HITACHI
Inspire the Next

JP1 Version 13

JP1/Network Node Manager i セットアップガイド

3021-3-L32-10

前書き

■ 対象製品

適用 OS：Windows Server 2016，Windows Server 2019，Windows Server 2022

P-2942-82DL JP1/Network Node Manager i 13-10

P-2942-89DL JP1/Network Node Manager i Developer's Toolkit 13-00

適用 OS：Linux 7.1 以降，Linux 8.1 以降，Linux 9.1 以降，Oracle Linux 7.1 以降，Oracle Linux 8.1 以降，Oracle Linux 9.1 以降，SUSE Linux 12

P-8442-82DL JP1/Network Node Manager i 13-10

P-8442-89DL JP1/Network Node Manager i Developer's Toolkit 13-00

■ 輸出時の注意

本製品を輸出される場合には、外国為替及び外国貿易法の規制並びに米国輸出管理規則など外国の輸出関連法規をご確認の上、必要な手続きをお取りください。

なお、不明な場合は、弊社担当営業にお問い合わせください。

■ 商標類

HITACHI，HA モニタ，Job Management Partner 1，JP1 は，株式会社 日立製作所の商標または登録商標です。

その他記載の会社名，製品名などは，それぞれの会社の商標もしくは登録商標です。

■ マイクロソフト製品の表記について

このマニュアルでは，マイクロソフト製品の名称を次のように表記しています。

表記		製品名
Active Directory		Microsoft(R) Active Directory
Excel		Microsoft(R) Office Excel
Microsoft Cluster Service		Microsoft(R) Cluster Service
Windows	Windows Server 2016	Microsoft(R) Windows Server(R) 2016 Datacenter
		Microsoft(R) Windows Server(R) 2016 Standard
	Windows Server 2019	Microsoft(R) Windows Server(R) 2019 Datacenter
		Microsoft(R) Windows Server(R) 2019 Standard
	Windows Server 2022	Microsoft(R) Windows Server(R) 2022 Datacenter
		Microsoft(R) Windows Server(R) 2022 Standard

表記	製品名
WSFC	Microsoft(R) Windows Server(R) Failover Cluster

■ その他

この製品には，Apache Software Foundation で開発されたソフトウェアが含まれています。
(http://www.apache.org)
この製品には，Indiana University Extreme! Lab で開発されたソフトウェアが含まれています。
(http://www.extreme.indiana.edu)
この製品には，The Legion Of The Bouncy Castle によって開発されたソフトウェアが含まれています。
(http://www.bouncycastle.org)

■ 発行

2024 年 9 月 3021-3-L32-10

■ 著作権

All Rights Reserved. Copyright (C) 2023, 2024, Hitachi, Ltd.
This software and documentation are based in part on software and documentation under license from Open Text or one of its affiliates, including Micro Focus. The copyright for Open Text Software and Documentation is owned by Open Text. 2014-2024 Copyright Open Text.

変更内容

変更内容（3021-3-L32-10）JP1/Network Node Manager i 13-10，JP1/Network Node Manager i Developer's Toolkit 13-00

追加・変更内容	変更個所
Application Experience サービスに関する記述を削除した。	1.2
次のトラップサーバープロパティを削除した。 • com.hp.nms.trapd.updateSourcesPeriod • com.hp.nms.trapd.notifySourcesPeriod • com.hp.ov.nms.trapd.hosted.object.trapstorm.enabled • com.hp.ov.nms.trapd.hosted.object.trapstorm.threshold • com.hp.ov.nms.trapd.analysis.windowSize • com.hp.ov.nms.trapd.analysis.numSources • com.hp.ov.nms.trapd.rmiPort • com.hp.ov.nms.trapd.connectToWinSNMP	8.3.5
高可用性（HA）に関する記述を次のとおり追加・変更した。 • CLUSTERPRO の追加 • Windows の場合の説明を，WSFC も含める説明に変更 • 共有ディスクについての説明の追加 • HA 構成を解除する順番の説明の追加	19.1，19.2，19.3.1，19.3.3，19.4.2，19.4.2(1)，19.4.3，19.4.4，19.6.2，19.7
SNMPv3 以外で管理されているノードまたは監視対象外のノードの SNMPv3 トラップを認証するための設定について記述を変更した。	21.8.1
NNMi の再起動後も設定を維持する手順にあるプロパティファイル内の検索用コメントを変更した。	21.23.2
Destination Unreachable を受信した場合でも，リトライ処理を継続するように設定する手順を追加した。	21.28
スタンドアロンの NNMi 管理サーバーの IP アドレスを変更した後の説明を変更した。	22.4
サポートする TLS プロトコルに TLSv1.3 を追加した。	23.2
「バージョン 9・10・11・12 の NNMi からの移行」の説明を，13-10 へバージョンアップするための説明に変更した。	24.
SNMPv3 通信の認証プロトコルの設定値を変更した。	付録 G.11
`nnmsamlconfig.ovpl` コマンドの説明を追加した。	付録 G.46

単なる誤字・脱字などはお断りなく訂正しました。

はじめに

このマニュアルは，JP1/Network Node Manager i および JP1/Network Node Manager i Advanced（以降，製品ごとに差異がない場合は NNMi と省略します）を導入するために必要な設定について説明したものです。

なお，このマニュアルは各 OS 共通のマニュアルです。OS ごとに差異がある場合は，本文中でそのつど内容を書き分けています。

■ 対象読者

NNMi を使用してネットワークの分散管理システムの構築を検討および実現する方を対象としています。熟練したシステム管理者，ネットワークエンジニア，または大規模システムのネットワークの導入および管理の経験がある方を対象としています。

■ マニュアルの構成

このマニュアルは，次に示す編から構成されています。

第 1 編　準備編

NNMi をインストールする前に必要な準備作業とインストール・アンインストールについて説明しています。

第 2 編　入門編

NNMi でネットワーク管理を始めるために最低限必要な設定について説明しています。

第 3 編　設定編

ネットワーク管理をするための設定について説明しています。

第 4 編　詳細設定編

証明書や，NNMi と LDAP によるディレクトリサービスの統合など，NNMi の機能を使用するための設定について説明しています。

第 5 編　高可用性環境設定編

高可用性（HA）クラスタやアプリケーションフェイルオーバーへの対応について説明しています。

第 6 編　NNMi のメンテナンス編

NNMi のバックアップ，リストア，および保守方法について説明しています。

第 7 編　移行編

バージョン 13 へ NNMi を移行するために必要な操作について説明しています。

第 8 編　NNMi との統合編

関連製品と NNMi との統合について説明しています。

第９編　連携編

NNMi と関連製品との連携について説明しています。

■ 新・旧の対応表

このマニュアル「JP1/Network Node Manager i セットアップガイド（3021-3-L32）」は，次のマニュアルの内容を取り込んだ上で，バージョン 13 のエンハンス内容を追加，更新しました。

- JP1/Cm2/Network Node Manager i　インストールガイド（3021-3-241-20）
- Job Management Partner 1/Consolidated Management 2/Network Node Manager i　インストールガイド（3021-3-342-20）

旧セットアップガイドと新セットアップガイドの構成の対応は次のとおりです。

旧セットアップガイドと新セットアップガイドの構成対応表

旧（セットアップガイド）	新（セットアップガイド）
【第 1 編　準備編】	【第 1 編　準備編】
1　ハードウェアとソフトウェアの要件	－
1.1　対応ハードウェアとソフトウェア	（「1.1　ハードウェアおよびソフトウェアを確認する」に対応）
1.2　システム設定（UNIX）	（「付録 A　NNMi の man ページを表示できない場合（Linux）」に移動）
（インストールガイドの「2　インストール前チェックリスト」を取り込み）	1　インストール前チェックリスト
	1.1　ハードウェアおよびソフトウェアを確認する
	1.2　インストール前の NNMi 管理サーバー環境を準備する
	1.3　DNS 設定を確認する
	1.4　NNMi クイックスタート設定ウィザードを使用するための準備をする
（インストールガイドの「付録 A　インストール時の補足情報」を取り込み）	1.5　インストール時の補足情報
（インストールガイドの「3　NNMi のインストールとアンインストール」を取り込み）	2　NNMi のインストールとアンインストール
	2.1　NNMi をインストールする
	2.2　クイックスタート設定ウィザードを使用する
	2.3　NNMi のライセンスを取得する
	2.4　NNMi をアンインストールする
（インストールガイドの「付録 B　インストールおよび初期スタートアップのトラブルシューティング」を取り込み）	2.5　インストールおよび初期スタートアップのトラブルシューティング

旧（セットアップガイド）	新（セットアップガイド）
（インストールガイドの「4　NNMi 入門」を取り込み）	【第 2 編　入門編】
	3　NNMi 入門
	3.1　NNMi へアクセスする
	3.2　NNMi ヘルプへアクセスする
	3.3　ネットワーク検出を設定する
【第 2 編　設定編】	【第 3 編　設定編】
2　設定の一般概念	4　設定の一般概念
3　NNMi 通信	5　NNMi 通信
4　NNMi 検出	6　NNMi 検出
5　NNMi ステータスポーリング	7　NNMi ステータスポーリング
6　NNMi インシデント	8　NNMi インシデント
7　NNMi コンソール	9　NNMi コンソール
【第 3 編　詳細設定編】	【第 4 編　詳細設定編】
8　NNMi での証明書の使用	10　NNMi での証明書の使用
9　NNMi で使用する Telnet および SSH プロトコルの設定	11　NNMi で使用する Telnet および SSH プロトコルの設定
10　NNMi と LDAP によるディレクトリサービスの統合	12　NNMi と LDAP によるディレクトリサービスの統合
11　NAT 環境の重複 IP アドレスの管理	13　NAT 環境の重複 IP アドレスの管理
12　NNMi のセキュリティおよびマルチテナント	14　NNMi のセキュリティおよびマルチテナント
13　グローバルネットワーク管理	15　グローバルネットワーク管理
14　NNMiIPv6 管理機能	16　NNMi IPv6 管理機能
【第 4 編　高可用性環境設定編】	【第 5 編　高可用性環境設定編】
15　NNMi がサポートするデータの保護	17　NNMi がサポートするデータの保護
16　アプリケーションフェイルオーバー構成の NNMi を設定する	18　アプリケーションフェイルオーバー構成の NNMi を設定する
17　高可用性クラスタに NNMi を設定する	19　高可用性クラスタに NNMi を設定する
【第 5 編　NNMi のメンテナンス編】	【第 6 編　NNMi のメンテナンス編】
18　NNMi のバックアップおよびリストアツール	20　NNMi のバックアップおよびリストアツール
19　NNMi の保守	21　NNMi の保守
20　NNMi 管理サーバーの変更	22　NNMi 管理サーバーの変更

旧（セットアップガイド）	新（セットアップガイド）
21　NNMi セキュリティ	23　NNMi セキュリティ
【第 6 編　移行編】	【第 7 編　移行編】
22　バージョン 9・10-00・10-10 の NNMi からの移行	24　バージョン 9・10・11・12 の NNMi からの移行
23　バージョン 8 以前の NNM との比較	－
24　バージョン 8 以前の NNM からの移行	－
【第 7 編　NNMi との統合編】	【第 8 編　NNMi との統合編】
25　NNMi Northbound インタフェース	25　NNMi Northbound インタフェース
26　JP1/Integrated Management-UniversalCMDB10.1Full※	－
－	【第 9 編　連携編】
－	26　JP1/IM2 のインテリジェント統合管理基盤との連携
－	27　RESTful API
付録	付録
（「1.2　システム設定（UNIX）」を移動）	付録 A　NNMi の man ページを表示できない場合（Linux）
（インストールガイドの「付録 C　新規インストール中に読み込む MIB 一覧」を取り込み）	付録 B　新規インストール中に読み込む MIB 一覧
付録 A　NNMi 環境変数	付録 C　NNMi 環境変数
付録 B　Causal Engine と NNMi インシデント	付録 D　Causal Engine と NNMi インシデント
付録 C　NNMi が使用するポートの一覧	付録 E　NNMi が使用するポートの一覧
－	付録 F　リファレンスページ（User Commands）
－	付録 G　リファレンスページ（Administrator Commands）
－	付録 H　リファレンスページ（File Formats）
付録 D　各バージョンの変更内容	付録 I　各バージョンの変更内容
付録 E　このマニュアルの参考情報	付録 J　このマニュアルの参考情報
付録 F　用語解説	付録 K　用語解説

注※

この章は，「Job Management Partner 1/Consolidated Management 2/Network Node Manager i　セットアップガイド（3021-3-343-20）」には記載されていません。

目次

前書き　2
変更内容　4
はじめに　5

第 1 編　準備編

1　インストール前チェックリスト　26
1.1　ハードウェアおよびソフトウェアを確認する　27
1.2　インストール前の NNMi 管理サーバー環境を準備する　28
1.3　DNS 設定を確認する　34
1.4　NNMi クイックスタート設定ウィザードを使用するための準備をする　36
1.5　インストール時の補足情報　37
1.5.1　ディスクドライブのセキュリティ設定（Windows の場合）　37
1.5.2　正式な完全修飾ドメイン名の取得または設定　37
1.5.3　NNMi コンソール用の Web ブラウザの有効化　38
1.5.4　Linux への必要なライブラリのインストール（Linux の場合）　39
1.5.5　システムアカウントのパスワードの設定　39

