - errno 125
Address already in use
```

Exemple 6 :

```
E-IVM-005-002: OpenServer - Error: 16240/10/0:
Net-Library routine sybnet_listen() failed in
srv__start_net
Network error: status = 23 - Net-Lib protocol
driver call to register a listener failed
Error -1:Socket bind failed - errno 126
Cannot assign requested address
```

Le message `E-IVM-005-002` indique que le serveur est confronté à un problème de communication.

Les exemples de messages répertoriés ici indiquent qu'il existe des paramètres incorrects dans l'éditeur de serveur. Sur les systèmes d'exploitation UNIX et Linux, ces paramètres sont également spécifiés dans le fichier de données de connexion, `$NCHOME/etc/omni.dat`.

Les exemples de messages 1, 2, 4, 5 et 6 sont généralement accompagné dans le journal par le message fatal suivant :

```
F-IVM-005-001:
OpenServer - Fatal Error:
16029/20/0: Failed to start any network
listeners
```

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Les erreurs OpenServer peuvent avoir de nombreuses causes. Pour plus d'informations, examinez le texte du message et les messages qui l'accompagnent dans le journal.

Les exemples de messages décrits ici peuvent être provoqués par les conditions suivantes :

- Dans l'exemple de message 1, le chiffrement SSL est spécifié dans l'éditeur de serveur, mais le fichier de clés ou le fichier de dissimulation est manquant.
- Dans l'exemple de message 2, le chiffrement SSL est spécifié dans l'éditeur de serveur et le fichier de clés est présent, mais le mot de passe stocké dans le fichier de dissimulation est incorrect ou ne répond pas aux exigences de sécurité. Dans certains cas, le fichier de dissimulation peut être manquant.
- Dans l'exemple de message 3, le chiffrement SSL est spécifié dans l'éditeur de serveur et le fichier de clés est présent, mais le certificat du serveur est manquant. Cette erreur n'est pas fatale. Le serveur continue de s'exécuter mais il ne peut pas se connecter au serveur ObjectServer.
- Dans l'exemple de message 4, le numéro de port de l'hôte spécifié dans l'éditeur de serveur est incorrect ou manquant.
- Dans l'exemple de message 5, le numéro de port de l'hôte spécifié dans l'éditeur de serveur est déjà utilisé.
- Dans l'exemple de message 6, le nom d'hôte spécifié dans l'éditeur de serveur est incorrect

Pour plus d'informations sur la configuration des informations de communication ou des connexions SSL du serveur, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Information associée:

«F-IVM-005-001», à la page 475
OpenServer - Fatal Error: *%0d*:*%1s*

E-IVM-053-005 OpenClient (Couche = *%0s*, **Origine =** *%1s*, **Gravité =** *%2s*, **Numéro =** *%3s*) **- Session '***%4s***' :** *%5s*

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal.

Exemple 1 :

```
E-IVM-053-005: OpenClient (Couche = 5, Origine = 3,
Gravité = 5, Nombre = 6) - Session 'DEVPNCOMS' :
incident de lecture anticipée : couche de paquet
rêseau : erreur de bibliothèque de rêseau
interne : opération de Net-Library arrêtée en
raison de déconnexion
```

Exemple 2 :

```
E-IVM-053-005: OpenClient (Couche = 6, Origine = 8,
Gravité = 5, Nombre = 3) - Session 'INPOSSSP' :
ct_connect() : couche de service d'annuaire :
erreur de couche de contrôle de répertoire
interne : Nom de serveur demandé non trouvé.
```

Exemple 3 :

```
E-IVM-053-005: OpenClient (Couche = 1, Origine = 2,
Gravité = 2, Numéro = 63) - Session 'AGG_V':
ct_results() : couche d'api utilisateur :
erreur de bibliothèque client interne :
Expiration du délai de la lecture sur le serveur.
```

Exemple 4 :

```
E-IVM-053-005: OpenClient (Couche = 1, Origine = 1,
Gravité = 1, Numéro = 50) - Session 'AGG_V':
ct_cmd_drop() : couche d'api utilisateur :
erreur externe : La connexion a été marquée
comme perdue.
```

Les exemples de message indiquent qu'un client ObjectServer a rencontré un problème de communication.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Les erreurs OpenClient peuvent avoir plusieurs causes. Pour plus d'informations, examinez le texte du message et les messages qui l'accompagnent dans le journal.

E-IVM-053-005

Les exemples de message 1 et 4 indiquent que la connexion client à l'ObjectServer a été perdue. Vérifiez les journaux de l'ObjectServer pour plus d'informations sur les raisons de la déconnexion du client.

L'exemple de message 2 indique que le nom de l'ObjectServer est introuvable dans les fichiers d'interfaces. Configurez les informations de connexion de l'ObjectServer dans l'éditeur de serveur (ou dans le fichier de données de connexions `$NCHOME/etc/omni.dat`).

L'exemple 3 indique une expiration du délai de la lecture sur l'ObjectServer. Si le client utilise une propriété d'expiration du délai (telle que **NetworkTimeout**), vérifiez que cette valeur est suffisamment élevée pour permettre les périodes d'occupation de l'ObjectServer. Si le client est une sonde connectée à une paire d'ObjectServers dans une configuration de reprise en ligne, vérifiez que la valeur de la propriété **PollServer** de la sonde est supérieure à la valeur de la propriété **NetworkTimeout**.

Pour plus d'informations sur la configuration des informations de communication du serveur, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Pour de plus amples informations sur la configuration des propriétés de sonde, voir le manuel *Guide des sondes et des passerelles d'IBM Tivoli Netcool/OMNIbus*.

F-IVM-005-001 OpenServer - Fatal Error: *%0d*:*%1s*

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal.

Exemple 1 :

```
F-IVM-005-001: OpenServer - Fatal Error:
16029/20/0: Failed to start any network
listeners
```

Exemple 2 :

```
F-IVM-005-001: OpenServer -
Fatal Error:
16013/20/0: Could not find server name
'NCOMS' in interfaces file
```

L'exemple de message indique qu'il existe un problème lié aux paramètres dans l'éditeur de serveur. Sur les systèmes d'exploitation UNIX et Linux, ces paramètres sont également spécifiés dans le fichier de données de connexion, `$NCHOME/etc/omni.dat`.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Les erreurs OpenServer, comme celle de l'exemple de message 1, peuvent avoir de nombreuses causes. Pour plus d'informations, examinez le texte du message et les messages qui l'accompagnent dans le journal.

Dans l'exemple de message 2, il existe une entrée dans l'éditeur de serveur pour le serveur appelé `NCOMS`.

Pour plus d'informations sur la configuration des informations de communication du serveur, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

OBX

Messages relatifs à l'application ObjectServer (**nco_objserv**).

E-OBX-102-017 Language command from *%0s*@*%1s* **failed.** *%2s* **on line** *%3u* **of statement** *%4s*

Explication : Les exemples suivants montrent comment ce message est consigné dans le journal.

Exemple 1 :

```
E-OBX-102-017 : la commande de langage de
probe@lcdsd928 a échoué. Tentative
d'insérer une ligne en double dans la
ligne 1 de l'instruction
'exemple_instruction_SQL'
```

Exemple 2 :

```
E-OBX-102-017 : la commande de langage de
root@yfcdatv1in a échoué. Le groupe est
déjàaffecté à un utilisateur ou à un
rôle dans la ligne 1 de l'instruction
'exemple_instruction_SQL'
```

Exemple 3 :

```
E-OBX-102-017 : la commande de langage de
root@hcwweb01 a échoué. Espace insuffisant
dans la ligne 1 de l'instruction
'exemple_instruction_SQL'
```

Exemple 4 :

```
E-OBX-102-017 : la commande de langage de
gateway@tmrt02 a échoué. Actuellement, la
mémoire disponible est insuffisante dans
la ligne 1 de l'instruction
'exemple_instruction_SQL'
```

Exemple 5 :

```
E-OBX-102-017 : la commande de langage de
probe@SVHJ0361 a échoué. La limite souple
de la région est dépassée dans la ligne 1
de l'instruction
'exemple_instruction_SQL'
```

Exemple 6 :

```
E-OBX-102-017 : la commande de langage de
gateway@y03100u0010 a échoué. Tentative de
modifier une ligne qui a été supprimée
dans la ligne 2 de l'instruction
'exemple_instruction_SQL'
```

Les exemples de message ont été écrits dans les fichiers journaux de l'ObjectServer et indiquent qu'une instruction SQL n'a pas été traitée. Les instructions SQL sont omises des exemples.

Dans l'exemple de message 1, la cause du problème est une tentative d'insertion d'une ligne en double. Dans l'exemple de message 2, la cause du problème est liée aux droits d'accès. Dans les exemples de message 3, 4 et 5, le problème est causé par une mémoire insuffisante. Dans l'exemple de message 6, la ligne n'existe plus dans l'ObjectServer, elle ne peut donc pas être modifiée.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Le code d'erreur (*%2s*) indique la cause du problème. Lorsque cela est indiqué, vérifiez les problèmes de ressources insuffisantes ou de droits d'accès. Vérifiez les éventuels problèmes de syntaxe SQL dans l'instruction SQL.

Dans l'exemple de message 1, un déclencheur de dédoublonnage n'existe pas pour la table dans laquelle l'événement est inséré. Dans ce cas, vous pouvez ajouter un déclencheur de dédoublonnage pour traiter les lignes en double dans la table concernée.

L'exemple de déclencheur suivant supprime les données d'événement en double entrantes et n'effectue pas de mises à jour :