2　NNMi のインストールとアンインストール　40
2.1　NNMi をインストールする　41
2.1.1　NNMi をインストールする（Windows の場合）　41
2.1.2　NNMi をインストールする（Linux の場合）　45
2.1.3　インストール終了後の作業　47
2.2　クイックスタート設定ウィザードを使用する　51
2.3　NNMi のライセンスを取得する　55
2.3.1　恒久ライセンスキーのインストールを準備する　55
2.3.2　恒久ライセンスキーを取得してインストールする　56
2.3.3　一時試用ライセンスの切り替えについて　56
2.4　NNMi をアンインストールする　57
2.4.1　NNMi をアンインストールする（Windows の場合）　57
2.4.2　NNMi をアンインストールする（Linux の場合）　58
2.5　インストールおよび初期スタートアップのトラブルシューティング　60
2.5.1　インストールの問題　60
2.5.2　初期スタートアップの問題　61

第 2 編　入門編

3　NNMi 入門　65

3.1　NNMi へアクセスする　66
3.2　NNMi ヘルプへアクセスする　68
3.3　ネットワーク検出を設定する　69
3.3.1　コミュニティ文字列を設定する　69
3.3.2　自動検出ルールを設定する　70
3.3.3　検出の進行状況を確認する　72

第 3 編　設定編

4　設定の一般概念　74

4.1　タスクフローモデル　75
4.2　ベストプラクティス：既存の設定を保存する　76
4.3　ベストプラクティス：作成者属性を使用する　77
4.4　ユーザーインタフェースモデル　78
4.5　順序　79
4.6　ノードグループおよびインタフェースグループ　80
4.6.1　グループの重複　80
4.6.2　ノードグループのメンバーシップ　81
4.6.3　ノードグループのステータス　84
4.6.4　インタフェースグループ　84
4.7　ノード／インタフェース／アドレス階層　86
4.8　NNMi 設定およびデータベースのリセット　87
4.9　NNMi 設定ファイルのモデルファイル　89

5　NNMi 通信　90

5.1　通信の概念　91
5.1.1　通信の設定レベル　91
5.1.2　ネットワーク待ち時間とタイムアウト　92
5.1.3　SNMP アクセス制御　92
5.1.4　SNMP バージョンの優先　93
5.1.5　管理アドレスの優先　95
5.1.6　SNMPv3 トラップと通知　95
5.1.7　ポーリングプロトコル　96
5.1.8　nnmsnmp*.ovpl コマンドの動作　97
5.2　通信の計画作成　98
5.2.1　デフォルトの通信設定を計画する　98
5.2.2　通信設定領域を計画する　98

5.2.3 特定のノードの設定を計画する 99
5.2.4 再試行とタイムアウトの値を計画する 100
5.2.5 アクティブなプロトコルを計画する 100
5.2.6 コミュニティ文字列と認証プロファイルを計画する 101
5.3 通信の設定 103
5.3.1 SNMP プロキシを設定する 103
5.3.2 NETCONF を使用するデバイスのサポート 105
5.3.3 VMware ハイパーバイザーベースの仮想ネットワークの検出と監視 107
5.3.4 Cisco ACI ネットワークの検出と監視 111
5.3.5 マルチホーム NNMi 管理サーバー 113
5.4 通信の評価 115
5.4.1 ノードの SNMP の設定を確認する 115
5.4.2 SNMP アクセスを確認する 115
5.4.3 SNMP デバイスの管理 IP アドレスを確認する 116
5.4.4 通信設定を確認する 116
5.4.5 監視設定と通信設定の一致を確認する 116
5.5 通信の調整 118

6 NNMi 検出 119
6.1 検出の概念 120
6.1.1 デバイスプロファイルとデバイスの属性 121
6.2 検出の計画 123
6.2.1 基本的な検出方法を選択する 123
6.2.2 自動検出ルールを計画する 124
6.2.3 ノード名の解決順序を計画する 127
6.2.4 サブネット接続ルールを計画する 128
6.2.5 検出シードを計画する 128
6.2.6 再検出の間隔を計画する 129
6.2.7 オブジェクトを検出しない方法を計画する 130
6.2.8 インタフェースの検出範囲 131
6.3 検出の設定 132
6.3.1 自動検出ルールを設定する場合のヒント 132
6.3.2 シードを設定する場合のヒント 132
6.3.3 リンクアグリゲーションの検出 133
6.3.4 サーバーからスイッチへのリンクアグリゲーション（S2SLA）の検出について 133
6.4 検出の評価 135
6.4.1 初期検出の進行状況をたどる 135
6.4.2 シードの検出を確認する 135
6.4.3 有効なデバイスプロファイルを確認する 136

6.4.4 ノードの検出を確認する 136
6.4.5 自動検出ルールを評価する（ルールベース検出だけ） 137
6.4.6 接続と VLAN を評価する 138
6.4.7 デバイスを再検出する 138
6.5 検出の調整 139
6.5.1 応答のないオブジェクトを削除する 139

7 NNMi ステータスポーリング 140
7.1 ステータスポーリングの概念 141
7.1.1 評価の順序 141
7.2 ステータスポーリングの計画 143
7.2.1 ポーリングチェックリスト 143
7.2.2 NNMi で監視できる項目 144
7.2.3 監視の停止 145
7.2.4 監視されないノードへのインタフェース 146
7.2.5 モニタリングの拡張 146
7.2.6 ノードグループとインタフェースグループを作成する 147
7.2.7 ポーリング間隔を計画する 150
7.2.8 収集するデータを計画する 151
7.2.9 SNMP トラップが NNMi に送信する内容を決定する 151
7.3 ステータスポーリングの設定 154
7.3.1 監視するインタフェースグループとノードグループを設定する 154
7.3.2 インタフェースの監視を設定する 155
7.3.3 ノードの監視を設定する 155
7.3.4 監視のデフォルトを設定する 156
7.4 ステータスポーリングの評価 157
7.4.1 ネットワーク監視の設定を確認する 157
7.4.2 ステータスポーリングのパフォーマンスの評価 158
7.5 ステータスポーリングの調整 160

8 NNMi インシデント 161
8.1 インシデントの概念 162
8.1.1 インシデントライフサイクル 162
8.1.2 トラップおよびインシデント転送 163
8.1.3 受信済み SNMP トラップ 164
8.1.4 MIB 165
8.1.5 カスタムインシデント属性 166
8.1.6 インシデント数の削減 167
8.1.7 インシデントの抑制，強化，およびダンプニング 168

8.1.8 ライフサイクルの移行アクション 169
8.2 インシデントの計画 170
8.2.1 処理する SNMP トラップを計画する 170
8.2.2 表示するインシデントを計画する 170
8.2.3 インシデントに対する NNMi の対応方法を計画する 170
8.3 インシデントの設定 171
8.3.1 インシデントの抑制・強化・ダンプニングを設定する 171
8.3.2 ライフサイクル移行アクションを設定する 171
8.3.3 トラップログを設定する 172
8.3.4 インシデントログを設定する 172
8.3.5 トラップサーバープロパティを設定する 173
8.4 インシデント設定のバッチロード 174
8.4.1 nnmincidentcfgdump.ovpl でインシデント設定ファイルを生成する 174
8.4.2 nnmincidentcfgload.ovpl でインシデント設定をロードする 174
8.5 インシデントの評価 176
8.6 インシデントの調整 177
8.6.1 未定義のトラップのインシデントを有効化にする 177
8.6.2 SNMP トラップの MIB データの文字列を正しく解釈し表示する 178

9 NNMi コンソール 180
9.1 ノードグループの使用例 181
9.1.1 ノードグループを作成する 182
9.1.2 ノードグループマップを設定する 184
9.1.3 ノードグループを削除する 187
9.2 ネットワークの概要マップに表示されるノードの最大数を削減する 188
9.3 ノードグループマップに表示されるノードの最大数を削減する 189
9.4 分析ペインのゲージの設定 190
9.4.1 分析ペインを無効にする 190
9.4.2 表示されるゲージ数の制限 191
9.4.3 分析ペインにあるゲージの更新間隔の設定 191
9.4.4 ゲージの非表示 191
9.4.5 表示されるノードゲージの順序の制御 192
9.4.6 表示されるインタフェースゲージの順序の制御 192
9.4.7 表示されるカスタムポーラーゲージの順序の制御 192
9.4.8 ゲージプロパティの適用方法の理解 193
9.4.9 ゲージに関する問題のトラブルシューティング 193
9.5 マップラベルのスケールサイズと境界の設定 194
9.6 Loom 図および Wheel 図の自動折りたたみしきい値の設定 195
9.7 デバイスのプロファイルアイコンをカスタマイズする 196

9.8 テーブルビューのリフレッシュレートをオーバーライドする 197

第 4 編 詳細設定編

10 NNMi での証明書の使用 198
10.1 NNMi 証明書について 199
10.2 アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定 201
10.3 PKCS #12 リポジトリを使った証明書の使用 204
10.3.1 自己署名証明書の生成 204
10.3.2 CA 署名証明書の生成 206
10.3.3 NNMi キーストアーからの証明書の削除 212
10.3.4 既存の証明書と新規の自己署名証明書または CA 署名証明書との置き換え 213
10.3.5 アプリケーションフェイルオーバー環境での証明書の使用 214
10.3.6 高可用性環境での証明書の使用 215
10.3.7 グローバルネットワーク管理環境での証明書の使用 217
10.3.8 ディレクトリサービスへの SSL 接続を設定する 219

11 NNMi で使用する Telnet および SSH プロトコルの設定 222
11.1 Telnet または SSH メニュー項目を無効にする 223
11.2 Windows 上のブラウザに Telnet または SSH クライアントを設定する 224
11.2.1 Windows オペレーティングシステム提供の Telnet クライアント 225
11.2.2 サードパーティ Telnet クライアント（標準 Windows） 226
11.2.3 サードパーティ Telnet クライアント（Windows on Windows） 227
11.2.4 サードパーティ SSH クライアント（標準 Windows および Windows on Windows） 228
11.3 Windows レジストリを変更するファイル例 230
11.3.1 nnmtelnet.reg の例 230
11.3.2 nnmputtytelnet.reg の例 230
11.3.3 nnmtelnet32on64.reg の例 230
11.3.4 nnmssh.reg の例 231

12 NNMi と LDAP によるディレクトリサービスの統合 232
12.1 NNMi ユーザーのアクセス情報と設定の方法 233
12.1.1 内部モード：NNMi データベースにすべての NNMi ユーザー情報を保存 234
12.1.2 混合モード：一部の NNMi ユーザー情報を NNMi データベースに，一部の NNMi ユーザー情報をディレクトリサービスに保存 235
12.1.3 外部モード：すべての NNMi ユーザー情報をディレクトリサービスに保存 236
12.2 ディレクトリサービスへのアクセスを設定する 238
12.2.1 タスク 1：現在の NNMi ユーザー情報をバックアップする 239
12.2.2 タスク 2：（任意）ディレクトリサービスへのセキュア接続を設定する 239
12.2.3 タスク 3：ディレクトリサービスからのユーザーアクセスを設定する 239

12.2.4 タスク 4：ユーザー名とパスワードの設定をテストする 242
12.2.5 タスク 5：（「外部モード」の設定だけ）ディレクトリサービスからのグループの取得を設定する 243
12.2.6 タスク 6：（「外部モード」の設定だけ）ディレクトリサービスグループを NNMi ユーザーグループにマッピングする 244
12.2.7 タスク 7：（「外部モード」の設定だけ）NNMi ユーザーグループ設定をテストする 245
12.2.8 タスク 8：（「外部モード」の設定だけ）インシデント割り当ての NNMi ユーザーグループを設定する 246
12.2.9 タスク 9：クリーンアップして NNMi の予期せぬアクセスを防止する 246
12.2.10 タスク 10：（任意）ユーザーグループをセキュリティグループにマッピングする 247
12.3 ディレクトリサービスのクエリー 248
12.3.1 ディレクトリサービスアクセス 248
12.3.2 ディレクトリサービスの情報 248
12.3.3 ディレクトリサービス管理者が所有する情報 253
12.3.4 ユーザー識別 254
12.3.5 ユーザーグループ識別 255
12.4 NNMi ユーザーグループを保存するディレクトリサービスの設定 259
12.5 ディレクトリサービス統合のトラブルシューティング 260
12.6 LDAP 設定ファイルリファレンス 262
12.6.1 nms-auth-config.xml ファイル 262
12.7 nms-auth-config.xml ファイルへの切り替え 265

13 NAT 環境の重複 IP アドレスの管理 266

13.1 NAT とは 267
13.2 NAT の利点 268
13.3 サポートされる NAT タイプ 269
13.4 NNMi に NAT を実装する方法 270
13.5 静的 NAT の考慮事項 271
13.5.1 静的 NAT のハードウェアとソフトウェアの要件 273
13.5.2 静的 NAT での通信 273
13.5.3 検出と静的 NAT 275
13.5.4 静的 NAT のモニタリングの設定 276
13.5.5 トラップと静的 NAT 276
13.5.6 サブネットと静的 NAT 281
13.5.7 グローバルネットワーク管理と静的 NAT 281
13.6 動的 NAT および動的 PAT の考慮事項 282
13.6.1 動的 NAT および動的 PAT のハードウェアとソフトウェアの要件 284
13.6.2 検出と動的 NAT および動的 PAT 284
13.6.3 動的 NAT のモニタリングの設定 284
13.6.4 サブネットと動的 NAT および動的 PAT 285
13.6.5 グローバルネットワーク管理と動的 NAT および動的 PAT 285

13.6.6 ネットワークアドレス変換（NAT）環境での NNMi の配備 285
13.6.7 状態とステータスの NNMi 計算 288
13.7 重複する IP アドレスマッピング 290
13.7.1 プライベート IP アドレスの範囲 290