```
create or replace trigger custom_dedup
 group default_triggers
 debug false
 enabled true
 priority 1
 comment 'Deduplication processing for
 custom.status'
 before reinsert on custom.status
 for each row
 begin

 cancel;

end;
```

Dans l'exemple de message 5, le message `Limite souple de la région dépassée` indique que la taille de base de données de l'ObjectServer maximale est dépassée. Vous devrez peut-être augmenter la limite souple de table_store.

Pour plus d'informations sur la modification des limites flexible et absolue de table_store, voir *IBM Tivoli Netcool/OMNIbus - Guide d'administration*.

Référence associée:

«Modification des limites logicielle et matérielle du magasin de mémoire table_store», à la page 34
Si la base de données d'alertes prend des proportions exceptionnellement importantes, vous pouvez utiliser une commande afin de changer les limites souples et strictes.

E-OBX-102-018 Failed to resolve parsed language command. (*%0d*:*%1s*)

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-OBX-102-018: Failed to resolve parsed language
 command. (-636:Type mismatch for column)
```

Cet exemple de message indique que le serveur ObjectServer n'a pas pu résoudre une commande SQL. Dans ce cas, le type de colonne dans une commande `INSERT` ne correspond pas au type de colonne de la table ObjectServer.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre le problème indiqué par l'exemple de message, vérifiez que le type de données source correspond au type de données de la table ObjectServer. Lorsque vous utilisez des commandes `INSERT`, placez les valeurs de type caractère entre des apostrophes (’). Ne placez pas les valeurs de type entier entre guillemets.

Des erreurs de non-concordance de type peuvent également se produire lorsque vous utilisez la commande `INSERT` dans une automatisation pour insérer des lignes dans la table ObjectServer alerts.journal. Dans ce cas, utilisez la procédure d'automatisation standard `jinsert` pour insérer des lignes dans la table alerts.journal.

Pour plus d'informations sur la commande `INSERT` et la procédure `jinsert`, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Référence associée:

«INSERT», à la page 397
La commande `INSERT` permet d'insérer une nouvelle ligne de données dans une table existante.

«Exemples d'automatisation», à la page 203
Les exemples suivants indiquent comment créer des déclencheurs et des procédures pour certaines automatisations fréquemment exécutées.

E-OBX-102-019 Failed to execute parsed language command. (*%0d*:*%1s*)

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
Erreur : E-OBX-102-019 : Failed to execute
parsed language command.
(-402:Region soft limit exceeded)
```

L'exemple de message indique que le magasin de mémoire table_store a dépassé sa limite flexible.

Le message `E-OBX-102-019` peut indiquer n'importe laquelle des différentes erreurs et n'est pas limité aux problèmes liés au magasin de mémoire. La variable de message (*%0d*:*%1s*) contient les principales informations relatives au problème.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre le problème indiqué par l'exemple de message d'erreur, examinez les journaux statiques du profileur et du déclencheur ObjectServer pour déterminer si un problème lié à la taille de l'une des tables dynamiques a lieu. Si le problème est provoqué par un manque d'espace de table_store et que la taille des tables dynamiques ObjectServer est dans des limites normales, augmentez les limites flexible et absolue de table_store.

Pour plus d'informations sur la modification des limites flexible et absolue de table_store, voir *IBM Tivoli Netcool/OMNIbus - Guide d'administration*.

Référence associée:

«Modification des limites logicielle et matérielle du magasin de mémoire table_store», à la page 34
Si la base de données d'alertes prend des proportions exceptionnellement importantes, vous pouvez utiliser une commande afin de changer les limites souples et strictes.

E-OBX-102-020 Failed to start the storage system. (*%0d*:*%1s*)

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
Erreur : E-OBX-102-020 : Failed to start the
storage system. (-490:Extent statistics mismatch)
```

Ce message d'erreur indique que le démarrage du serveur ObjectServer a échoué pour l'une des nombreuses raisons.

Dans l'exemple de message d'erreur, `Extent statistics mismatch` indique que la base de données du serveur ObjectServer est altérée. La base de données peut s'endommager lorsqu'un serveur ObjectServer est arrêté de façon anormale, par exemple en raison d'une panne système ou si le processus `nco_objserv` est arrêté.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre le problème indiqué par l'exemple de message d'erreur, restaurez le dernier serveur ObjectServer non altéré.

Remarque : Les procédures de restauration décrites ci-dessous donnent lieu à une perte de données. Les données sont restaurées à l'état dans lequel était le serveur ObjectServer lors de la création des fichiers. Si vous disposez d'une sauvegarde récente du serveur ObjectServer (faite par exemple à l'aide de la commande `ALTER SYSTEM BACKUP`), procédez comme suit :

1. Effectuez une sauvegarde du répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`*.
2. Supprimez tous les fichiers du répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`* en cours.
3. Copiez les fichiers `master_store.tab` et `table_store.tab` du répertoire de sauvegarde vers le répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`* `en cours`.
4. Démarrez le serveur ObjectServer.

Si vous ne disposez pas de sauvegarde récente du serveur ObjectServer, vous pouvez essayer de restaurer l'ObjectServer à l'aide des fichiers de base de données en cours :

1. Effectuez une sauvegarde du répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`*.
2. Dans le répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`*, identifiez tous les fichiers comportant les suffixes `_0` et `_1` et dont la taille est égale à 0 octet, puis supprimez les fichiers `.chk` et `.log` correspondants.

   Par exemple, dans le répertoire de base de données suivant, les fichiers dont le suffixe et `_1` ont une taille de 0 octet ; supprimez `master_store_1.chk`, `master_store_1.log`, `table_store_1.chk` et `table_store_1.log` :

   ```
   root 126340 Dec 7 08:47 master_store_0.chk
   root 126340 Dec 7 08:47 master_store_0.log
   root 0 Dec 7 08:47 master_store_1.chk
   root 0 Dec 7 08:47 master_store_1.log
   root 2822144 Dec 7 08:45 master_store.tab
   root 7656 Dec 7 08:47 table_store_0.chk
   root 7656 Dec 7 08:47 table_store_0.log
   root 0 Dec 7 08:47 table_store_1.chk
   root 0 Dec 7 08:47 table_store_1.log
   root 3543040 Dec 7 08:45 table_store.tab
   ```
3. Démarrez le serveur ObjectServer.
4. Si le démarrage du serveur ObjectServer échoue, procédez comme suit :
   a. Restaurez le répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`* que vous avez sauvegardé à l'étape 1.
   b. Dans le répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`* restauré, supprimez les fichiers `.chk` et `.log` que vous n'avez pas supprimés à l'étape 2.

      Par exemple, dans le répertoire de base de données de l'étape 2, vous devez supprimer `master_store_0.chk`, `master_store_0.log`, `table_store_0.chk` et `table_store_0.log`.
   c. Démarrez le serveur ObjectServer.
5. Si le serveur ObjectServer ne démarre toujours pas, procédez comme suit :
   a. Supprimez tous les fichiers `.chk` et `.log` du répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`*.

      Ne supprimez pas les fichiers `table_store.tab` et `master_store.tab`.
   b. Démarrez le serveur ObjectServer.

Pour plus d'informations sur les points de contrôle et les sauvegardes d'ObjectServer, voir *IBM Tivoli Netcool/OMNIbus - Guide d'administration*.

Concepts associés:

«Introduction à l'utilisation des points de contrôle», à la page 31
Le serveur ObjectServer prend en charge la persistance des données à l'aide de points de contrôle et de journaux, pour copier les données en mémoire sur disque. Cela permet de récupérer les données après un arrêt planifié ou non planifié.

Référence associée:

«ALTER SYSTEM», à la page 356
La commande `ALTER SYSTEM` permet de modifier les paramètres par défaut et actuels du serveur ObjectServer en paramétrant les propriétés, d'arrêter le serveur ObjectServer, de supprimer des connexions utilisateur ou de sauvegarder le serveur ObjectServer.

E-OBX-102-068 Failed to send wait message to client blocked meta/data wait. (*%0d*:*%1s*)

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-OBX-102-068: Failed to send wait message
 to client blocked meta/data wait.
 (-2250:Failed to send message to client)
```

Cet exemple de message indique que le serveur ObjectServer est occupé et n'a pas pu envoyer un message d'attente au client qui a essayé de se connecter au serveur.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : L'exemple de message indique qu'un problème lié aux performances a lieu au niveau du serveur ObjectServer. Cette erreur se produit généralement lorsqu'il existe un problème lié à l'utilisation de la mémoire par les tables dynamiques ou lié à un ou plusieurs déclencheurs.

L'erreur peut également survenir lorsqu'une passerelle se resynchronise avec le serveur ObjectServer. Le serveur ObjectServer peut être verrouillé lors de la resynchronisation et le message est consigné jusqu'à ce que la resynchronisation soit terminée.

Pour plus d'informations sur la surveillance des performances ObjectServer, les déclencheurs et les automatisations, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Concepts associés:

«Déclencheurs», à la page 176
Vous pouvez utiliser les déclencheurs pour détecter les modifications du serveur ObjectServer et exécuter des réponses automatiques à ces modifications. Vous permettez ainsi au serveur ObjectServer de traiter des alertes sans qu'un opérateur n'aie besoin d'intervenir. Vous pouvez également utiliser l'automatisation pour gérer le dédoublonnage, qui permet de réduire la quantité de données traitées dans le serveur ObjectServer en éliminant les événements en double.

Référence associée:

«Indicateurs clés de performances du serveur ObjectServer», à la page 310
Surveillez ces indicateurs clés de performances (KPI) pour déterminer l'efficacité du serveur ObjectServer.

E-OBX-102-122 Client *%0u* **has missed** *%1d* **IDUC cycles for user** *%2s* **on host** *%3s* **from application** *%4s*:*%5s* **.**

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
10/22/09 21:46:01: Erreur : E-OBX-102-122:
Client '1' has missed '2' IDUC cycles
for user 'Gateway' on host 'hostname'
from application 'GATEWAY'.
```

L'exemple de message indique que le client n'a pas demandé d'informations IDUC à l'ObjectServer dans la période de granularité spécifiée par la propriété **Granularity** de l'ObjectServer. Si ce problème n'est pas corrigé, un client manquant plusieurs cycles IDUC entraîne l'augmentation de l'utilisation de la mémoire de l'ObjectServer et risque d'augmenter les temps de profilage pour certains clients.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Si ce message apparaît régulièrement dans le journal, vous pouvez effectuer une ou plusieurs des actions suivantes pour le résoudre :

- Assurez-vous que la propriété **Profile** de l'ObjectServer est activée et vérifiez les journaux de statistiques de profil et de déclencheur dans les fichiers `$NCHOME/omnibus/logs/`*`nom_ObjectServer`*`_profiler_report.log1` et `$NCHOME/omnibus/logs/`*`nom_ObjectServer`*`_trigger_stats.log1`.

  Assurez-vous que le temps total des automatisations et demandes client dans les journaux de statistiques de profil et de déclencheur est inférieur à la valeur de la propriété **Granularity** de l'ObjectServer (valeur par défaut : 60 secondes).

  Si le temps total est régulièrement supérieur à la valeur de la propriété **Granularity**, les performances de l'ObjectServer se détérioreront. Déterminez les incidents liés aux clients et déclencheurs identifiés dans les fichiers journaux comme consommant la plupart des cycles IDUC. Notez que le temps client peut être supérieur à la période de granularité en raison du traitement multitâche de l'ObjectServer.
- Activez la consignation de débogage pour le client nommé dans le message d'erreur, puis résolvez les éventuelles erreurs. Vérifiez les propriétés client à la recherche d'optimisations de performances disponibles. Si la période IDUC client est inférieure à la granularité de l'ObjectServer, il est peut-être nécessaire de l'augmenter.
- Activez le déclencheur disconnect_iduc_missed. Ce déclencheur déconnecte les clients qui ne parviennent pas à communiquer avec le serveur ObjectServer pendant 100 périodes de granularité, afin d'empêcher celui-ci de consommer trop de mémoire pour prendre en charge les clients qui manquent des cycles IDUC.

  Si le déclencheur déconnecte inutilement des clients, augmentez le paramètre `%signal.missed_cycles` en indiquant au moins 100.

Concepts associés:

«Mises à jour d'outil client via IDUC», à la page 25
Une quantité importante de données est transmise entre le serveur ObjectServer et un client bureau à chaque mise à jour d'une liste d'événements. Pour empêcher toute surcharge du serveur ObjectServer avec les requêtes des mises à jour de liste d'événements, le serveur ObjectServer envoie une invite au client bureau lorsqu'une mise à jour est nécessaire. Le bureau demande ensuite les données mises à jour depuis le serveur ObjectServer.

E-OBX-102-133 Number of failed client heartbeats exceeded *%0u***. The keep-alive message will no longer be sent to client with id =** *%1u* **.**

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-OBX-102-133 : Le nombre de signaux de présence
des clients en échec a dépassé 5. Le message
de signal de présence ne sera plus envoyé au
client avec id = '52'
```

L'exemple de message a été écrit dans un fichier journal ObjectServer et indique qu'un client a cessé de fonctionner.

Ce message est consigné lorsque l'ObjectServer est occupé et que l'un de ses clients en attente se déconnecte. Lorsque l'ObjectServer n'est plus occupé, le message n'est plus consigné car l'ObjectServer a traité la déconnexion du client.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Vérifiez que le serveur ObjectServer s'exécute correctement et qu'il gère la charge en cours. Si ce message revient régulièrement dans les journaux, vous pouvez affiner les performances de l'ObjectServer.

Pour plus d'informations sur le réglage des performances de l'ObjectServer, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Référence associée:

Chapitre 8, «Optimisation des performances», à la page 309
Il est possible de mesurer les performances de Tivoli Netcool/OMNIbus en termes de temps de réponse, de débit et de disponibilité.

PRO

Messages relatifs à l'agent de processus.

E-PRO-004-004 Connection to Process Agent failed. Dropping external action *%0s* **:** *%1s*

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-PRO-004-004 : La connexion à l'agent de
processus a échoué. Suppression de l'action
externe 'idbi_send_mail' :
Echec de la connexion à l'AP
```

L'exemple de message indique que la connexion entre l'ObjectServer et l'agent de processus a échoué. L'échec de la connexion a entraîné l'annulation par l'agent de processus de l'exécution de l'action externe `idbi_send_mail` définie par l'utilisateur.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Examinez le fichier journal de l'agent de processus (`$NCHOME/omnibus/log/`*`nom_ap`*`.log`) et vérifiez que l'agent de processus fonctionne correctement. Le nom de l'agent de processus par défaut est `NCO_PA`.

Si vous exécutez un grand nombre d'actions externes, vérifiez que vous disposez de ressources système suffisantes pour gérer la charge.

REG

Messages relatifs au stockage de base de données ObjectServer.

E-REG-002-025 Region *%0s* **, from directory** *%1s* **, has been recovered in a corrupt state:** *%2s*

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
Erreur : E-REG-002-025 : Region 'table_store',
 from directory '/opt/netcool/omnibus/db/NCOMS/',
 has been recovered in a corrupt state:
 Extent statistics mismatch
```

Ce message d'erreur indique que la base de données du serveur ObjectServer est altérée.

Dans l'exemple de message d'erreur, `Extent statistics mismatch` indique que la base de données du serveur ObjectServer NCOMS est altérée. La base de données peut s'endommager lorsqu'un serveur ObjectServer est arrêté de façon anormale, par exemple en raison d'une panne système ou si le processus `nco_objserv` est arrêté.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre le problème indiqué par l'exemple de message d'erreur, vous devez restaurer le dernier serveur ObjectServer non altéré.

Remarque : Les procédures de restauration décrites ci-dessous donneront lieu à une perte de données. Les données seront restaurées à l'état dans lequel était le serveur ObjectServer lors de la création des fichiers. Si vous disposez d'une sauvegarde récente du serveur ObjectServer (faite par exemple à l'aide de la commande `ALTER SYSTEM BACKUP`), procédez comme suit :

1. Effectuez une sauvegarde du répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`*.
2. Supprimez tous les fichiers du répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`*.
3. Copiez les fichiers `master_store.tab` et `table_store.tab` du répertoire de sauvegarde vers le répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`*.
4. Démarrez le serveur ObjectServer.

Si vous ne disposez pas de sauvegarde récente du serveur ObjectServer, vous pouvez essayer de le restaurer à l'aide des fichiers de base de données en cours :

1. Effectuez une sauvegarde du répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`*.
2. Dans le répertoire `$NCHOME/omnibus/db/`*`nom_ObjectServer`*, identifiez tous les fichiers comportant les suffixes `_0` et `_1` et dont la taille est égale à 0 octet, puis supprimez les fichiers `.chk` et `.log` correspondants.

   Par exemple, dans le répertoire de base de données suivant, les fichiers dont le suffixe et `_1` ont une taille de 0 octet ; vous devez donc supprimer `master_store_1.chk`, `master_store_1.log`, `table_store_1.chk` et `table_store_1.log` :

   ```
   root 126340 Dec 7 08:47 master_store_0.chk
   root 126340 Dec 7 08:47 master_store_0.log
   root 0 Dec 7 08:47 master_store_1.chk
   root 0 Dec 7 08:47 master_store_1.log
   root 2822144 Dec 7 08:45 master_store.tab
   root 7656 Dec 7 08:47 table_store_0.chk
   root 7656 Dec 7 08:47 table_store_0.log
   root 0 Dec 7 08:47 table_store_1.chk
   root 0 Dec 7 08:47 table_store_1.log
   root 3543040 Dec 7 08:45 table_store.tab
   ```
3. Démarrez le serveur ObjectServer.

4. Si le démarrage du serveur ObjectServer échoue, procédez comme suit :
   a. Restaurez le répertoire `$NCHOME/omnibus/db/nom_ObjectServer` que vous avez sauvegardé à l'étape 1.
   b. Dans le répertoire `$NCHOME/omnibus/db/nom_ObjectServer` restauré, supprimez les fichiers `.chk` et `.log` que vous n'avez pas supprimés à l'étape 2.

      Par exemple, dans le répertoire de base de données de l'étape 2, vous devez supprimer `master_store_0.chk`, `master_store_0.log`, `table_store_0.chk` et `table_store_0.log`.
   c. Démarrez le serveur ObjectServer.
5. Si le serveur ObjectServer ne démarre toujours pas, procédez comme suit :
   a. Supprimez tous les fichiers `.chk` et `.log` du répertoire `$NCHOME/omnibus/db/nom_ObjectServer`.

      Ne supprimez pas les fichiers `table_store.tab` et `master_store.tab`.
   b. Démarrez le serveur ObjectServer.

Pour plus d'informations sur les points de contrôle et les sauvegardes d'ObjectServer, voir *IBM Tivoli Netcool/OMNIbus - Guide d'administration*.

Concepts associés:

«Introduction à l'utilisation des points de contrôle», à la page 31
Le serveur ObjectServer prend en charge la persistance des données à l'aide de points de contrôle et de journaux, pour copier les données en mémoire sur disque. Cela permet de récupérer les données après un arrêt planifié ou non planifié.

Référence associée:

«ALTER SYSTEM», à la page 356
La commande `ALTER SYSTEM` permet de modifier les paramètres par défaut et actuels du serveur ObjectServer en paramétrant les propriétés, d'arrêter le serveur ObjectServer, de supprimer des connexions utilisateur ou de sauvegarder le serveur ObjectServer.

E-REG-002-029 Soft limit exceeded for region : *%0s*

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-REG-002-029 : limite souple dépassée pour la
région : nco_g_oracle_cache
```

L'exemple de message a été écrit dans un fichier journal de la passerelle Gateway for Oracle (`nco_g_oracle`) et indique que la mémoire cache est saturée.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Les problèmes de limite souple peuvent être causés par une mémoire insuffisante. Vérifiez que le système dispose de suffisamment de mémoire physique disponible pour les processus obligatoires.