14 NNMi のセキュリティおよびマルチテナント 291

14.1 オブジェクトのアクセス制限による影響 292
14.2 NNMi のセキュリティモデル 294
14.2.1 セキュリティグループ 294
14.2.2 セキュリティグループ構造の例 296
14.3 NNMi のテナントモデル 299
14.3.1 テナント 299
14.3.2 テナント構造の例 300
14.4 NNMi のセキュリティおよびマルチテナントを設定する 302
14.4.1 セキュリティおよびマルチテナントの設定ツール 303
14.4.2 マルチテナントを設定する 305
14.4.3 セキュリティグループを設定する 306
14.4.4 セキュリティ設定を確認する 308
14.4.5 セキュリティおよびマルチテナントの設定をエクスポートする 310
14.5 NNMi セキュリティとマルチテナントをグローバルネットワーク管理に定義する 312
14.5.1 グローバルネットワーク管理にセキュリティおよびマルチテナントの初期設定をする 313
14.5.2 セキュリティおよびマルチテナントの割り当てのグローバルネットワーク管理への影響 314

15 グローバルネットワーク管理 315

15.1 グローバルネットワーク管理の前提条件 316
15.2 グローバルネットワーク管理の利点 317
15.3 グローバルネットワーク管理の適用を検討する 319
15.3.1 複数サイトのネットワークを継続的に監視する 319
15.3.2 重要なデバイスを選択して監視する 319
15.3.3 ライセンスを考慮する 319
15.4 実践的なグローバルネットワーク管理の例 321
15.4.1 要件のレビュー 321
15.4.2 初期準備 323
15.5 リージョナルマネージャーで転送フィルタを設定する 327
15.5.1 転送されるノードを制限する転送フィルタを設定する 327
15.6 グローバルマネージャーとリージョナルマネージャーを接続する 337
15.7 global1 から regional1 と regional2 への接続ステータスを確認する 341
15.8 global1 のインベントリを確認する 343
15.9 global1 と regional1 との通信を切断する 346

15.10 グローバルネットワーク管理の追加情報 350
15.10.1 検出とデータの同期化 350
15.10.2 リージョナルマネージャーからグローバルマネージャーへのカスタム属性の複製 352
15.10.3 デバイスに対するステータスポーリングまたは設定ポーリング 353
15.10.4 グローバルマネージャーでのデバイスステータスの判定とインシデントの生成 354
15.11 グローバルネットワーク管理のトラブルシューティングのヒント 356
15.11.1 NNMi ヘルプのトラブルシューティング情報 356
15.11.2 クロック同期 356
15.11.3 グローバルネットワーク管理のシステム情報 356
15.11.4 グローバルマネージャーとリージョナルマネージャーの検出情報の同期 357
15.12 グローバルネットワーク管理環境での NNMi のバージョンアップ手順 358
15.13 グローバルネットワーク管理とアドレス変換プロトコル 359

16 NNMi IPv6 管理機能 360
16.1 NNMi IPv6 管理機能の概要 361
16.2 NNMi IPv6 管理機能を使用するための必要条件 363
16.3 NNMi IPv6 管理機能を使用するためのライセンス 364
16.4 NNMi IPv6 管理機能がサポートする環境 365
16.4.1 NNMi 管理サーバーの種類とサポートする機能 365
16.4.2 IPv6 をサポートしている SNMP MIB 365
16.5 NNMi のインストールと IPv6 管理機能の有効化 366
16.6 IPv6 管理機能を無効にする 367
16.6.1 IPv6 管理機能を無効にしたあとの IPv6 監視 368
16.6.2 IPv6 管理機能を無効にしたあとの IPv6 インベントリ 368
16.6.3 IPv6 インベントリクリーンアップ時の既知の問題点 368
16.7 IPv6 管理機能を再度有効にする 370

第 5 編 高可用性環境設定編

17 NNMi がサポートするデータの保護 373
17.1 NNMi がサポートするデータ保護の仕組み 374
17.2 NNMi がサポートするデータ保護の仕組みの比較 375

18 アプリケーションフェイルオーバー構成の NNMi を設定する 376
18.1 アプリケーションフェイルオーバーの概要 377
18.2 アプリケーションフェイルオーバーの基本セットアップ 378
18.2.1 アプリケーションフェイルオーバーを設定するための前提条件 378
18.2.2 アプリケーションフェイルオーバーの注意事項 380
18.3 アプリケーションフェイルオーバー構成の NNMi を設定する 381
18.3.1 手動によるアプリケーションフェイルオーバーの設定 381

18.3.2 NNMi クラスタセットアップウィザードを使用したアプリケーションフェイルオーバーの設定 385
18.3.3 アプリケーションフェイルオーバー通信の設定 387
18.4 アプリケーションフェイルオーバー機能の使用 389
18.4.1 アプリケーションフェイルオーバーの動作 389
18.4.2 アプリケーションフェイルオーバーのシナリオ 392
18.4.3 アプリケーションフェイルオーバー構成の NNMi 管理サーバーで使用する ovstart および ovstop コマンド 394
18.4.4 アプリケーションフェイルオーバーのインシデント 395
18.5 フェイルオーバーの問題解決後の設定 396
18.6 アプリケーションフェイルオーバーを無効にする 397
18.7 管理タスクとアプリケーションフェイルオーバー 399
18.7.1 NNMi のバージョンアップ（修正版の適用を含む） 399
18.7.2 NNMi の起動と停止および再起動 399
18.7.3 NNMi のバックアップとリストア 401
18.7.4 NNMi の設定の変更 403
18.7.5 NNMi データベースパスワードの変更 406
18.8 ネットワークレイテンシ/帯域に関する考慮 407
18.8.1 アプリケーションフェイルオーバーと NNMi データベース 407

19 高可用性クラスタに NNMi を設定する 412
19.1 HA の概念 413
19.1.1 HA 用語集 414
19.1.2 NNMi HA クラスタのシナリオ 415
19.1.3 man ページ 416
19.2 HA 用 NNMi を設定するための前提条件の検証 417
19.3 HA 設定の注意事項 419
19.3.1 関連製品を使用する場合の注意 419
19.3.2 設定作業や運用操作の注意 419
19.3.3 そのほかの注意 420
19.4 HA を設定する 421
19.4.1 HA 用の NNMi 証明書を設定する 421
19.4.2 HA 用に NNMi を設定する 421
19.4.3 HA 用に NNMi を設定する（Windows の場合） 426
19.4.4 HA 用に NNMi を設定する（Linux の場合） 435
19.5 共有 NNMi データ 444
19.5.1 NNMi の共有ディスク内のデータ 444
19.5.2 設定ファイルの複製 445
19.6 HA 設定のメンテナンス 446
19.6.1 NNMi をメンテナンスモードにする 446
19.6.2 HA クラスタ内の NNMi をメンテナンスする 447

19.7 HA クラスタ内の NNMi の設定を解除する 452
19.7.1 アクティブなクラスタノードの特定 452
19.7.2 パッシブなクラスタノードでの設定解除 452
19.7.3 アクティブなクラスタノードでの設定解除 454
19.8 HA 設定のトラブルシューティング 459
19.8.1 一般的な設定の誤り 459
19.8.2 HA リソーステスト 460
19.8.3 一般的な HA のトラブルシューティング 461
19.8.4 NNMi 固有の HA のトラブルシューティング 463
19.9 HA 設定リファレンス 468
19.9.1 NNMi HA 設定ファイル 468
19.9.2 NNMi に付属している HA 設定スクリプト 468
19.9.3 NNMi HA 設定のログファイル 469

第 6 編 NNMi のメンテナンス編

20 NNMi のバックアップおよびリストアツール 471
20.1 バックアップコマンドとリストアコマンド 472
20.2 NNMi データをバックアップする 473
20.2.1 バックアップタイプ 473
20.2.2 バックアップ領域 473
20.3 NNMi データをリストアする 476
20.3.1 同じシステムでのリストア 477
20.3.2 異なるシステムでのリストア 477
20.4 バックアップとリストアの方針 479
20.4.1 すべてのデータを定期的にバックアップする 479
20.4.2 設定変更前のデータをバックアップする 479
20.4.3 NNMi またはオペレーティングシステムのバージョンアップ前のデータをバックアップする 480
20.4.4 ファイルシステムのファイルだけをリストアする 480
20.5 データベースをバックアップおよびリストアする 481

21 NNMi の保守 482
21.1 NNMi フォルダのアクセス制御リストの管理 483
21.2 ノードグループの設定 484
21.3 ノードグループマップ設定の構成 485
21.4 通信設定の構成 486
21.5 カスタムポーラー収集エクスポートの管理 487
21.5.1 カスタムポーラー収集のエクスポートディレクトリを変更する 487
21.5.2 カスタムポーラー収集のエクスポートに使用する最大ディスク容量を変更する 488
21.5.3 カスタムポーラーメトリックスの累積周期を変更する 488

21.6 インシデントアクションの管理 490
21.6.1 同時アクション数を設定する 490
21.6.2 Jython アクションのスレッド数を設定する 490
21.6.3 アクションサーバー名のパラメーターを設定する 491
21.6.4 アクションサーバーのキューサイズを変更する 492
21.6.5 インシデントアクションのログ 492
21.7 server.properties ファイルの設定の上書き 494
21.7.1 ブラウザのロケール設定の上書き 494
21.7.2 SNMP Set オブジェクトアクセス権限の設定 495
21.8 SNMP トラップの管理 497
21.8.1 SNMPv3 以外で管理されているノードまたは監視対象外のノードの SNMPv3 トラップを認証するための NNMi の設定 497
21.8.2 SNMPv1 トラップまたは SNMPv2c トラップのブロック 499
21.8.3 Causal Engine がトラップを受け入れる期間の設定 500
21.9 trapFilter.conf ファイルでインシデントをブロックする 501
21.10 NNMi の文字セットエンコードの設定 502
21.11 MIB ブラウザパラメータの変更 503
21.12 レベル 2 オペレータがノードおよびインシデントを削除できるように構成する 504
21.13 レベル 2 オペレータがマップを編集できるように構成する 505
21.14 レベル 1 オペレータがステータスのポーリングおよび設定のポーリングを実行できるように構成する 507
21.15 プロキシ SNMP ゲートウェイによって送信されたトラップから元のトラップアドレスを判別する 509
21.15.1 トラップアドレスの順序 510
21.16 NNMi NmsTrapReceiver プロセス 511
21.16.1 NmsTrapReceiver の設定 511
21.16.2 NmsTrapReceiver プロセスの開始と停止 511
21.17 NNMi コンソールに HTTPS だけで接続する 513
21.18 リモートアクセスには暗号化を必須とするように NNMi を設定する 514
21.19 以前にサポートされていた varbind 順序を保持するように NNMi を構成する 515
21.20 古い SNMP トラップインシデントを自動でトリムする 517
21.20.1 インシデントの自動トリムを有効にする（インシデントのアーカイブを作成しない場合） 517
21.20.2 SNMP トラップインシデントの自動トリムを有効にする（インシデントのアーカイブを作成する場合） 519
21.20.3 アーカイブファイルのローテーション 520
21.20.4 保存される SNMP トラップインシデント数の最大値を変更する 521
21.20.5 SNMP トラップインシデントの自動トリムの状態を監視する 523
21.20.6 SNMP トラップインシデントの自動トリムを無効にする 523
21.21 NNMi 正規化プロパティを変更する 525
21.21.1 初期検出後の正規化プロパティ変更時の注意事項 526
21.22 データベースポートを変更する 527

21.23　NNMi 自己監視　528
21.23.1　NNMi のステータスメッセージを無効にするための設定　528
21.23.2　ヘルスインシデントを抑制するための設定　528
21.24　特定ノードに対して検出プロトコルを使用しないように設定する　530
21.24.1　検出プロトコルを使用しないように設定する　530
21.25　二次的な根本原因管理イベントにアクションを設定する　532
21.26　計画停止　533
21.27　センサーステータスの設定　534
21.27.1　物理センサーステータスの設定　534
21.27.2　ノードセンサーステータスの設定　536
21.28　Destination Unreachable を受信した場合でも，リトライ処理を継続するように設定する　539

22　NNMi 管理サーバーの変更　540
22.1　NNMi 設定移動の準備のベストプラクティス　541
22.2　NNMi 設定およびデータベースを移動する　542
22.3　NNMi 設定を移動する　543
22.4　スタンドアロンの NNMi 管理サーバーの IP アドレスを変更する　544
22.5　NNMi 管理サーバーのホスト名またはドメイン名を変更する　545

23　NNMi セキュリティ　546
23.1　組み込みデータベースツールのパスワードを入力する　547
23.2　TLS プロトコルの設定　548
23.3　NNMi データ暗号化　549
23.3.1　暗号化およびユーザーアカウントパスワード　549

第 7 編　移行編

24　バージョン 9・10・11・12 の NNMi からの移行　551
24.1　NNMi 管理サーバーをバージョンアップする　552
24.1.1　バージョン 13-00 の NNMi 管理サーバーをバージョンアップする　552
24.1.2　バージョン 9・10・11・12 の NNMi 管理サーバーをバージョンアップする　552
24.2　別の NNMi 管理サーバーにバージョンアップする　553
24.3　NNMi 13-00 からのグローバルマネージャーとリージョナルマネージャーのアップグレード　554
24.3.1　グローバルネットワーク管理によってサポートされている NNMi のバージョン　554
24.3.2　グローバルネットワーク管理のアップグレード手順　554
24.4　アプリケーションフェイルオーバー構成の NNMi 13-10 へのアップグレード　555
24.4.1　アプリケーションフェイルオーバー構成の NNMi 13-00 からのアップグレード　555