Le problème indiqué par l'exemple de message peut se produire lorsque des événements dans la table d'ObjectServer alerts.status ne sont pas effacés. Il peut également se produire lorsque l'unité d'exécution du lecteur lit plus d'événements de l'ObjectServer qu'elle ne peut en traiter.

Pour résoudre ce problème, réglez la propriété **`Oracle.BatchSize`** (ou la propriété **`ODBC.BatchSize`** dans le cas de la passerelle pour ODBC). Cette propriété spécifie le nombre maximal de lignes transmis à la base de données dans un seul lot.

Pour plus d'informations sur la configuration des passerelles Gateway for Oracle et Gateway for ODBC, voir les guides de référence correspondant à chaque passerelle.

REP

Messages relatifs à l'utilitaire Probe Rules Utility (**`nco_rules_util`**).

E-REP-102-008 Impossible de trouver le fichier ou le répertoire {0} dans le référentiel ({1})

Explication : Un fichier ou un répertoire que vous avez demandé est introuvable sur le système de fichiers.

Réaction du système : Probe Rules Utility n'affiche en sortie aucun détail sur le fichier demandé. Si le fichier est un fichier d'inclusion ou de consultation, aucun total de contrôle pour le fichier de règles faisant référence à ces fichiers n'est affiché en sortie.

Action de l'utilisateur : Consultez la partie *{1}* du message pour obtenir plus d'informations sur la cause du problème. Vérifiez que le fichier *{0}* existe sur le système de fichiers et que votre compte utilisateur y a accès.

Action de l'administrateur : Non applicable.

E-REP-102-009 Pas d'accès en lecture au fichier {0}

Explication : Le système d'exploitation a empêché l'accès au fichier *{0}*.

Réaction du système : Probe Rules Utility n'affiche en sortie aucun détail sur le fichier demandé. Si le fichier est un fichier d'inclusion ou de consultation, aucun total de contrôle pour le fichier de règles faisant référence à ces fichiers n'est affiché en sortie.

Action de l'utilisateur : Vérifiez que votre compte utilisateur a accès au fichier *{0}*.

Action de l'administrateur : Non applicable.

E-REP-102-010 Exception d'E-S {0} lors de la lecture du fichier {1}

Explication : Une erreur inattendue s'est produite lors du traitement d'une demande.

Réaction du système : Non applicable.

Action de l'utilisateur : Contactez votre administrateur système.

Action de l'administrateur : Consultez la partie *{0}* du message pour obtenir plus d'informations sur la cause du problème. Si le problème persiste, contactez votre équipe de support.

E-REP-102-011 La racine de référentiel spécifiée, {0}, n'est pas un répertoire

Explication : Vous avez indiqué une racine de référentiel qui n'est pas un répertoire.

Réaction du système : Probe Rules Utility s'arrête.

Action de l'utilisateur : Indiquez un répertoire pour la racine de référentiel ou n'indiquez pas de racine de référentiel.

Action de l'administrateur : Non applicable.

E-REP-102-012 Exception inattendue {0}

Explication : Une erreur inattendue s'est produite lors du traitement d'une demande.

Réaction du système : Non applicable.

Action de l'utilisateur : Contactez votre administrateur système.

Action de l'administrateur : Consultez la partie *{0}* du message pour obtenir plus d'informations sur la cause du problème. Si le problème persiste, contactez votre équipe de support.

E-REP-102-013 Le fichier de propriétés {0} contient une séquence d'échappement incorrecte en Unicode

Explication : Le fichier de propriétés spécifié contient une séquence d'échappement qui n'est pas valide en Unicode.

Réaction du système : Probe Rules Utility ne démarre pas.

Action de l'utilisateur : Editez le fichier de propriétés *{0}* et supprimez la séquence d'échappement incorrecte.

Action de l'administrateur : Editez le fichier de propriétés *{0}* et supprimez la séquence d'échappement incorrecte.

E-REP-102-014 Impossible de démarrer la consignation dans le fichier {0} ({1})

Explication : Probe Rules Utility n'a pas pu écrire dans le fichier journal *{0}* en raison de l'exception *{1}*.

Réaction du système : Les messages de journal sont copiés dans le flux d'erreur standard de la console.

Action de l'utilisateur : Consultez la partie *{1}* du message pour obtenir plus d'informations sur la cause du problème. Vérifiez que votre compte utilisateur est autorisé à écrire dans l'emplacement de fichier journal *{0}*. S'il existe déjà un fichier journal, vérifiez que votre compte utilisateur est autorisé à remplacer le fichier.

Action de l'administrateur : Consultez la partie *{1}* du message pour obtenir plus d'informations sur la cause du problème. Vérifiez que votre compte utilisateur est autorisé à écrire dans l'emplacement de fichier journal *{0}*. S'il existe déjà un fichier journal, vérifiez que votre compte utilisateur est autorisé à remplacer le fichier.

E-REP-102-015 Taille de fichier journal spécifiée non valide. Par défaut, la taille est illimitée.

Explication : Vous avez spécifié une valeur négative pour la taille de fichier journal maximale.

Réaction du système : Le système ne limite pas la taille du fichier journal.

Action de l'utilisateur : Configurez la taille de fichier journal maximale qui vous convient.

Action de l'administrateur : Configurez la taille de fichier journal maximale qui vous convient.

E-REP-102-016 Nombre maximum de fichiers journaux spécifié non valide. Par défaut, le nombre maximal de fichiers journaux est de 1.

Explication : Le système écrit dans plusieurs fichiers journaux en rotation lorsque la taille de fichier journal maximale est dépassée. Le nombre maximum de fichiers journaux est contrôlé par la propriété **log.maxcount**. Les valeurs admises sont 1 ou plus.

Réaction du système : Les messages de journal sont écrits dans un seul fichier. Ce fichier n'effectue pas de rotation lorsque la taille maximale de fichier journal est dépassée et les messages les plus anciens sont perdus.

Action de l'utilisateur : Donnez à la propriété **log.maxcount** une valeur supérieure ou égale à 1.

Action de l'administrateur : Donnez à la propriété **log.maxcount** une valeur supérieure ou égale à 1.

E-REP-102-018 La propriété {0} est inconnue

Explication : Une propriété inconnue a été spécifiée dans le fichier de propriétés.

Réaction du système : Probe Rules Utility ne démarre pas.

Action de l'utilisateur : Vérifiez que le fichier de propriétés contient uniquement des propriétés prises en charge.

Action de l'administrateur : Vérifiez que le fichier de propriétés contient uniquement des propriétés prises en charge.

E-REP-102-019 Vous devez indiquer une valeur entière pour le paramètre {0}

Explication : Le paramètre ou la propriété de lancement *{0}* requiert une valeur entière. Vous avez indiqué une valeur non entière.

Réaction du système : Ce message est affiché et l'outil de ligne de commande ne démarre pas.

Action de l'utilisateur : Remplacez la valeur *{0}* par une valeur entière.

Action de l'administrateur : Remplacez la valeur *{0}* par une valeur entière.

E-REP-102-020 La valeur {0} n'est pas valide pour la propriété {1}

Explication : La valeur *{0}* n'est pas valide pour la propriété *{1}*.

Réaction du système : Probe Rules Utility ne démarre pas.

Action de l'utilisateur : Vérifiez que le fichier de propriétés contient uniquement des valeurs de propriété prises en charge.

Action de l'administrateur : Vérifiez que le fichier de propriétés contient uniquement des valeurs de propriété prises en charge.

E-REP-102-021 L'option {0} requiert une valeur

Explication : La syntaxe de l'utilitaire Probe Rules Utility est une commande suivie d'options et d'un chemin cible. L'option *{0}* requiert une valeur et aucune valeur n'a été indiquée.

Réaction du système : Probe Rules Utility ne démarre pas.

Action de l'utilisateur : Indiquez une valeur valide pour l'option.

Action de l'administrateur : Indiquez une valeur valide pour l'option.

E-REP-102-022 L'option {0} est inconnue

Explication : La syntaxe de l'utilitaire Probe Rules Utility est une commande suivie d'options et d'un chemin cible. Une option de ligne de commande inconnue (*{0}*) a été indiquée.

Réaction du système : Probe Rules Utility ne démarre pas.

Action de l'utilisateur : Spécifiez une option valide.

Action de l'administrateur : Spécifiez une option valide.

E-REP-102-023 {0} n'est pas une commande

Explication : La syntaxe de l'utilitaire Probe Rules Utility est une commande suivie d'options et d'un chemin cible. Une commande inconnue (*{0}*) a été indiquée.

Réaction du système : Probe Rules Utility ne démarre pas.

Action de l'utilisateur : Indiquez une commande valide.

Action de l'administrateur : Indiquez une commande valide.

E-REP-102-024 Erreur d'E-S lors du listage du contenu du répertoire {0}

Explication : L'utilisateur a demandé une liste du répertoire *{0}*. Une erreur d'E-S s'est produite lors de l'accès au répertoire.

Réaction du système : Le contenu du répertoire n'est pas affiché.

Action de l'utilisateur : Vérifiez que votre compte utilisateur a accès au répertoire dans le système de fichiers et que ce répertoire existe.

Action de l'administrateur : Vérifiez que votre compte utilisateur a accès au répertoire dans le système de fichiers et que ce répertoire existe.

E-REP-102-025 La racine de référentiel doit être configurée sur un chemin absolu

Explication : L'utilisateur a spécifié une racine de référentiel qui est un chemin relatif.

Réaction du système : Probe Rules Utility ne démarre pas.

Action de l'utilisateur : Modifiez la racine de référentiel afin d'indiquer un chemin absolu.

Action de l'administrateur : Modifiez la racine de référentiel afin d'indiquer un chemin absolu.

E-REP-103-002 Impossible de trouver le fichier de propriétés {0}

Explication : Le fichier de propriétés est introuvable à l'emplacement *{0}*.

Réaction du système : Probe Rules Utility démarre avec les options par défaut.

Action de l'utilisateur : Indiquez un emplacement de fichier de propriétés valide.

Action de l'administrateur : Indiquez un emplacement de fichier de propriétés valide.

E-REP-103-003 Exception {0} lors de la lecture du fichier de propriétés {1}

Explication : Une erreur d'E-S s'est produite lors de la lecture du fichier de propriétés.

Réaction du système : Probe Rules Utility démarre avec les options par défaut.

Action de l'utilisateur : Vérifiez que le nom du fichier de propriétés *{1}* est indiqué correctement et que votre compte utilisateur y a accès. Consultez le texte de l'exception *{0}* afin de déterminer la cause du problème.

Action de l'administrateur : Vérifiez que le nom du fichier de propriétés *{1}* est indiqué correctement et que votre compte utilisateur y a accès. Consultez le texte de l'exception *{0}* afin de déterminer la cause du problème.

STK

Messages relatifs aux serveurs.

E-STK-102-003 [nstk]: User installed client language handler(s) failed to complete successfully. (*%0d*:*%1s*)

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
E-STK-102-003: [nstk] : Le ou les gestionnaires de
langue de client installés par l'utilisateur n'ont
pas pu s'exécuter correctement.
(-2353:Failed to send done to client)
```

L'exemple de message indique que la connexion entre un ObjectServer et une application client a échoué.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Cette erreur peut être provoquée par différentes conditions. La chaîne d'erreur indique la cause de l'erreur.

La cause probable de cet exemple de message d'erreur est que le client a été brusquement déconnecté de l'ObjectServer, soit parce que le processus client a été arrêté, soit en raison de problèmes réseau. Examinez le fichier journal de l'ObjectServer pour consulter les messages associés.

Si ce message apparaît fréquemment dans le fichier journal, consultez le profil suivant et déclenchez les fichiers journaux de statistiques :

- `$NCHOME/omnibus/logs/`*`nom_ObjectServer`*`_profiler_report.log1`

  Examinez le fichier journal du rapport du profileur pour connaître les problèmes de performances.
- `$NCHOME/omnibus/logs/`*`nom_ObjectServer`*`_trigger_stats.log1`

  Examinez le fichier journal des statistiques du déclencheur pour connaître les problèmes liés aux déclencheurs. Certains déclencheurs, tels que disconnect_iduc_missed et disconnect_all_clients, peuvent entraîner une brusque déconnexion des clients.

Vous devez activer la propriété **`Profile`** de l'ObjectServer pour générer ces fichiers journaux.

Pour plus d'informations sur la surveillance des performances du serveur ObjectServer, voir le *Guide d'administration d'IBM Tivoli Netcool/OMNIbus*.

Concepts associés:

«Indicateurs clés de performances Tivoli Netcool/OMNIbus», à la page 310
Pour être certain qu'un système Tivoli Netcool/OMNIbus s'exécute efficacement, vous avez la possibilité de surveiller plusieurs indicateurs clés de performances (KPI).

E-STK-102-013 [nstk]: *%0s*: Failed to register server's PID. (*%1d*:*%2s*)

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
Erreur : E-STK-102-013 : [nstk]: NCOMS1:
 Failed to register server's PID. (-100:pid file
 exists, service already running)
```

Ce message d'erreur indique que le démarrage du serveur ObjectServer a échoué car son processus est déjà en cours d'exécution.

Dans l'exemple de message, le serveur ObjectServer NCOMS1 n'a pas réussi à démarrer. Voici quelques causes possibles de cet incident :

- Une autre instance d'ObjectServer portant le même nom est déjà en cours d'exécution sur le même ordinateur.
- Le fichier d'ID processus (PID), `$NCHOME/omnibus/var/NCOMS1.pid`, comporte une entrée pour une instance du serveur ObjectServer qui n'est plus en cours d'exécution, mais un autre processus sur l'ordinateur utilise désormais le même PID.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Pour résoudre le problème indiqué par l'exemple de message, vérifiez que le PID dans le fichier `NCOMS1.pid` existe et identifiez le processus auquel il appartient.

Si le PID appartient à un processus ObjectServer (`nco_objserv`), une instance de l'ObjectServer NCOMS1 est déjà en cours d'exécution ; n'essayez donc pas de

démarrer une nouvelle instance.

Si le PID appartient à un autre processus (qui utilise le même PID qu'une instance précédemment exécutée du serveur ObjectServer), supprimez le fichier de PID et redémarrez le serveur ObjectServer.

E-STK-102-014 [nstk]: %0s: Failed to start server processing loop. (%1d:%2s)

Explication : L'exemple suivant montre comment ce message est consigné dans le journal :

```
Erreur : E-STK-102-014 : [nstk]: NCOMS01:
Failed to start server processing loop.
(-19:General failure)
```

Ce message d'erreur indique que le processus serveur a rencontré un problème et qu'il va s'arrêter.

Réaction du système : Non applicable.

Action de l'utilisateur : Non applicable.

Action de l'administrateur : Consultez les messages précédents dans le journal des erreurs pour obtenir de plus amples informations sur la cause de cette erreur.

Messages d'erreur générés par le composant interface graphique Web

Les rubriques suivantes décrivent les réponses de l'administrateur pour un sous-ensemble de messages d'erreur générés par le composant d'interface graphique Web de Tivoli Netcool/OMNIbus.

Le catalogue des messages d'erreur complet est disponible sur le site de téléchargement Tivoli au format CSV :

ftp://ftp.software.ibm.com/software/tivoli/Netcool/NetcoolOmnibus/messages/

Remarque : Le délimiteur de zone dans le fichier CSV est le caractère # (connu sous le nom de signe «dièse» ou «numéro»).

HEMIN

Messages relatifs au composant HEMIN.

HEMIN0010W

Action système

Lors de l'installation, le programme d'installation demande le nom d'utilisateur et le mot de passe de l'ObjectServer à utiliser pour la connexion de l'interface graphique Web à l'ObjectServer. Le programme d'installation ne valide pas l'exactitude du mot de passe et exécute l'installation, même si le mot de passe est incorrect, mais aucune donnée n'est extraite de l'ObjectServer.

Action de l'utilisateur

Non applicable.

Action de l'administrateur

Une fois l'installation terminée, vérifiez que le mot de passe est correct. Si le mot de passe est incorrect, modifiez-le dans les fichiers suivants :

- `ncwDataSourceDefinitions.xml` : Vous pouvez éditer ce fichier manuellement. Pour plus d'informations sur le fichier `ncwDataSourceDefinitions.xml` et pour savoir comment l'éditer, reportez-vous au *Guide d'administration et d'utilisation de l'interface graphique Web d'IBM Tivoli Netcool/OMNIbus* et recherchez *Informations de référence sur les données du fichier de configuration de la source de données*. Après avoir édité ce fichier, redémarrez le serveur Concentrateur des services d'application du tableau de bord afin que la modification prenne effet.
- `wimconfig.xml` : Vous pouvez éditer ce fichier en exécutant le script **`confvmm4ncos`**. Pour savoir comment exécuter ce script, reportez-vous au *Guide*

d'installation et de déploiement d'IBM Tivoli Netcool/OMNIbus et recherchez *Définition d'un serveur ObjectServer en tant que référentiel d'utilisateurs*..

Remarques

Ces informations ont été développées pour les produits et les services proposés aux Etats-Unis.