第 8 編　NNMi との統合編

25　NNMi Northbound インタフェース　563

25.1　NNMi Northbound インタフェースの概要　564
25.2　NNMi Northbound インタフェースの有効化　565
25.3　NNMi Northbound インタフェースの使用法　566
25.3.1　インシデント転送　566
25.3.2　インシデントライフサイクル状態変化通知　567
25.3.3　インシデント相関処理通知　568
25.3.4　インシデント削除通知　569
25.3.5　イベント転送フィルター　569
25.4　NNMi Northbound インタフェースの変更　571
25.5　NNMi Northbound インタフェースの無効化　572
25.6　NNMi Northbound インタフェースのトラブルシューティング　573
25.7　アプリケーションフェイルオーバーと NNMi Northbound インタフェース　575
25.7.1　ローカル Northbound アプリケーション　575
25.7.2　リモート Northbound アプリケーション　575
25.8　［NNMi-Northbound インタフェースデスティネーション］フォームのリファレンス　576
25.8.1　NNMi Northbound アプリケーションの接続パラメーター　576
25.8.2　NNMi Northbound インタフェース統合の内容　577
25.8.3　NNMi Northbound インタフェース転送先のステータス情報　580
25.8.4　NNMi Northbound インタフェースで使用される MIB 情報　580
25.8.5　NNMi Northbound インタフェースで使用される SNMP トラップ情報　581

第 9 編　連携編

26　JP1/IM3 のインテリジェント統合管理基盤との連携　582

26.1　JP1/IM3 のインテリジェント統合管理基盤との連携　583

27　RESTful API　584

27.1　RESTful API　585

付録　586

付録 A　NNMi の man ページを表示できない場合（Linux）　587
付録 B　新規インストール中に読み込む MIB 一覧　588
付録 C　NNMi 環境変数　596
付録 C.1　マニュアルで使用する環境変数　596
付録 C.2　ほかの使用可能な環境変数　597
付録 D　Causal Engine と NNMi インシデント　600
付録 D.1　因果関係解析－高度な考察　600

付録 D.2　Causal Engine の概念　600
付録 D.3　ステータスの概念　601
付録 D.4　エピソードとは　602
付録 D.5　NNMi は何を解析するのか？　602
付録 D.6　失敗のシナリオは何ですか？　605
付録 D.7　ネットワーク設定の変更　634
付録 D.8　NNMi 管理設定の変更　635
付録 E　NNMi が使用するポートの一覧　637
付録 F　リファレンスページ（User Commands）　644
付録 F.1　nnm.envvars　644
付録 F.2　nnmfindattachedswport.ovpl　645
付録 F.3　nnmprops　647
付録 F.4　nnmsetcmduserpw.ovpl　649
付録 F.5　nnmsnmpnotify.ovpl　651
付録 F.6　ovstatus　655
付録 G　リファレンスページ（Administrator Commands）　658
付録 G.1　jp1nnmiinitconfig.ovpl　658
付録 G.2　nmsdbmgr　662
付録 G.3　nnmaction　663
付録 G.4　nnmbackup.ovpl　664
付録 G.5　nnmbackupembdb.ovpl　667
付録 G.6　nnmcertmerge.ovpl　669
付録 G.7　nnmchangeembdbpw.ovpl　670
付録 G.8　nnmchangesyspw.ovpl　672
付録 G.9　nnmcluster　673
付録 G.10　nnmcommconf.ovpl　678
付録 G.11　nnmcommload.ovpl　680
付録 G.12　nnmcommunication.ovpl　683
付録 G.13　nnmconfigexport.ovpl　696
付録 G.14　nnmconfigimport.ovpl　699
付録 G.15　nnmconfigpoll.ovpl　701
付録 G.16　nnmconnedit.ovpl　703
付録 G.17　nnmcustompollerconfig.ovpl　705
付録 G.18　nnmdeleteattributes.ovpl　708
付録 G.19　nnmdeleteurlaction.ovpl　712
付録 G.20　nnmdiscocfg.ovpl　713
付録 G.21　nnmengineidfile.ovpl　715
付録 G.22　nnmhealth.ovpl　716
付録 G.23　nnmicons.ovpl　719

付録 G.24　nnmincidentcfg.ovpl　721
付録 G.25　nnmincidentcfgdump.ovpl　724
付録 G.26　nnmincidentcfgload.ovpl　727
付録 G.27　nnmldap.ovpl　730
付録 G.28　nnmlicense.ovpl　731
付録 G.29　nnmloadattributes.ovpl　733
付録 G.30　nnmloadinterfacegroups.ovpl　737
付録 G.31　nnmloadipmappings.ovpl　743
付録 G.32　nnmloadmib.ovpl　744
付録 G.33　nnmloadnodegroups.ovpl　748
付録 G.34　nnmloadseeds.ovpl　755
付録 G.35　nnmmanagementmode.ovpl　757
付録 G.36　nnmmonconfig.ovpl　762
付録 G.37　nnmnodedelete.ovpl　763
付録 G.38　nnmnodegroup.ovpl　765
付録 G.39　nnmnodegroupmapsettings.ovpl　773
付録 G.40　nnmnoderediscover.ovpl　780
付録 G.41　nnmnotes.ovpl　782
付録 G.42　nnmofficialfqdn.ovpl　786
付録 G.43　nnmresetembdb.ovpl　788
付録 G.44　nnmrestore.ovpl　790
付録 G.45　nnmrestoreembdb.ovpl　792
付録 G.46　nnmsamlconfig.ovpl　794
付録 G.47　nnmscheduledoutage.ovpl　796
付録 G.48　nnmsecurity.ovpl　802
付録 G.49　nnmseeddelete.ovpl　810
付録 G.50　nnmsetdampenedinterval.ovpl　811
付録 G.51　nnmsetofficialfqdn.ovpl　813
付録 G.52　nnmsnmpbulk.ovpl　814
付録 G.53　nnmsnmpset.ovpl　817
付録 G.54　nnmsnmpwalk.ovpl, nnmsnmpget.ovpl, nnmsnmpnext.ovpl　821
付録 G.55　nnmstatuspoll.ovpl　825
付録 G.56　nnmtopodump.ovpl　827
付録 G.57　nnmtopoquery.ovpl　841
付録 G.58　nnmtrapconfig.ovpl　844
付録 G.59　nnmtrapdump.ovpl　849
付録 G.60　nnmtrimincidents.ovpl　851
付録 G.61　ovjboss　855
付録 G.62　ovspmd　857

付録 G.63　ovstart　862
付録 G.64　ovstop　866
付録 H　リファレンスページ（File Formats）　870
付録 H.1　disco.NoVLANIndexing　870
付録 H.2　disco.SkipXdpProcessing　871
付録 H.3　hostnolookup.conf　873
付録 H.4　ipnolookup.conf　874
付録 H.5　macdedupexceptions.txt　876
付録 H.6　nnm.ports　877
付録 H.7　nnm.properties　879
付録 H.8　incidentconfiguration.format　880
付録 H.9　nnmtrapd.conf　896
付録 H.10　trapFilter.conf　898
付録 I　各バージョンの変更内容　901
付録 I.1　13-10 の変更内容　901
付録 I.2　13-00 の変更内容　902
付録 I.3　12-60 の変更内容　903
付録 I.4　12-50 の変更内容　903
付録 I.5　12-10 の変更内容　904
付録 I.6　12-00 の変更内容　905
付録 I.7　11-50 の変更内容　907
付録 I.8　11-10 の変更内容　909
付録 I.9　11-00 の変更内容　913
付録 I.10　10-50 の変更内容　914
付録 I.11　10-10 の変更内容　917
付録 J　このマニュアルの参考情報　925
付録 J.1　関連マニュアル　925
付録 J.2　このマニュアルでの表記　925
付録 J.3　このマニュアルで使用する英略語　925
付録 J.4　このマニュアルで使用する記号　927
付録 J.5　KB（キロバイト）などの単位表記について　927
付録 K　用語解説　928

索引　940

1 インストール前チェックリスト

ここでは，NNMi をインストールする前に必要な準備と確認方法について説明しています。

1.1　ハードウェアおよびソフトウェアを確認する

NNMi のインストールを開始する前に，NNMi のリリースノートに記載されているハードウェアおよびソフトウェアに関する情報をお読みください。

また，監視対象の規模に変更があった場合，リリースノートの「4.　メモリ所要量およびディスク占有量」および「9.1　システム」を参考にして，Java 最大ヒープサイズ（-Xmx）の値を見直してください。

1.2　インストール前の NNMi 管理サーバー環境を準備する

NNMi 管理サーバーとは，NNMi ソフトウェアがインストールされているサーバーのことです。各 NNMi 管理サーバーは，64 ビットマシンである必要があります。ハードウェア要件の詳細については，「1.1 ハードウェアおよびソフトウェアを確認する」を参照してください。

NNMi 管理サーバーに NNMi をインストールする前に，表 1-1 のチェックリストでチェックを実施してください。

重要

NNMi のインストールや設定作業を行う前に，リモートデスクトップに次の設定を行ってください。なお，この設定を行うと Windows の使用リソースが増加しますので，作業終了後に必要に応じて設定を元に戻してください。

- 設定個所

  ［ローカル グループ ポリシー エディタ］※＞［コンピューターの構成］＞［管理用テンプレート］＞［Windows コンポーネント］＞［リモート デスクトップ サービス］＞［リモート デスクトップ セッション ホスト］＞［一時フォルダ］

  注※　［ローカル グループ ポリシー エディタ］は，スタート画面で「gpedit.msc」と入力して開くことができます。

- 設定内容

  「セッションごとの一時フォルダを使用しない」と「終了時に一時フォルダを削除しない」を有効に設定する。設定の変更をシステムに反映させるため，一度ログオフし，再度ログオンする。

表 1-1　NNMi 管理サーバーのインストール前チェックリスト

チェック欄 （はい/いいえ）	NNMi 管理サーバーの準備
	NNMi のインストールに失敗した環境で再度インストールを行う場合，リリースノートの付録に記載されている NNMi の手動削除手順を実施してからインストールを実施してください。
	NNMi をインストールするサーバーのホスト名は，RFC に準拠したホスト名にしてください。 ホスト名に使用できる文字は，英数字（A-Z，a-z，0-9），ハイフン（-），およびドメイン名を区切るドット（.）です。 RFC に準拠していないホスト名（例：アンダーバー（_）を含むホスト名）が設定されていると，NNMi コンソールの接続やコマンドの実行に失敗する場合があります。
	NNMi をインストールするサーバーで，あらかじめ自ホストの名前解決ができること，および localhost が 127.0.0.1 で名前解決できることを確認してください。
	Windows OS のシステムドライブが C ドライブであることを確認してください。システムドライブが C ドライブ以外の環境には，NNMi をインストールできません。

チェック欄 （はい/いいえ）	NNMi 管理サーバーの準備
	Windows NNMi をインストールするディスクドライブやデータディレクトリを配置するディスクドライブに厳しいセキュリティ設定をしている場合，セキュリティの設定が必要になる場合があります。詳細については，「1.5.1　ディスクドライブのセキュリティ設定（Windows の場合）」を参照してください。
	Windows SNMP サービスをチェックしてください。SNMP Service がインストールされている場合，このサーバーで SNMP Trap Service を無効にする必要があります。
	対応 Web ブラウザをインストールして有効にします。「1.1　ハードウェアおよびソフトウェアを確認する」および「1.5.3　NNMi コンソール用の Web ブラウザの有効化」を参照してください。
	DHCP（Dynamic Host Configuration Protocol）をお使いの場合，NNMi 管理サーバーには，常に同じ IP アドレスが割り当てられるように設定してください。
	NNMi のインストールが完了するまで，ウイルス対策ソフトウェアを無効にしてください。NNMi のインストールが完了したら，各ウイルス対策サービスを再起動してください。
	Linux Linux サーバーに NNMi をインストールするためには，NNMi が必要とする次のパッケージをインストールしておくことが必要です。また，それぞれの依存関係があるライブラリファイルについてもインストールしてください。 • `libaio (x86_64)` • `libXtst (x86_64)` • `libXi (x86_64)` • `net-tools (x86_64)` • `unzip (x86_64)` • `fontconfig (x86_64)` • `liberation-sans-fonts (noarch)`※ 注※ `fc-list`コマンドで一つ以上のフォントが出力されている場合は，このパッケージは不要です。 RHEL 8.1 以降，または Oracle Linux 8.1 以降では追加で以下をインストールしてください。 • `libnsl (x86_64)` 詳細については，「1.5.4　Linux への必要なライブラリのインストール（Linux の場合）」を参照してください。
	NNMi が使用するデータベースは PostgreSQL です。PostgreSQL がインストールされているサーバーに NNMi をインストールする場合，ポート競合が発生しないようにする必要があります。NNMi で使用する PostgreSQL のポートは，5432／TCP です。そのため，既存の PostgreSQL のポートを 5432／TCP 以外に変更してからインストールしてください。必要に応じて，インストール後に NNMi で使用する PostgreSQL のポートを変更してください。
	NNMi が使用するすべてのポートが利用できることを確認します。NNMi が使用するポート一覧やファイアウォールの通過方向は，「付録 E　NNMi が使用するポートの一覧」を参照してください。
	自ホストの IP アドレスとの通信をファイアウォールなどでブロックしないでください。