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

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous octroie aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Pour le Canada, veuillez adresser votre courrier à :

IBM Director of Commercial Relations
IBM Canada Ltd.
3600 Steeles Avenue East
Markham, Ontario
L3R 9Z7
Canada

Pour obtenir des informations sur les licences concernant les produits utilisant un jeu de caractères codé sur deux octets, contactez le service de propriété intellectuelle d'IBM de votre pays ou envoyez vos demandes par écrit à l'adresse suivante :

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales : LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT

© Copyright IBM Corp. 1994, 2014

D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation
958/NH04
IBM Centre, St Leonards
601 Pacific Hwy
St Leonards, NSW, 2069
Australia

IBM Corporation
896471/H128B
76 Upper Ground
London SE1 9PZ
United Kingdom

IBM Corporation
JBF1/SOM1
294 Route 100
Somers, NY, 10589-0100
United States of America

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans ce document et tous les éléments sous licence disponibles s'y rapportant sont fournis par IBM conformément aux dispositions de l'ICA, des Conditions internationales d'utilisation des logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les

résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

LICENCE DE COPYRIGHT :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Des parties de ce produit contiennent du logiciel développé par Daniel Veillard.

- `libxml2-2.7.8`

Le logiciel `libxml2-2.7.8` est distribué conformément au contrat de licence suivant :
© Copyright 1998-2003 Daniel Veillard.
All Rights Reserved. Toute personne possédant une copie de ce Logiciel et des fichiers de documentation associés (le «Logiciel») est autorisée gratuitement à exploiter le Logiciel sans restriction, y compris et sans limitation à utiliser, copier, modifier, fusionner, publier, distribuer, octroyer une sous-licence, et/ou vendre des copies du logiciel et à autoriser les personnes auxquelles le Logiciel est fourni à en faire de même, sous réserve des conditions suivantes :
Les déclarations relatives au copyright ci-dessus et cette déclaration de permission, doivent être incluses dans toutes les copies ou toute partie substantielle du Logiciel.
LE LOGICIEL EST FOURNI «EN L'ETAT» SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFAÇON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. DANIEL VEILLARD NE POURRA EN AUCUN CAS ÊTRE TENU RESPONSABLE POUR TOUT DOMMAGE, QUELLES QU'EN SOIT

LA CAUSE ET LES RESPONSABILITES, CONTRACTUELLES OU NON, OU PAR FAUTE LIEE A L'UTILISATION DE CE LOGICIEL.
Sauf indication contraire dans cette notice, le nom de Daniel Veillard ne doit pas être utilisé à des fins de publicité ou de promotion de ce Logiciel sans autorisation écrite préalable de Daniel Veillard.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

Marques

AIX, IBM, le logo IBM, ibm.com, Netcool et Tivoli sont des marques d'International Business Machines Corporation aux Etats-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 Etats-Unis et/ou dans certains autres pays.

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

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

Microsoft, Windows, Windows NT et le logo Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque de The Open Group aux Etats-Unis et/ou dans certains autres pays.

Les autres noms de sociétés, de produits et de services peuvent appartenir à des tiers.

Index

Caractères spéciaux

(agrégation) SELECT 403

A

accessibilité xi
actions externes
  exécution 306
ADD COLUMN 353
Administrateur Netcool/OMNIbus
  alignement des colonnes 79
  collage d'objets 80
  configuration des couleurs de syntaxe 81
  copie d'objets 80
  couleurs de syntaxe SQL 81
  définition des préférences 79
  démarrage 70
  filtrage des lignes 80
  masquage des colonnes 79
  Navigateur Web pour l'aide en ligne 82
  nco_config 70
  options de ligne de commande 70
  propriétés 70
  sélection d'objets ObjectServer 78
  sélection des colonnes 79
  sélection des lignes à afficher 80
  sortie de 83
  support multiculturel 69
  tri des tables de résultats 79
affectation des autorisations
  rôles 392
affichage
  détails de l'index 175
  valeur de propriété de serveur pare-feu 67
AFFICHAGE DES FLOTS DE DONNEES 67
agent de processus
  arrêt 284
  connexion 74
  exécution 263
agents de processus
  arrêt 307
  démarrage automatique 266, 268
  démarrage manuel 256
aide-mémoire des commandes 345
ajout
  colonnes 151, 353
  colonnes de table 151, 353
  d'outils à des menus 103
  de séparateurs à des menus 103
  de sous-menus à des menus 103
alertes
  envoi dans le serveur ObjectServer 35
ALTER COLUMN 354
ALTER FILE 354
ALTER GROUP 355
ALTER ROLE 355
ALTER SYSTEM 356
ALTER TABLE 357
ALTER TRIGGER 358
ALTER TRIGGER GROUP 359
ALTER USER 360
arch
  répertoire du système d'exploitation xi
arrêt
  agent de processus 307
  processus 302
  serveur pare-feu 68
  services 297
arrêt du serveur pare-feu
  arrêt 68
attribution de rôles
  groupes 394
automatisations
  configuration 115
  événements affectés par le service 202
  standard 196
automatisations des événements affectés par le service 202
autorisation
  description 83
autorisations d'objet 392
autorisations système 392

B

bases de données
  configuration 148
  création 149, 362
  initialisées par le système 169
  suppression 157, 385
bases de données initialisées par le système 169
boutons d'aide 411
boutons d'aide SQL 411

C

canaux
  collage 242
  configuration 163
  copie 242
  création 238
  modification 238
  suppression 242
CASE WHEN 361
centre de documentation de Tivoli ix
CHECK STATEMENT 361
chiffrement
  mots de passe du serveur ObjectServer 23
chiffrement de mots de passe 23, 50
classes
  création 147
  modification 147
  suppression 147
collage
  canaux 242
  processus 304
  services 304
colonnes
  ajout 151, 353
  modification 151, 354
  propriétés facultatives 174
  suppression 158, 383
  types de données 171
colonnes de table
  ajout 151, 353
  modification 151, 354
  suppression 158, 383
commande de publications ix
commandes d'application
  application 345
commandes SQL 411
composants
  contrôle de processus 249
conditions 228
configuration
  bases de données 148
  canaux 163
  déclencheurs 116
  fichiers 160
  filtres de restriction 99
  groupes 89
  invites 111
  notification d'événement accéléré 236, 237, 245
  notification des événements accélérés 235, 238
  outils 107
  procédures 130
  propriétés 158
  rôles 84
  signaux 139
  utilisateurs 94
Configuration de reprise en ligne du pont pare-feu
  reprise en ligne 57
connexion
  agent de processus 74, 289
  ObjectServer 73
Connexions SSL 76
  validation des certificats serveur 76
contrôle de processus
  affichage des processus 292
  affichage des services 292
  affichage du statut de service 281
  affichage du statut du processus 281
  agents de processus 247
  ajout de processus 285
  ajout de services 285
  arrêt d'un agent de processus 284
  arrêt de l'agent de processus 307

© Copyright IBM Corp. 1994, 2014

contrôle de processus *(suite)*
arrêt de services 297
arrêt des processus 283, 302
arrêt des services 283
collage de processus 304
collage de services 304
composants 249
comptes utilisateur Windows 254
configuration du serveur de communication 254
connexion à un agent de processus 289
contrôle de processus routage de l'hôte 291
copie de processus 304
copie de services 304
création de groupes d'utilisateurs UNIX 253
création de processus 298
création de réseau 252
création de services 294
définition de dépendances 274
définition de services 274
définition des hôtes de routage 277
définition des hôtes sécurisés 275
définition des processus 270
démarrage 252
démarrage de services 296
démarrage des processus 282, 302
démarrage des services 282
édition de processus 298
édition de services 294
envoi de signaux 303
exécution des procédures externes 307
fichier de configuration 255
informations d'état de l'agent de processus 291
mise à jour du fichier de configuration 255
nco_pa_addentry 285
nco_pa_shutdown 284
nco_pa_start 282
nco_pa_status 281
nco_pa_stop 283
nco_pad 256
niveau de consignation 291
options de ligne de commande 257
présentation 247
résolution de nom d'hôte 249
suppression de processus 301
suppression de services 295
utilitaires 251, 280
conventions, police xi
conventions typographiques xi
conversions
création 143
modification 143
suppression 144
copie
canaux 242
processus 304
services 304
couleurs de gravité
création 144
modification 144
couleurs de syntaxe SQL
configuration 81
CREATE DATABASE 362
CREATE FILE 362
CREATE GROUP 363
CREATE INDEX 364
CREATE PROCEDURE 365, 367
CREATE RESTRICTION FILTER 370
CREATE ROLE 371
CREATE SIGNAL 371
CREATE TABLE 372
CREATE TRIGGER 373, 375, 377
CREATE TRIGGER GROUP 379
CREATE USER 379
CREATE VIEW 380
création
bases de données 149, 362
canaux 238
classes 147
conversions 143
couleurs de gravité 144
déclencheurs de base de données 117, 373
déclencheurs de signal 121, 375
déclencheurs temporels 124, 377
fichiers 362
Fichiers ObjectServer 160
filtres de restriction 100, 370
groupes 90, 363
groupes de déclencheurs 116, 379
index 364
invites 112
outils 108
procédures externes 134, 365
procédures SQL 130, 367
processus 298
rôles 87, 371
services 294
signaux définis par l'utilisateur 140, 371
tables 150, 372
utilisateurs 94, 379
visuels de colonne 145
vues 380

D

déclencheurs
configuration 116
configuration d'éditeurs externes 127
création 117, 121, 124
création dans les éditeurs externes 128
édition dans les éditeurs externes 128
exécution de commandes 177
modification 117, 121, 124, 358
notification des événements accélérés 181
suppression 130, 388
valeurs recommandées 340
variables 178
variables implicites 180
variables utilisateur implicites 195
déclencheurs de base de données
création 117, 373
modification 117
déclencheurs de signal
création 121, 375
modification 121
déclencheurs temporels
création 124, 377
modification 124
définition
hôtes de routage 277
hôtes sécurisés 275
DELETE 381
démarrage
Administrateur Netcool/OMNIbus 70
agents de processus 256, 266, 268
processus 302
serveur proxy 45, 46
services 296
dépendances de processus 274
DESCRIBE 382
détails de l'index
affichage 175
documentation en ligne ix
droits
description 83
DROP COLUMN 383
DROP DATABASE 385
DROP FILE 386
DROP GROUP 386
DROP INDEX 386
DROP PROCEDURE 387
DROP RESTRICTION FILTER 387
DROP ROLE 387
DROP SIGNAL 387
DROP TABLE 388
DROP TRIGGER 388
DROP TRIGGER GROUP 388
DROP USER 389
DROP VIEW 389

E

E-ALD-102-026 461
E-ALD-102-027 462
E-ALD-102-028 463
E-ALD-102-029 463
E-ALD-102-030 463
E-ALD-102-031 463
E-ALD-102-032 463
E-ALD-102-033 464
E-ALD-102-034 464
E-APM-102-005 466
E-AUT-002-006 466
E-AUT-003-004 467
E-AUT-003-011 467
E-ETC-102-036 468
E-GOB-102-041 469
E-GOB-102-057 469
E-GOB-102-061 470
E-GOB-102-128 470
E-GTK-102-073 471
E-GTK-102-113 472
E-IVM-005-001 472
E-IVM-005-002 472
E-IVM-053-005 473
E-OBX-102-017 475
E-OBX-102-018 476
E-OBX-102-019 476

E-OBX-102-020 477
E-OBX-102-068 478
E-OBX-102-122 479
E-OBX-102-133 479
E-PRO-004-004 480
E-REG-002-025 480
E-REG-002-029 481
E-REP-102-008 481
E-REP-102-009 482
E-REP-102-010 482
E-REP-102-011 482
E-REP-102-012 482
E-REP-102-013 483
E-REP-102-014 483
E-REP-102-015 483
E-REP-102-016 484
E-REP-102-018 484
E-REP-102-019 484
E-REP-102-020 485
E-REP-102-021 485
E-REP-102-022 485
E-REP-102-023 485
E-REP-102-024 486
E-REP-102-025 486
E-REP-103-002 487
E-REP-103-003 488
E-STK-102-003 488
E-STK-102-013 488
E-STK-102-014 490
éditeurs externes pour les déclencheurs 127
éditeurs externes pour les procédures 137
émission
  signaux définis par l'utilisateur 398
envoi d'alertes
  nco_postmsg 35
envoi d'événements
  nco_postmsg 35
envoi de signaux
  processus 303
événements
  envoi dans le serveur ObjectServer 35
EXECUTE PROCEDURE 389
exécution
  actions externes 306
  procédures 389
exemples
  nco_postmsg 42
expressions 227

F

F-IVM-005-001 475
fichiers
  configuration 160
  création 362
  modification 354
  suppression 386
fichiers de point de contrôle 32
Fichiers ObjectServer
  création 160
  modification 160
  suppression 162
filtres de restriction
  configuration 99
filtres de restriction *(suite)*
  création 100, 370
  modification 100
  suppression 102, 387
flots de données du serveur pare-feu
  liste 67
FOR 390
FOR EACH ROW 390
formation
  voir formation technique Tivoli xi
formation technique Tivoli xi
Formation technique Tivoli xi

G

GET CONFIG; 66
GET PROP; 67
GRANT 392
GRANT ROLE 394
GROUP BY SELECT 405
groupes
  attribution de rôles 394
  configuration 89
  création 90, 363
  modification 90, 355
  par défaut 89
  suppression 93, 386
groupes de déclencheurs
  création 116, 379
  modification 116, 359
  suppression 129, 388

H

hôtes de routage
  contrôle de processus 277
hôtes sécurisés
  contrôle de processus 275

I

Identification et résolution des problèmes
  messages d'erreur 459
IDUC 25
  intervalle de mise à jour 26
  spécification du port 26
IDUC EVTFT 395
IDUC FLUSH 396
IDUC SNDMSG 397
IF THEN ELSE 395
index
  création 364
  suppression 386
indicateur-clé de performance
  passerelles 315
indicateurs-clés de performance
  sondes 314
indicateurs clés de performances
  ObjectServer 310
indiquer
  niveau de consignation 68
informations de support xi
INSERT 397
instructions d'indexation 336
instructions de requête SQL
  optimisation AND 334
instructions de requête SQL *(suite)*
  règles d'optimisation 334
  réorganisation des prédicats 334
Instructions de requête SQL 333
instructions de requêtes SQL
  optimisation OR 334
interface interactive SQL
  isql 229, 232
  nco_sql 229, 232
  exécution des commandes SQL 229
  options de lignes de commande 232
Interface interactive SQL
  isql 229
  nco_sql 229
  présentation 229
interface SQL interactive
  mode interface graphique 164
invites
  configuration 111
  création 112
  modification 112
  suppression 115

L

lancement
  serveur pare-feu 59
liste
  flots de données du serveur pare-feu 67
  propriétés du serveur pare-feu 66

M

magasin de mémoire table_store 34
magasins de mémoire
  Stockage des données du serveur ObjectServer 31
  table_store 34
manuels ix
menus
  modification 105, 106
  personnalisation 103
  prévisualisation de la structure 107
mode sécurisé
  ObjectServer 23
  serveur proxy 50
modification
  canaux 238
  classes 147
  colonnes 151, 354
  colonnes de table 151, 354
  conversions 143
  couleurs de gravité 144
  déclencheurs 358
  déclencheurs de base de données 117
  déclencheurs de signal 121
  déclencheurs temporels 124
  fichiers 354
  Fichiers ObjectServer 160
  filtres de restriction 100
  groupes 90, 355
  groupes de déclencheurs 116, 359
  invites 112
  menus 105, 106
  outils 108

modification *(suite)*
procédures externes 134
procédures SQL 130
processus 298
rôles 87, 355
services 294
signaux définis par l'utilisateur 140
tables 357
utilisateurs 94, 360
visuels de colonne 145
mots réservés 211

N

navigateur Web
Administrateur Netcool/OMNIbus 82
nco_aes_crypt 23, 50
nco_bridgeserv 59
nco_check_store 32
nco_config 70
options de ligne de commande 70
propriétés 70
nco_g_crypt 23, 50
nco_objserv 2
nco_pa_addentry 285
nco_pa_shutdown 284
nco_pa_start 282
nco_pa_status 281
nco_pa_stop 283
nco_pad 256
options de ligne de commande 257
nco_postmsg 35
exemples 42
options de ligne de commande 39
propriétés 39
nco_proxyserv 46
nco_store_resize 31
options de ligne de commande 34
Niveau de journalisation
spécification 68
notification d'événement accéléré
arrêt des clients 244
configuration de la passerelle 236
configuration de la table alerts.status 237
configuration des déclencheurs 245
configuration des règles de la sonde 236
déconnexion des clients 243
envoi de messages aux destinataires du canal 242
notification des événements accélérés
configuration 235
configuration des canaux 238

O

ObjectServer
affichage des propriétés 158
ALTER SYSTEM 2
configuration des automatisations 115
connexion 73
fichier de propriétés 2
ObjectServer *(suite)*
gestion des fichiers d'une table de données sur un disque 31
IDUC 25
Intervalle de mise à jour IDUC 26
magasins de mémoire 31
mode sécurisé 23
modification des propriétés 158
mots réservés 211
nco_aes_crypt 23
nco_g_crypt 23
options de ligne de commande 4
priorité de l'opérateur 220
propriété Granularity 26
propriétés 4
récupération des données 32
séquence de création des fichiers 160
spécification des options de ligne de commande 2
spécification des propriétés 2
support multiculturel 27
surveillance des connexions 162
tables système 171
traitement des alertes 1
utilisation du point de contrôle 31
opérateurs 214
chaîne 215
comparaison binaire 216
comparaison de liste 217
logiques 218
mathématiques 215
opérateurs de chaîne 215
opérateurs de comparaison binaires 216
opérateurs de comparaison de liste 217
opérateurs logiques 218
opérateurs mathématiques 215
optimisation des performances 309
exemples de déclencheurs 339
exemples de requête SQL 338
instructions d'indexation 336
instructions de requête SQL 333
meilleures pratiques
agent de surveillance 323
architecture du système 321
collecte de statistiques sur les déclencheurs 318
fichiers de configuration de la sonde 322
KPI du serveur ObjectServer 310
passerelle KPI 315
profilage du serveur ObjectServer 316
Requêtes SQL 324
sonde KPI 314
statistiques des déclencheurs 321
suivi des performances 326
Table alerts.details 323
valeurs recommandées 316
flot d'événements 322
options de ligne de commande
ObjectServer 4
outils
ajout à des menus 103
configuration 107
création 108
modification 108
suppression 111
outils *(suite)*
suppression du menu 107

P

personnalisation
menus 103
pont pare-feu
configuration standard 54
pont pare-feu multiple
configuration multiple 56
ports
IDUC 26
préférences
Administrateur Netcool/OMNIbus 79
priorité de l'opérateur 220
procédure SQL
composants 193
procédures
configuration 130
configuration d'éditeurs externes 137
création 130, 134
création dans les éditeurs externes 138
édition dans les éditeurs externes 138
exécution 389
modification 130, 134
suppression 139, 387
variables utilisateur implicites 195
procédures externes
création 134, 365
modification 134
procédures SQL
création 130, 367
FOR 390
FOR EACH ROW 390
IF THEN ELSE 395
instruction du corps 193
modification 130
SET 406
Procédures SQL
CASE WHEN 361
processus
arrêt 302
collage 304
copie 304
création 298
démarrage 302
envoi de signaux 303
modification 298
suppression 301
propriété Granularity 26
propriétés
configuration 158
ObjectServer 4
propriétés du serveur pare-feu
liste 66
propriétés facultatives
colonnes 174
public ix
publications ix