チェック欄 (はい/いいえ)	NNMi 管理サーバーの準備
	Windows NNMi は，［コントロールパネル］の言語の設定で，［形式］に設定した言語でインストールされます。［形式］に設定した以外の言語で NNMi を使用することはできません。OS インストール時の言語から変更する場合は，OS インストール時の言語は英語だけとします。NNMi をインストールする前に，次の設定を行ってください。 1. ［コントロールパネル］の言語の設定で，［形式］の言語を［表示言語］の設定に合わせ，日本語または英語または中国語に設定します。 なお，上書きインストールの場合は，ここで指定する言語を上書き前の NNMi の言語と一致させる必要があります。 2. ［設定のコピー］でシステムアカウントに設定をコピーします。 3. ［Unicode 対応ではないプログラムの言語］の言語を設定します。 手順 1 で［形式］に設定した言語と同じ言語を設定します。 Linux NNMi 管理サーバーのロケールには次のどれかを設定してください。 ja_JP.utf8，ja_JP.UTF-8，C，en_US.utf8，en_US.UTF-8，zh_CN.utf8
	古いバージョンの NNM をアンインストールしてもその設定情報が残っています。以前の古い情報は，古いバージョンの NNM のリリースノートを参照して，インストール前に削除しておくようにしてください。
	Windows この製品をインストールする前に，Windows のサービス画面（［コントロール パネル］＞［管理ツール］＞［サービス］より起動される画面）が起動していないことを確認してください。起動している場合はサービス画面を閉じてください。
	Windows システム環境変数Path に設定されている文字列の長さと次のディレクトリのパスの長さの和が 950 バイト以上になる場合，この製品のインストールに成功しても，次のパスがシステム環境変数Path に追加されないことがあります。 • %NnmInstallDir%bin¥; • %NnmDataDir%shared¥nnm¥actions¥; インストール終了後，システム環境変数Path に上記のパスが追加されていない場合は，手動で追加してください。 環境変数の詳細については，「付録 C.1　マニュアルで使用する環境変数」を参照してください。
	Windows NNMi をデフォルト以外のパスにインストールする場合，インストールディレクトリとデータディレクトリの名称に使用できる文字は，英数字（A-Z，a-z，0-9），ハイフン（-），ピリオド（.），アンダーバー（_），半角スペース（ ）です。ただし，複数連続する半角スペースは使用できません。また，それぞれの絶対パスの最大長は 60 文字です。
	Windows *<drive>*:¥Documents and Settings など接合点を含むパスは指定しないでください。一時ファイルが削除されないなどの不具合が発生する場合があります。

チェック欄 (はい/いいえ)	NNMi 管理サーバーの準備
	Windows 環境変数%TEMP%と%TMP%の値が異なる環境に，NNMi をインストールすると，インストールに失敗することがあります。インストール前に環境変数%TEMP%と%TMP%の値が同じであることを確認してください。異なる場合は，%TEMP%と%TMP%に同じ値を設定してください。
	Windows 環境変数に次の変数を設定しないでください。 • LANG • LC_で始まるもの ほかの製品でこれらの環境変数が設定されている場合は，この製品との共存ができない場合があります。これらの環境変数を設定したまま，NNMi をインストールすると，インストールに失敗する場合があります。
	Windows リモートデスクトップサービスのリモートデスクトップセッションホストがインストールされている場合は，NNMi をインストールする前に次の設定が必要です。 • change user /install を実行してインストールモードに変更します。 設定の詳細はリモートデスクトップセッションホストのヘルプを参照してください。
	Windows NNMi をインストールするシステムに，OS の再起動が必要な変更をした場合は，NNMi をインストールする前に OS を再起動してください。 OS の再起動が必要な例として，次のレジストリ値が存在する場合があります。値が存在している場合，NNMi はインストールを中断することがあります。 HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Control¥Session Manager¥PendingFileRenameOperations 通常は，OS を再起動することでこのレジストリ値はなくなります。
	Windows NNMi はインストールおよびアンインストール時に，%TEMP%ディレクトリを最大 500MB 使用します。ディスク容量が不足していると，インストールやアンインストールが失敗する場合があります。 Linux NNMi はインストールおよびアンインストール時に，/tmp ディレクトリを最大 1GB 使用します。ディスク容量が不足していると，インストールやアンインストールが失敗する場合があります。
	Windows インストール作業中は，[ローカル グループ ポリシー エディタ] ＞ [コンピューターの構成] ＞ [管理用テンプレート] ＞ [Windows コンポーネント] ＞ [リモート デスクトップ サービス] ＞ [リモート デスクトップ セッション ホスト] ＞ [一時フォルダ] の「セッションごとの一時フォルダを使用しない」と「終了時に一時フォルダを削除しない」を「有効」にしてください。設定の変更をシステムに反映させるため，一度ログオフし，再度ログオンしてください。
	Linux この製品を再インストールした場合は，NNMi を前提としているほかのアプリケーションについても再インストールおよびそれに伴う各種設定をしてください。

チェック欄 (はい/いいえ)	NNMi 管理サーバーの準備
	Linux この製品を前提とするアプリケーションのインストールは，この製品のインストール後の環境設定が終了してから行ってください。また，アプリケーションをインストールする際は，必ず`ovstop` コマンドを実行してから行ってください。
	Linux Hitachi PP Installer が実行されているターミナルウィンドウのサイズは，NNMi インストール中は変更しないでください。インストール中にウィンドウサイズを変更すると，NNMi が正常にインストールできない場合があります。

	Linux
	NNMi では，UDP の受信バッファに 8MB，送信バッファに 2MB 必要です。
	バッファ用に確保されているメモリ容量の設定を変更するには，`/etc/sysctl.conf` ファイルを編集して，次のエントリを追加してください。
	`# NNM settings for UDP receive and send buffer sizes`
	`net.core.rmem_max = 8388608`
	`net.core.wmem_max = 2097152`
	`/etc/sysctl.conf` ファイルの編集後に OS を再起動するか，`/sbin/sysctl -p` コマンドを実行して変更を反映させてください。

	Linux
	kernel.shmmax や kernel.shmall の値が小さ過ぎる場合があります。この場合，`/etc/sysctl.conf` ファイルを編集して次のエントリを追加してください。推奨する値は 64GB です。
	`# NNM settings for embedded database`
	`kernel.shmmax = 68719476736`
	`kernel.shmall = 68719476736`
	kernel.shmmax や kernel.shmall の値を設定する場合は，`/etc/sysctl.conf` ファイルの編集後に OS を再起動するか，`/sbin/sysctl -p` コマンドを実行して変更を反映させてください。

	Linux NNMi インストールスクリプトは，2 つのグループ（nmsgrp と nmsdb），2 つのユーザー（nmsproc と nmsdbmgr），および対応する$HOME ディレクトリを自動的に作成します。これらの操作は，次の理由によって失敗することがあります。 • IT 部門で`useradd` コマンドまたは`groupadd` コマンドを無効にしたため，ユーザーおよびグループを作成できない。 • $HOME ディレクトリが NFS 上に存在する場合に，ルートユーザーが$HOME ディレクトリを作成できない。 NNMi インストーラがこれらのグループ，ユーザー，またはディレクトリの作成に失敗すると，インストールが中止されます。この場合は，インストールを実行する前にユーザーを手動で作成してからインストールしてください。 1. nmsgrp グループで nmsproc ユーザーを作成する。 $HOME ディレクトリを任意のディレクトリに設定しますが，そのディレクトリは存在する必要があります。 2. nmsdb グループで nmsdbmgr ユーザーを作成する。 $HOME ディレクトリを任意のディレクトリに設定しますが，そのディレクトリは存在する必要があります。

チェック欄 （はい/いいえ）	NNMi 管理サーバーの準備
	これらの操作に失敗することがわかっていて，ユーザー ID，グループ ID，または$HOME の場所を制御する必要がある場合は，グループ，ユーザー，および$HOME ディレクトリを作成してからインストーラを起動できます。 useradd コマンドによってユーザーを作成した場合，デフォルトでは/home/<ユーザー名>がホームディレクトリになります。

1.3 DNS 設定を確認する

NNMi は，**ドメインネームシステム（DNS）**を使用してホスト名と IP アドレスの関係を判断します。これによって，自動検出が有効になっている場合は，大量の名前解決要求が行われることがあります。

名前解決要求を解決する際に長時間にわたる遅延を防ぐよう DNS サーバーが正しく設定されていることを確認します。NNMi の名前解決要求に応答する DNS サーバーが，次の機能を備えている必要があります。

- DNS サーバーは，権限サーバーであり，DNS 要求を転送しません。
- DNS サーバーには，ホスト名から IP アドレスへの，および IP アドレスからホスト名への一貫したマッピング情報があります。

ネットワーク内で複数の DNS サーバーが使用される場合，それらのサーバーはすべての名前解決要求に矛盾がないように応答する必要があります。

重要

ラウンドロビン DNS（Web アプリケーションサーバーの負荷分散に使用される）では，任意のホスト名が時間の経過に伴って異なる IP アドレスにマップされるおそれがあります。

メモ

`nslookup` の応答時間を改善するには，セカンダリ DNS サービスを NNMi 管理サーバーまたは NNMi 管理サーバーと同一のサブセット内の別のシステムに配置します。そして，プライマリ DNS サービスの情報をミラーリングするように，このセカンダリ DNS サービスを設定してください。また，小規模な環境では，DNS の代わりに次のファイルを使用する設定方法もあります。

- Windows：`%SystemRoot%¥system32¥drivers¥etc¥hosts`
- Linux：`/etc/hosts`

NNMi 管理サーバー上で，使用している環境に対して次が適切に設定されているかを確認します。

- オペレーティングシステムの設定によって，hosts ファイルが優先されます。hosts ファイルに最低限次の 2 つのエントリが含まれていることを確認します。

  `127.0.0.1 localhost`

  <NNMi 管理サーバーの IP アドレス> *< NNMi 管理サーバー名 >*

  NNMi 管理サーバーの IP アドレスは，NNMi 管理サーバーの FQDN の IP アドレスです。NNMi 管理サーバー名は，インストール時に設定された NNMi 管理サーバーの正式な完全修飾ドメイン名（FQDN）です。

- Windows

NNMi 管理サーバーが使用するすべての DNS サーバーに，ホスト名から IP アドレスへの，および IP アドレスからホスト名への一貫したマッピング情報があることを確認してください。

- Linux

  nslookup 検索が，nsswitch.conf ファイルで設定されているnslookup コマンド検索順序に適合することを確認してください。

  また，認識されているすべての DNS サーバーに，ホスト名から IP アドレスへの，および IP アドレスからホスト名への一貫したマッピング情報があることを確認してください。

ネットワークドメイン内の DNS の設定に問題がある（適切に解決されないホスト名やアドレスがある）ことがわかっている場合は，重要ではないデバイスが対象のnslookup 要求を避けるように設定します。これを行う利点は次のとおりです。

- スパイラル検出の速度向上
- NNMi が引き起こすネットワークトラフィックの最小化

NNMi が問題のあるデバイスを識別するには，NNMi の検出を設定する前に次の 2 つのファイルを作成します。NNMi は，これらのファイルで識別されたホスト名または IP アドレスの DNS 要求を発行しません。

- hostnolookup.conf（完全修飾ドメイン名またはホスト名のグループを識別するワイルドカードを入力）
- ipnolookup.conf（IP アドレスまたは IP アドレスのグループを識別するワイルドカードを入力）

ファイルを作成するには，テキストエディタを使用します。ファイルを NNMi 管理サーバー上の次の場所に配置します。

- Windows：%NnmDataDir%shared¥nnm¥conf¥

  %NnmDataDir%は，インストール時に指定するデータディレクトリです。
- Linux：/var/opt/OV/shared/nnm/conf/

1.4　NNMi クイックスタート設定ウィザードを使用するための準備をする

インストール後に**クイックスタート設定ウィザード**を起動すると，制限された環境（またはテスト環境）に対して，NNMi を設定できます。このウィザードを使用する場合は，表 1-2 のチェックリストでチェックを実施してください。

表 1-2　NNMi クイックスタート設定ウィザードのインストール前チェックリスト

チェック欄 （はい/いいえ）	初期環境設定の事前準備
	自動検出での IP 設定範囲を決定します※。ライセンス数（管理ノード数）に関する情報は，「2.3　NNMi のライセンスを取得する」を参照してください。
	検出シードの IP アドレスを決定します。シードの詳細については，「2.2　クイックスタート設定ウィザードを使用する」の「検出シードおよび自動検出ルールについて」を参照してください。
	検出領域内のノードの読み取り専用 SNMP コミュニティ文字列を，ネットワーク管理者から取得します。
	NNMi 管理者アカウントのユーザー名とパスワードを決定します。

注※

ネットワークアドレス変換（NAT）を使用した結果，重複する IP アドレスを含むネットワークでエリアを管理する場合は，クイックスタートウィザードで検出する 1 つのアドレスドメイン（重複しないアドレス）を選択します。次に，NNMi ヘルプの「NAT 環境内で重複するアドレス」または「13.　NAT 環境の重複 IP アドレスの管理」を参照してください。

1.5 インストール時の補足情報

ここでは，NNMi インストール時の補足情報について説明します。

1.5.1 ディスクドライブのセキュリティ設定（Windows の場合）

NNMi をインストールする前にディスクドライブのセキュリティを設定するには，次の手順に従ってください。

1. **[コンピューター]** を開いて，ディスクドライブを表示する。
2. NNMi のインストールで使用するドライブの **[プロパティ]** ＞ **[セキュリティ]** タブを開く。
3. 管理者権限のユーザーとしてログオンし，（直接，またはグループメンバーシップを介して）**[フルコントロール]** に設定されていることを確認する。設定されていない場合は，設定を変更する。
4. [セキュリティ] タブ内の詳細設定を開き，管理者権限のユーザーの [適用先] が [このフォルダー、サブフォルダーおよびファイル] に設定されているかを確認する。設定されていない場合は，設定を変更する。
5. ビルトインLocal Service ユーザーが，（直接，または所属する Users グループから継承して）**[読み取りと実行]** を選択していることを確認する。設定されていない場合は，設定を変更する。
6. [セキュリティ] タブ内の詳細設定を開き，ビルトインLocal Service ユーザーの [適用先] が [このフォルダー、サブフォルダーおよびファイル] に設定されているかを確認する。設定されていない場合は，設定を変更する。
7. 変更を適用する。
8. NNMi のインストールを続行する。

1.5.2 正式な完全修飾ドメイン名の取得または設定

NNMi ユーザーは，正式な完全修飾ドメイン名（FQDN）を使用して NNMi にアクセスします。

1. NNMi 管理サーバーの正式な FQDN を判別するには，次のどちらかの方法を使用する。
   - nnmofficialfqdn.ovpl コマンドを使用して，FQDN 設定の値を表示します。詳細については，nnmofficialfqdn.ovpl リファレンスページを参照してください。
   - NNMi コンソールで，**[ヘルプ]** ＞ **[システム情報]** の順にクリックします。**[サーバー]** タブから完全修飾ドメイン名の値を見つけます。
2. 設定した FQDN を変更する必要がある場合は，nnmsetofficialfqdn.ovpl コマンドを使用する。

   詳細については，nnmsetofficialfqdn.ovpl リファレンスページを参照してください。

1.5.3 NNMi コンソール用の Web ブラウザの有効化

NNMi にサインオンする前に，NNMi コンソールと相互動作するように Web ブラウザが設定されていることを確認してください。NNMi 管理サーバーにアクセスする各クライアントマシンの Web ブラウザで次の項目を設定してください。

- JavaScript を有効にする。
- NNMi 管理サーバーからのポップアップウィンドウの表示を許可する。
- NNMi 管理サーバーからの Cookie の保存を許可する。
- ActiveX を有効にする。
- ページの自動読み込みを有効にする。

次の手順は，Web ブラウザの設定の一例です。

重要

次の手順を完了するには，NNMi 管理サーバーの完全修飾ドメイン名が必要になります。

使用している NNMi 管理サーバーに複数のドメイン名がある場合は，NNMi では，インストール時にその中から 1 つを選択します。NNMi が使用している完全修飾ドメイン名を判断するには，`nnmofficialfqdn.ovpl` スクリプトを実行します。詳細については，`nnmofficialfqdn.ovpl` リファレンスページを参照してください。

(1) Mozilla Firefox の場合

Mozilla Firefox では，デフォルトで JavaScript の使用が有効になっています。JavaScript を無効にするにはプライバシー拡張が必要です。NNMi で JavaScript が無効になっていることを示すエラーが生成される場合，Firefox の［**アドオンマネージャ**］の［**拡張機能**］オプションを確認して，プライバシー拡張が使用されているかどうかを判別します。

1. ポップアップウィンドウを有効にするには，次の手順を実行する。
   a Mozilla Firefox で，［**ツール**］＞［**オプション**］の順にクリックします。
   b ［**コンテンツ**］をクリックします。
   c ［**ポップアップウィンドウをブロックする**］チェックボックスを有効にします。
   d ［**ポップアップウィンドウをブロックする**］チェックボックスの横にある［**許可サイト**］をクリックします。
   e NNMi 管理サーバーの完全修飾ドメイン名を，許可サイトのリストに追加し，［**許可**］をクリックします。
   f ［**閉じる**］をクリックします。
2. Cookie を有効にするには，次の手順を実行する。
   a Mozilla Firefox で，［**ツール**］＞［**オプション**］の順にクリックします。

b ［プライバシー］をクリックします。

c ［履歴］に移動し，［履歴を記憶させる］を選択します。

3. Web ブラウザを再起動する。

1.5.4 Linux への必要なライブラリのインストール（Linux の場合）

Linux サーバーに NNMi をインストールするためには，NNMi が必要とする次のパッケージをインストールしておくことが必要です。また，それぞれの依存関係があるライブラリファイルについてもインストールしてください。

- libaio (x86_64)
- libXtst (x86_64)
- libXi (x86_64)
- net-tools (x86_64)
- unzip (x86_64)
- fontconfig (x86_64)
- liberation-sans-fonts (noarch)※
  注※ fc-listコマンドで一つ以上のフォントが出力されている場合は，このパッケージは不要です。

RHEL 8.1 以降，または Oracle Linux 8.1 以降では追加で以下をインストールしてください。

- libnsl (x86_64)

詳細は，NNMi のリリースノート，およびオペレーティングシステムのドキュメントを参照してください。

1.5.5 システムアカウントのパスワードの設定

システムアカウントのパスワードは，インストール中に設定します。インストール時に設定をスキップした場合，またはパスワードを変更する場合は，nnmchangesyspw.ovpl スクリプトを使用して変更できます。次の手順に従います。

1. ovstop -c コマンドを使用して NNMi プロセスを停止する。
2. 管理者としてnnmchangesyspw.ovpl スクリプトを実行し，システムパスワードを設定する。
3. ovstart -c コマンドを使用して NNMi プロセスを開始する。

詳細については，nnmchangesyspw.ovpl リファレンスページを参照してください。

2 NNMi のインストールとアンインストール

ここでは，NNMi のインストールとアンインストールについて説明します。また，インストール後の設定方法やライセンスの取得方法についても記載しています。インストールについての情報は，リリースノートにも記載されています。併せてご覧ください。

2.1　NNMi をインストールする

Windows

ウイルス対策ソフトウェアの無効化も含めて，インストール前のチェックリストが完了していることを確認してください（「1.　インストール前チェックリスト」を参照）。

Linux

インストール前のチェックリストが完了していることを確認してください（「1.　インストール前チェックリスト」を参照）。

NNMi をバージョンアップ（修正版の適用を含む）する場合の手順については，リリースノートを参照してください。

2.1.1　NNMi をインストールする（Windows の場合）

Windows システムに NNMi を新規でインストールする手順を次に示します。

重要

NNMi のインストールや設定作業を行う前に，リモートデスクトップに次の設定を行ってください。なお，この設定を行うと Windows の使用リソースが増加しますので，作業終了後に必要に応じて設定を元に戻してください。

- 設定個所

  ［ローカル グループ ポリシー エディタ］※**＞［コンピューターの構成］＞［管理用テンプレート］＞［Windows コンポーネント］＞［リモート デスクトップ サービス］＞［リモート デスクトップ セッション ホスト］＞［一時フォルダ］**

  注※　［ローカル グループ ポリシー エディタ］は，スタート画面で「gpedit.msc」と入力して開くことができます。

- 設定内容

  「セッションごとの一時フォルダを使用しない」と「終了時に一時フォルダを削除しない」を有効に設定する。設定の変更をシステムに反映させるため，一度ログオフし，再度ログオンする。

1. NNMi をインストールするシステムに，管理者権限を持つユーザーでログオンする。

   UAC が有効な場合，ビルトイン Administrator 以外のユーザーでは，管理者への昇格が必要です。

2. NNMi の媒体をドライブにセットする。

   HITACHI 総合インストーラの画面が表示されます。

3. HITACHI 総合インストーラの指示に従って，インストールを開始する。

4. インストーラの指示に従って情報を入力する。

デフォルト値を使用する場合は，値を入力しないで，[Enter] キーを押してください。デフォルト値は，角括弧"[ ]"で囲まれた値です。

- NNMi Web サーバーの HTTP ポート番号の指定

  NNMi にアクセスするための NNMi Web サーバーの HTTP ポート番号を入力します。ほかのプログラムが使用していないポート番号を入力してください。デフォルト値は 20480 です。

  (入力例)

  ```
  ** Network Node Manager i Installer **
  * Starting NNMi installation.
  * Enter default port for HTTP server =>
  * [20480]
  8004 ↵
  ```

- NNMi Web サーバーの HTTPS ポート番号の指定

  NNMi にアクセスするための NNMi Web サーバーの HTTPS ポート番号を入力します。ほかのプログラムが使用していないポート番号を入力してください。デフォルト値は 20481 です。

  (入力例)

  ```
  * Enter default port for HTTPS server =>
  * [20481]
  8443 ↵
  ```

- インストールディレクトリの指定

  NNMi のプログラムをインストールするディレクトリを入力します。デフォルト値は，次のとおりです。

  <*drive*>:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥

  (入力例)

  ```
  * Enter program install directory =>
  * [C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥]
  C:¥Hitachi¥Cm2NNMi¥ ↵
  ```

  重要

  NNMi は<*drive*>:¥Program Files¥にインストールすることはできません。

- データディレクトリの指定

  NNMi の設定ファイルやデータベース，ログファイルなどのデータを格納するディレクトリを入力します。デフォルト値は，次のとおりです。

  <*drive*>:¥ProgramData¥Hitachi¥Cm2NNMi¥

  (入力例)

  ```
  * Enter program data directory =>
  * [C:¥ProgramData¥Hitachi¥Cm2NNMi¥]
  D:¥NNMiData¥ ↵
  ```

- **入力内容の表示およびインストール開始可否の確認**

  上記３つの入力内容が表示されるので，入力内容に問題がなくインストールを開始する場合は「yes」を，入力内容を変更したい場合は「no」を入力します。

  （入力例）

  ```
  * http port : 20480
  * https port : 20481
  * install directory : C:¥Hitachi¥Cm2NNMi¥
  * data directory : D:¥NNMiData¥
  * Do you start installation with above settings you entered ? (yes/no)
  * If you need to change the settings, please enter no.
  yes ↵
  ```

- **インストール前チェックの実施およびインストール続行可否の確認**

  インストールが開始すると，初めにインストール前チェック処理が実施されます。

  > **メモ**
  >
  > インストール前チェックでは，「表 1-1　NNMi 管理サーバーのインストール前チェックリスト」に記載されている項目を確認するため，NNMi が使用するポートが利用できるかをチェックします。
  >
  > NNMi が使用するすべてのポートが利用できることを確認します。NNMi が使用するポート一覧やファイアウォールの通過方向は，「付録 E　NNMi が使用するポートの一覧」を参照してください。

  このインストール前チェックに問題がなければ，そのままインストールを続けます。

  インストール前チェックに問題がある場合，インストーラは問題内容を出力し，インストールを続けるかユーザーに確認します。インストールを続ける場合は「yes」，インストールを終了する場合は「no」を入力してください。

  （インストール前チェックに問題があった場合の入力例）

  ```
  * Starting NNMi Precheck ...
  * TCP Port: [20481] is used.
  * UDP Port: [162] is used.
  * NNMi Precheck result: NG
  * There are some problem(s) with the settings.
  * Do you want to continue NNMi installation ? (yes/no)
  * If you enter no, the installation will stop.
  no ↵
  ```

重要

それぞれの入力の際には［Ctrl+z］を入力しないでください。［Ctrl+z］を入力するとインストールを中断します。インストールが中断した場合はインストール手順３から再開してください。

また，インストールは数十分掛かることがありますが，途中でインストールを中断しないでください。中断した場合，不正な状態となり，通常の手段では再インストールできなくなるおそれがあります。

インストール前チェックの結果が「OK」になっても，インストールの成功を保証するものではありません。このため，「1. インストール前チェックリスト」を参照し，インストール前のチェックリストが完了していることを確認してください。

- **システムアカウントのパスワードの設定**

  しばらく経過すると，システムアカウントのパスワードの設定画面が表示されるので，パスワードを入力します。

  パスワードは，1 文字以上，最大 40 文字まで入力できます。使用できる文字は，半角英数字（A-Z，a-z，0-9），およびアンダーバー（_）です。

  インストール後に設定する場合は，「¥quit」を入力します。

  ```
  * Setup system password for the initial sign-in to the NNMi console.
  * If you want to setup the password later, enter "¥quit".
  * Please enter system password:
  * Please enter system password again:
  ```

メモ

システムアカウントは，インストールプロセスで作成する特別な管理者アカウントです。NNMi コンソールへ最初にサインインするときに使用します。NNMi コンソールで管理者ロールのユーザーを作成したあとは，通常は使用しません。システムアカウントはインストール終了後も有効ですが，コマンドラインの実行や復旧目的にだけ使用されます。システムパスワードの変更方法については，「1.5.5 システムアカウントのパスワードの設定」を参照してください。

- **インストール完了の確認**

  インストールが正しく完了すると「Installation complete successfully.」という表示でコマンドプロンプトが止まりますので，Enter キーを押下してください。

- **インストール結果の確認**

  %TEMP%¥JP1NNMiInstaller.log ファイルの最後の行に，次のようにステータスの値が[0]の終了メッセージが出力されている場合，インストールは成功しています。ステータスが[0]以外の値の場合，インストールに失敗しているおそれがあります。

  ```
  [Trace] Process finished with [0]
  ```

NNMi のコマンドを実行する場合，インストール完了後に開いたコマンドプロンプト画面を使ってください。インストール前から開いている画面では NNMi の環境変数などが設定されていないため，正しく動作しません。