R

RAISE SIGNAL 398
registry.probes, table 453
règles d'optimisation
  optimisation AND 334
  optimisation OR 334
  réorganisation des prédicats 334
  requêtes SQL 334
répertoire du système d'exploitation
  arch xi
REVOKE 398
REVOKE ROLE 400
rôles
  affectation des autorisations 392
  configuration 84
  création 87, 371
  modification 87, 355
  par défaut 84
  révocation des droits 398
  révocation pour les groupes 400
  suppression 89, 387

S

SELECT (agrégation) 403
SELECT (GROUP BY) 405
SELECT (scalaire) 401
SELECT scalaire 401
séparateurs
  ajout à des menus 103
  suppression du menu 107
serveur ObjectServer
  SQL 167
serveur pare-feu
  démarrage à l'aide de services 59
  démarrage à l'aide du contrôle de processus 59
  démarrage manuel 59
  lancement 59
  nco_bridgeserv 59
  options de lignes de commande 60
  présentation 53
  propriétés 60
serveur proxy
  connexion de sondes 50
  démarrage 45
  démarrage à l'aide de services 46
  démarrage à l'aide du contrôle de processus 46
  démarrage manuel 46
  mode sécurisé 50
  nco_aes_crypt 50
  nco_g_crypt 50
  nco_proxyserv 46
  options de ligne de commande 47
  présentation 45
  propriétés 47
services
  arrêt 297
  collage 304
  copie 304
  création 294
  démarrage 296
  modification 294
  suppression 295
SET 406
SET DATABASE 407
SET LOG LEVEL TO 68
SHOW PROPS; 66
SHUTDOWN 68
signaux 181
  configuration 139
signaux définis par l'utilisateur
  création 140, 371
  émission 398
  modification 140
  suppression 142, 387
signaux système 181
sondes
  connexion au serveur proxy 50
sortie de
  Administrateur Netcool/OMNIbus 83
sous-menus
  ajout à des menus 103
  suppression du menu 107
SQL ObjectServer
  isql 231
  nco_sql 231
  aide-mémoire des commandes 353
  Automation (Automatisation)
    déclencheurs de base de données 176
    déclencheurs de signal 176
    déclencheurs temporels 176
    Exemples d'automatisation 203
  automatisations ObjectServer
    déclencheurs 176
    groupes de déclencheurs 176
    procédures 176
    signaux 176
  commandes SQL ObjectServer
    langage de définition de données (DDL) 207
    langage de manipulation de données (DML) 207
  conventions de dénomination 213
  définir manuellement les valeurs d'incrément 357
  définir manuellement les valeurs de série 357
  fichiers SQL
    exécution des commandes SQL à partir de fichiers 231
  fonctions 221
  héritage des droits d'accès 169
  interface interactive SQL
    chiffrement de mots de passe 232
  objets du serveur ObjectServer
    bases de données 167
    fichiers 167
    filtres de restriction 167
    index 167
    tables 167
    vues 167
  Procédures externes 192
  Procédures SQL 192
  SET INCR 357
  syntaxe 214
  utilisateurs, groupes et rôles 206
support multiculturel
  Administrateur Netcool/OMNIbus 69
support multiculturel *(suite)*
  ObjectServer 27
suppression
  bases de données 157, 385
  canaux 242
  classes 147
  colonnes 158, 383
  colonnes de table 158, 383
  conversions 144
  d'outils du menu 107
  de séparateurs du menu 107
  de sous-menus du menu 107
  déclencheurs 130, 388
  fichiers 386
  Fichiers ObjectServer 162
  filtres de restriction 102, 387
  groupes 93, 386
  groupes de déclencheurs 129, 388
  index 386
  invites 115
  outils 111
  procédures 139, 387
  processus 301
  rôles 89, 387
  services 295
  signaux définis par l'utilisateur 142, 387
  tables 157, 388
  utilisateurs 98, 389
  visuels de colonne 146
  vues 389
surveillance
  connexions au serveur ObjectServer 162
  ObjectServer
    connexions 162
surveillance automatique
  à propos de la surveillance automatique 326
  configuration de la surveillance automatique 328
  environnement d'équilibrage de charge 328
  seuils 329
SVC 407

T

table alerts.application_types 426
table alerts.col_visuals 444
table alerts.colors 444
table alerts.conversions 443
table alerts.details 424
table alerts.iduc_messages 426
table alerts.journal 425
table alerts.resolutions 443
table alerts.status 415
table catalog.base_tables 430
table catalog.channel_stats 442
table catalog.columns 431
table catalog.connections 437
table catalog.database_triggers 434
table catalog.databases 429
table catalog.external_procedures 435
table catalog.files 431
table catalog.indexes 439
table catalog.memstores 428

table
catalog.primitive_signal_parameters 432
table catalog.primitive_signals 432
table catalog.procedure_parameters 436
table catalog.procedures 435
table catalog.profiles 440
table catalog.properties 437
table catalog.restrictions 431
table catalog.security_permissions 438
table catalog.signal_triggers 434
table catalog.sql_procedures 435
table catalog.tables 429
table catalog.temporal_triggers 435
table catalog.trigger_groups 433
table catalog.trigger_stats 441
table catalog.triggers 433
table catalog.views 430
table iduc_system.channel 449
table iduc_system.channel_interest 450
table iduc_system.channel_summary 450
table
iduc_system.channel_summary_cols 450
table iduc_system.iduc_stats 451
table master.class_membership 427
table master.national 448
table master.servergroups 449
table master.stats 440
table precision.entity_service 452
table
precision.service_affecting_event 451
table precision.service_details 451
table service.status 428
table tools.action_access 446
table tools.actions 445
table tools.menu_defs 448
table tools.menu_items 447
table tools.menus 446
table tools.prompt_defs 447
tables
création 150, 372
insertion de lignes de données 397
mise à jour des colonnes 408
modification 357
suppression 157, 388
suppression de lignes 381
VIA 408
WHERE 408
tables d'alertes 415
tables d'événements affectés par un
service 451
tables d'outils de bureau 445
tables de bureau 448
tables de catalogues système 428
tables de prise en charge d'outils
client 443
tables de registre 452
tables de sécurité 449
Tables de serveur ObjectServer
table master.sm_activity 455
table master.sm_thresholds 456
tables de surveillance
automatique 455
tables de serveurs ObjectServer
iduc_system.iduc_stats 451
Tables de serveurs ObjectServer
registry.probes 453
tables de registre 452
tables de services 427
tables de statistiques 439
tables du serveur ObjectServer
alerts.application_types 426
alerts.col_visuals 444
alerts.colors 444
alerts.conversions 443
alerts.details 424
alerts.iduc_messages 426
alerts.journal 425
alerts.resolutions 443
catalog.base_tables 430
catalog.channel_stats 442
catalog.columns 431
catalog.connections 437
catalog.database_triggers 434
catalog.databases 429
catalog.external_procedures 435
catalog.files 431
catalog.indexes 439
catalog.memstores 428
catalog.primitive_signal_param. 432
catalog.primitive_signals 432
catalog.procedure_parameters 436
catalog.procedures 435
catalog.profiles 440
catalog.properties 437
catalog.restrictions 431
catalog.security_permissions 438
catalog.signal_triggers 434
catalog.sql_procedures 435
catalog.tables 429
catalog.temporal_triggers 435
catalog.trigger_groups 433
catalog.trigger_stats 441
catalog.triggers 433
catalog.views 430
iduc_system.channel 449
iduc_system.channel_interest 450
iduc_system.channel_sum*_cols 450
iduc_system.channel_summary 450
master.class_membership 427
master.national 448
master.servergroups 449
master.stats 440
precision.entity_service 452
precision.service_affecting_event 451
precision.service_details 451
présentation 415
service.status 428
tables d'alertes 415
tables d'événements affectés par un
service 451
tables d'outils de bureau 445
tables de bureau 448
tables de catalogues système 428
tables de prise en charge d'outils
client 443
tables de sécurité 449
tables de services 427
tables de statistiques 439
tables IDUC 449
tools.action_access 446
tools.actions 445
tools.menu_defs 448
tools.menu_items 447
tools.menus 446
tables du serveur ObjectServer *(suite)*
tools.prompt_defs 447
tables IDUC 449
tables ObjectServer
alerts.status 415
tables système 171
traitement des alertes 1
types de données
colonnes 171

U

UPDATE 408
USE DATABASE 407
utilisateurs
affichage des connexions 98
configuration 94
création 94, 379
modification 94, 360
par défaut 94
suppression 98, 389
utilisation du point de contrôle
fichiers de point de contrôle 32
nco_check_store 32
ObjectServer 31

V

valeur de propriété de serveur pare-feu
affichage 67
valeurs recommandées
déclencheurs 340
variables 180, 195
variables, notation xi
variables d'environnement, notation xi
variables implicites 180
variables utilisateur implicites 195
visuels de colonne
création 145
modification 145
suppression 146
vues
création 380
suppression 389

W

W-ALD-103-002 465
W-ALD-103-003 466
W-ETC-004-066 469
WRITE INTO 410

SC43-0823-01