2.1.2 NNMi をインストールする（Linux の場合）

Linux システムに NNMi を新規でインストールする手順を次に示します。

1. NNMi をインストールするシステムに，root 権限を持つユーザーでログインする。
2. 環境変数「LC_ALL」および「LANG」に，サポート対象のロケールを設定する。

```
# LC_ALL=ja_JP.UTF-8
# export LC_ALL
# LANG=ja_JP.UTF-8
# export LANG
```

   なお，サポート対象のロケールについてはリリースノートを参照してください。

3. NNMi の媒体をドライブにセットし，ドライブをマウントする。
   ドライブのマウント方法については，NNMi のリリースノートや，OS のマニュアルを参照してください。
4. Hitachi PP Installer を起動する。
   次のコマンドを実行します。*<mount_dir>*はドライブのマウントディレクトリを示します。

```
# /<mount_dir>/X64LIN/setup  /<mount_dir>
```

   Hitachi PP Installer の起動方法の詳細は，NNMi のリリースノートを参照してください。

5. Hitachi PP Installer の起動画面で「I」を入力し，インストールできるソフトウェアの一覧を表示する。
6. カーソルを「JP1/Network Node Manager i」に移動させ，スペースキーで選択し，「I」を入力する。
   インストールを続行するか確認するメッセージが表示されます。
7. 「y」または「Y」を入力する。
8. インストーラの指示に従って情報を入力する。
   デフォルト値を使用する場合は，値を入力しないで［Enter］キーを押してください。デフォルト値は，角括弧"[ ]"で囲まれた値です。

   - NNMi Web サーバーの HTTP ポート番号の指定
     NNMi にアクセスするための NNMi Web サーバーの HTTP ポート番号を入力します。ほかのプログラムが使用していないポート番号を入力してください。デフォルト値は 20480 です。
     （入力例）

```
** Network Node Manager i Installer **
* Starting NNMi installation.
*Enter default port for HTTP server
```

```
* [20480]
8004 ↵
```

- **NNMi Web サーバーの HTTPS ポート番号の指定**

  NNMi にアクセスするための NNMi Web サーバーの HTTPS ポート番号を入力します。ほかのプログラムが使用していないポート番号を入力してください。デフォルト値は 20481 です。

  (入力例)

```
*Enter default port for HTTPS server
* [20481]
8443 ↵
```

- **入力内容の表示およびインストール開始可否の確認**

  上記の入力内容が表示されるので，入力内容に問題がなくインストール開始する場合は「yes」を，入力内容を変更したい場合は「no」を入力します。

  (入力例)

```
* http port : 20480
* https port : 20481
* Do you start installation with above settings you entered ? (yes/no)
* If you need to change the settings, please enter no.
yes ↵
```

- **インストール前チェックの実施およびインストール続行可否の確認**

  インストールが開始すると，初めにインストール前チェック処理が実施されます。

> **メモ**
>
> インストール前チェックでは，「表 1-1　NNMi 管理サーバーのインストール前チェックリスト」に記載されている項目を確認するため，NNMi が使用するポートが利用できるかをチェックします。
>
> NNMi が使用するすべてのポートが利用できることを確認します。NNMi が使用するポート一覧やファイアウォールの通過方向は，「付録 E　NNMi が使用するポートの一覧」を参照してください。

  このインストール前チェックに問題がなければ，そのままインストールを続けます。

  インストール前チェックに問題がある場合，インストーラは問題内容を出力し，インストールを続けるかユーザーに確認します。インストールを続ける場合は「yes」，インストールを終了する場合は「no」を入力してください。

  (インストール前チェックに問題があった場合の入力例)

```
* Starting NNMi Precheck ...
* TCP Port: [20481] is used.
* UDP Port: [162] is used.
```

```
* NNMi Precheck result: NG
* There are some problem(s) with the settings.
* Do you want to continue NNMi installation ? (yes/no)
* If you enter no, the installation will stop.
no ↵
```

重要

インストールは数十分掛かることがありますが，途中でインストールを中断しないでください。中断した場合，不正な状態となり，通常の手段では再インストールできなくなるおそれがあります。

インストール前チェックの結果が「OK」になっても，インストールの成功を保証するものではありません。このため，「1. インストール前チェックリスト」を参照し，インストール前のチェックリストが完了していることを確認してください。

- **システムアカウントのパスワードの設定**

しばらく経過すると，システムアカウントのパスワードの設定画面が表示されるので，パスワードを入力します。

パスワードは，1 文字以上，最大 40 文字まで入力できます。使用できる文字は，半角英数字（A-Z，a-z，0-9），およびアンダーバー（_）です。

インストール後に設定する場合は，「¥quit」を入力します。

```
* Setup system password for the initial sign-in to the NNMi console.
* If you want to setup the password later, enter ”¥quit”.
* Please enter system password:
* Please enter system password again:
```

メモ

システムアカウントは，インストールプロセスで作成する特別な管理者アカウントです。NNMi コンソールへ最初にサインインするときに使用します。NNMi コンソールで管理者ロールのユーザーを作成したあとは，通常は使用しません。システムアカウントはインストール終了後も有効ですが，コマンドラインの実行や復旧目的にだけ使用されます。システムパスワードの変更方法については，「1.5.5 システムアカウントのパスワードの設定」を参照してください。

2.1.3 インストール終了後の作業

NNMi のインストール後に必要な作業を説明します。OS の種類に関わらず実施してください。

(1) ウイルス対策ソフトウェアのウイルスチェック除外設定をする

ウイルス対策ソフトウェアの影響で，NNMi が使用しているファイルおよびディレクトリに対するファイルアクセスに，排他制御によるロックが掛かることがあります。この影響で，NNMi サービスの起動失敗や異常終了，またはコマンドの実行が遅延することがあります。NNMi の稼働中にウイルスチェックをする場合は，次のディレクトリ配下をチェック対象から外してください。

Windows の場合

- NNMi のインストールディレクトリ
- NNMi のデータディレクトリ
- *<drive>*:¥Program Files¥Hitachi¥Cm2NNMi（存在しない場合があります）
- *<HA_mount_point>*¥NNM（クラスタ構成の場合）

Linux の場合

- /opt/OV
- /var/opt/OV
- *<HA_mount_point>*/NNM（クラスタ構成の場合）

NNMi の停止中にウイルスチェックをして NNMi を再起動する場合は，上記のディレクトリに対するウイルスチェックが完了したことを確認してから起動してください。

(2) NNMi のシステムアカウントのパスワードを設定する

NNMi インストール時に，システムアカウントの設定をスキップした場合，NNMi コンソールに最初にサインインするためのアカウントのパスワードを設定します。パスワードの設定には，nnmchangesyspw.ovpl スクリプトを使用します。nnmchangesyspw.ovpl スクリプトを引数なしで実行し，メッセージに従ってパスワードを登録してください。

(3) 言語環境を設定する（Linux の場合だけ）

OS の設定などによっては，マシンのリブート時にovstart コマンドがLANG=C で自動的に起動される場合があります。この場合，バックグラウンド・プロセスでは英語のメッセージが出力されます。意図した言語でメッセージの出力をおこなうためには，システム起動時にovstart コマンドがサポートしているロケールで起動されるように，次の設定を行います。

設定手順

ovstart コマンドが NNMi インストール時の言語で起動されるように，次の設定を行ってください。

[設定個所]

/opt/OV/bin/netmgt ファイル内の /opt/OV/bin/ovstart の前

/opt/OV/bin/nettrap ファイル内の /opt/OV/bin/ovstart nnmtrapreceivermd の前

[設定内容]

次のいずれかが記載されていることを確認し，必要であれば追加してください。

```
LANG=ja_JP.utf8
export LANG
```

または

```
LANG=ja_JP.UTF-8
export LANG
```

または

```
LANG=C
export LANG
```

または

```
LANG=en_US.utf8
export LANG
```

または

```
LANG=en_US.UTF-8
export LANG
```

または

```
LANG=zh_CN.utf8
export LANG
```

(4)　Java 最大ヒープサイズを確認する

インストール中に，物理メモリに応じて Java 最大ヒープサイズ（-Xmx）の値が自動的に設定されます。リリースノートの「4.　メモリ所要量およびディスク占有量」および「9.1 システム」を参考にして，-Xmx の値を見直してください。監視対象の規模に変更があった場合も，同様に見直してください。

(5)　インシデントの自動トリム設定を確認する

NNMi 12-10 以降，新規インストールの場合，インシデントの自動トリムが有効化されます。「21.20　古い SNMP トラップインシデントを自動でトリムする」を参考にして，自動トリムの有無，トリム実施のしきい値と削除量，削除対象の設定を見直してください。監視対象の規模に変更があった場合も，同様に見直してください。

(6)　NNMi サービスを開始する

ovstart コマンドを実行して，NNMi サービスを開始します。

(7)　管理者ロールのアカウントを作成する

NNMi コンソールにサインインして，管理者ロールのアカウントを作成します。

1. NNMi サインイン用ウィンドウを表示する。

   Web ブラウザのアドレス入力用のウィンドウに次の URL を入力します。

   ```
   http://<fully_qualified_domain_name>:<port>/nnm/
   ```

   *<fully_qualified_domain_name>*は，NNMi 管理サーバーの完全修飾ドメイン名を表し，*<port>*は，インストール中に設定した NNMi Web サーバーの HTTP ポート番号を表します。

2. システムアカウントのユーザー名とパスワードを入力し，サインインボタンをクリックする。

   - ユーザー名：system
   - パスワード：「(2)　NNMi のシステムアカウントのパスワードを設定する」で作成したシステムアカウントのパスワード

3. ユーザーアカウントを作成する。

   NNMi コンソールの［設定］ワークスペース＞［セキュリティ］＞［ユーザーアカウント］＞［新規作成］アイコンをクリックします。名前とパスワードを入力してから，［保存して閉じる］アイコンをクリックしてユーザーアカウントを保存します。詳細については，NNMi ヘルプの「ユーザーアカウントを設定する（［ユーザーアカウント］フォーム）」を参照してください。

   > **重要**
   >
   > 名前は，1～40 文字までが入力できます。使用できる文字は，英数字（A-Z，a-z，0-9），ピリオド（.），アンダーライン（_），アットマーク（@），およびハイフン（-）です。
   >
   > パスワードは，1 文字以上の任意の文字数が入力できます。使用できる文字は，英数字（A-Z，a-z，0-9），および半角記号です。

4. ユーザーアカウントに管理者ロールを割り当てる。

   NNMi コンソールの［設定］ワークスペース＞［セキュリティ］＞［ユーザーアカウントのマッピング］＞［新規作成］アイコンをクリックし，次の項目を指定します。

   - ユーザーアカウント：手順 3 で作成したユーザーアカウント
   - ユーザーグループ：NNMi 管理者

   ［保存して閉じる］アイコンをクリックしてマッピングを保存します。詳細については，NNMi ヘルプの「ユーザーアカウントのマッピングタスク」を参照してください。

   > **重要**
   >
   > ユーザーグループは新規に作成しないで，デフォルトのユーザーグループから選択してください。

2.2　クイックスタート設定ウィザードを使用する

この項では，NNMi の基本的な設定タスクについて説明します。これらのタスクは，必ず NNMi をインストールしたあとに行ってください。クイックスタート設定ウィザードで設定できる項目は非常に少ないため，NNMi で監視を始めるために必要なすべての設定を行うことはできません。通常は，NNMi コンソールから設定を行うことを推奨します。次のような初期設定（例えばテスト環境）では，**クイックスタート設定ウィザード**を使用することもできます。

- SNMP コミュニティ文字列の設定
- 限られた範囲のネットワークノードの検出

  ネットワークアドレス変換（NAT）を使用した結果，重複する IP アドレスを含むネットワークでエリアを管理する場合は，クイックスタートウィザードで検出する 1 つのアドレスドメイン（重複しないアドレス）を選択します。次に，NNMi ヘルプの「NAT 環境内で重複するアドレス」または「13.　NAT 環境の重複 IP アドレスの管理」を参照してください。
- 初期管理者アカウントの設定

重要

クイックスタート設定ウィザードを使用して，SNMP バージョン 3（SNMPv3）の設定を完了させることはできません。SNMPv3 を使用して監視するデバイスがある場合は次を実行します。

1. NNMi コンソールを開く。
2. **［設定］**ワークスペースの**［通信の設定］**を選択する。
3. SNMPv3 設定を完了する。

初期設定が完了したあとは，NNMi コンソールを使って，ネットワークトポロジへのノードの追加や監視の設定のような，追加の設定タスクを行うことができます。詳細については，NNMi ヘルプを参照してください。

メモ

検出シードおよび自動検出ルールについて

検出シードとは，NNMi によるネットワークトポロジの検出を助けるためのノードです。例えば，監視環境内のコアルータなどがシードになります。各シードは，IP アドレスまたはホスト名によって識別されます。NNMi ヘルプの「自動検出ルールを設定する」を参照してください。

- シードとして指定したデバイスが，追加検出の開始ポイントとなるように検出を設定するには，**自動検出ルール**を作成して設定してください。NNMi ヘルプの「検出シードを指定する」を参照してください。
- シードとして指定したデバイスだけが検出されるように検出を設定するには，自動検出ルールを作成しないでください。

検出プロセスの概要については，NNMi ヘルプの「スパイラル検出の動作原理」を参照してください。

1. インストールプロセスが完了したあとで，次の手順でクイックスタート設定ウィザードを起動する。

   クイックスタート設定ウィザードは，インストール後すぐに実行する必要があります。**クイックスタート設定ウィザード**を手動で起動するには，次の URL にアクセスします。

   http://<*fully_qualified_domain_name*>:<*port*>/quickstart/

   <*fully_qualified_domain_name*>は NNMi 管理サーバーの完全修飾ドメイン名で，<*port*>はインストール時に設定したポート番号です。

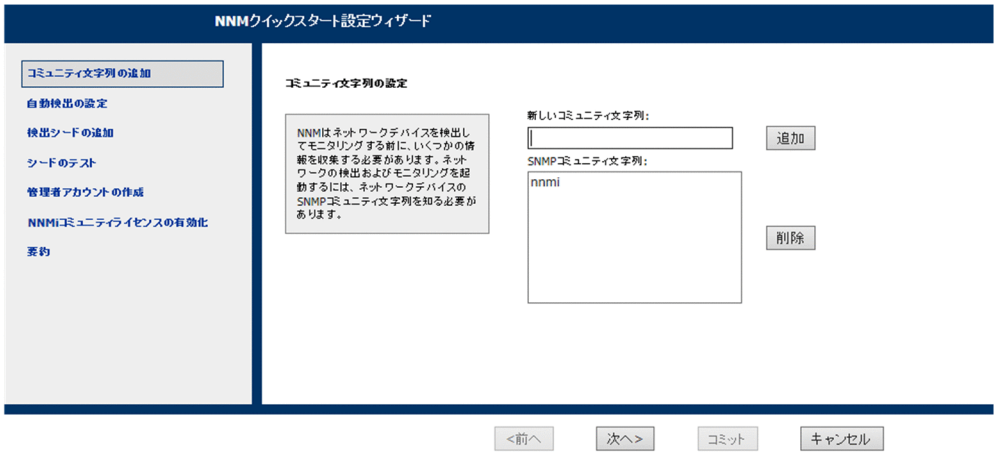
   使用している NNMi 管理サーバーに複数のドメイン名がある場合は，NNMi では，インストール時にその中から 1 つを選択します。NNMi が使用している完全修飾ドメイン名を判断するには，`nnmofficialfqdn.ovpl` スクリプトを実行します。詳細については，`nnmofficialfqdn.ovpl` リファレンスページを参照してください。

   NNMi クイックスタート設定ウィザードが，Web ブラウザのウィンドウで開きます。

2. 次のようにログインする。

   - ユーザー名：`system`
   - パスワード：「2.1.3　インストール終了後の作業」の「(1)　NNMi のシステムアカウントのパスワードを設定する」で作成したシステムアカウントのパスワードです。

3. **[コミュニティ文字列の設定]** ページで，検出範囲内にあるノードのどれかのコミュニティ文字列を入力し，**[追加]** をクリックする。



メモ

NNMi は，コミュニティ文字列を，既知のデバイスと自動的に照合します。特定のデバイスと各コミュニティ文字列の関連づけを，手動で行う必要はありません。

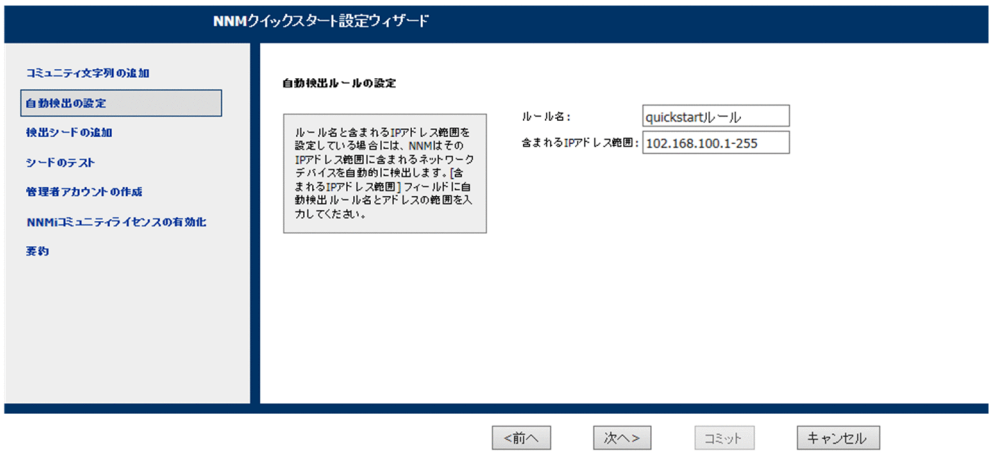
4. **[SNMP コミュニティ文字列]** のリストに，検出範囲内のすべてのノードのコミュニティ文字列が含まれるまで手順 3 を繰り返し，**[次へ]** をクリックする。

ここで追加した SNMP コミュニティ文字列が，NNMi データベースに保存されます。NNMi コンソールでは，SNMP コミュニティ文字列は，［通信の設定］フォームの［デフォルトの SNMPv1/v2 コミュニティ文字列］タブに表示されます。

5. ［自動検出ルールの設定］ページで，既存のルール名と［含まれる IP アドレス範囲］との関連づけを行う。検出規則のための IP アドレス範囲を入力し，［次へ］をクリックする。

   次は，有効な IP アドレス範囲の例です。

   - 10.1.1.*
   - 10.1.1.1-99
   - 10.10.50-55.*
   - 10.1-7.1-9.1-9



6. ［シードの設定］ページで，ネットワークに検出シードの情報を入力し，［追加］をクリックする。そのあと，［次へ］をクリックする。

   検出シードを，IP アドレスまたは完全修飾ドメイン名の形式で入力します。これらシードで示されたネットワークデバイスによって，NNMi のスパイラル検出プロセスがネットワークを検出できるようになります。

   メモ

   コマンドラインから，`nnmloadseeds.ovpl` コマンドを使用してシードをロードできます。詳細については，`nnmloadseeds.ovpl` リファレンスページを参照してください。

7. ［シードテストの結果］ページで，通信テストの結果を確認する。

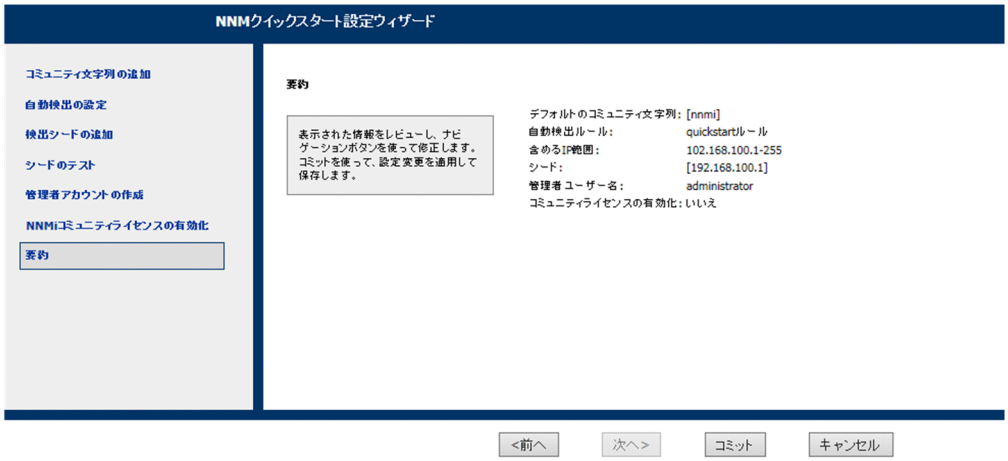
   手順 3 で特定したコミュニティ文字列では，どのシードノードにも到達できない場合には，［前へ］をクリックし，［コミュニティ文字列の設定］ページまで戻ってください。コミュニティ文字列を修正してから，［次へ］をクリックします。

8. すべてのノードに到達できるまで，手順 7 を繰り返したら，［次へ］をクリックする。

9. ［管理者アカウントの設定］ページで，NNMi ソフトウェアを管理する新規アカウントの**ユーザー名**を入力し，**パスワード**を設定して［次へ］をクリックする。

10. [要約] ページで，指定した情報を確認し，次のどちらかを実行する。

- 設定の変更を行う場合は，[前へ] をクリックします。
- 現在の設定を使用する場合，[コミット] をクリックします。



11. [ウィザードは終了しました] ページで，ネットワークの一部を検出するために NNMi を正常に設定したことが表示されたら，次のどちらかを実行する。

- 戻ってもう一度実行する場合は，[前へ] をクリックします。
- NNMi コンソールユーザーインタフェースを起動する場合は，[UI を起動] をクリックします。NNMi の使用を開始するには，「3. NNMi 入門」を参照してください。

メモ

Windows の場合，インストール後，ウイルス対策ソフトウェアを再起動します。

2.3 NNMi のライセンスを取得する

恒久ライセンスキーをインストールしていない場合は，NNMi 製品には，NNMi のインストール後 60 日間有効な一時試用ライセンスキーが含まれています。できるだけ早く，恒久ライセンスキーを入手し，インストールしてください。

重要

NNMi 12-60 より，ライセンスキーのフォーマットが変更になりました。

NNMi 12-50 以前のライセンスキーは，NNMi 12-60 以降には適用できません。NNMi 12-60 以降にバージョンアップして利用する場合は，事前に日立パスワードセンターから新フォーマットのライセンスキーを入手して，NNMi に適用する必要があります。（バージョンアップ後に新フォーマットのライセンスキーを NNMi に適用してください。）

2.3.1 恒久ライセンスキーのインストールを準備する

一時試用ライセンスでは，250 ノードまでの制限が付けられています。一時試用ライセンスキーで NNMi を実行している場合，恒久ライセンスでサポートできる数以上のノードを管理できる場合があります。しかし恒久ライセンスが有効になると，NNMi はライセンスされている数まで自動的に管理対象ノードを減少します。

ライセンス情報を追跡する際には，次の点に注意してください。

- **消費量**：NNMi は，NNMi のライセンス容量限界までノードを検出および管理します（切り上げ）。
  - **VMware環境**：デバイスプロファイルがvmwareVM の各デバイスは，1/10 のノードと同等です。
  - ほかのすべてのデバイスは 1 つの検出されたノードと同等です。

管理対象から除外するノードをご自身で決定する場合は，新規ライセンスキーをインストールする前に，重要でないノードを NNMi コンソールを使用して削除してください。

(1) ライセンスの種類および管理対象ノードの数の確認

現在，NNMi が使用しているライセンスの種類を確認するには，次の手順に従います。

1. NNMi コンソールで，［**ヘルプ**］＞［**Network Node Manager i について**］の順にクリックする。
2. ［**Network Node Manager i について**］ウィンドウで，［**ライセンス情報**］をクリックする。
3. ［**消費量**］フィールドに表示されている値を探す。
   この値が，現在 NNMi が管理しているノードの数となります。
4. 恒久ライセンスがサポートできるノード数が，現在 NNMi が管理しているノード数より少ない場合は，NNMi コンソールを使用して，あまり重要でないノードを削除する。

詳細については，NNMi ヘルプの「ノードを削除する」を参照してください。

2.3.2　恒久ライセンスキーを取得してインストールする

恒久ライセンスキーを申請するには，次の情報が必要です。

- ソフトウェア使用許諾契約書
- NNMi 管理サーバーの IP アドレス
- ライセンスキーを適用する製品媒体のバージョン・リビジョン
- お客様の企業情報または団体情報

恒久ライセンスキーの取得方法およびインストール方法は，リリースノートを参照してください。

2.3.3　一時試用ライセンスの切り替えについて

NNMi のインストール直後は，NNMi の一時試用ライセンスが適用されます。

NNMi Advanced の機能を検証する場合は，一時試用ライセンスを切り替えてください。

NNMi と NNMi Advanced の一時試用ライセンスを切り替える方法については，リリースノートを参照してください。

2.4 NNMi をアンインストールする

2.4.1 NNMi をアンインストールする（Windows の場合）

1. NNMi をアンインストールするシステムに，管理者権限を持つユーザーでログオンする。

   UAC が有効な場合，ビルトイン Administrator 以外のユーザーでは，管理者への昇格が必要です。

2. NNMi のサービスをすべて停止する。

3. **［コントロールパネル］** ＞ **［プログラムと機能］** で，「Network Node Manager」を選択して **［アンインストールと変更］** をクリックする。

4. アンインストールを開始するか確認するメッセージが表示されるので，「yes」を入力し，アンインストールを開始する。

   （入力例）

```
** Network Node Manager i Installer **
* Starting uninstallation ? (yes/no) =>
yes
```

5. アンインストールが終了したら，NNMi のインストールディレクトリとデータディレクトリを削除する。

   アンインストールを実行しても，NNMi のインストールディレクトリとデータディレクトリが削除されない場合は，手動で削除します。

   インストール時にデフォルト値を選択した場合は，次のディレクトリを削除します。

   - *<drive>*:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥
   - *<drive>*:¥ProgramData¥Hitachi¥Cm2NNMi¥

6. 一時ディレクトリおよび一時ファイルなどを削除する。

   NNMi が作成する次の一時ディレクトリおよび一時ファイルなどを削除します。

   存在できるものをすべて列挙しているため，存在しなくても問題ありません。なお，アンインストールのログ出力ファイル（NNMUninstall.log）は，必要に応じてコピーを取得してから削除してください。

```
%TEMP%¥HPOvInstaller¥
%TEMP%¥HPOvLic.log
%TEMP%¥HPOvPerlA-install.log
%TEMP%¥Install_Autopass.log
%TEMP%¥hsperfdata_Administrator
%TEMP%¥InstallerData
%TEMP%¥JP1NNMiMIBLoad.log
%TEMP%¥MicroFocusOvInstaller¥
%TEMP%¥NNMUninstall.log
%TEMP%¥NNM_X.X.X_HPOvInstaller.txt （Xには1桁以上の数字が入ります）
%TEMP%¥NNM_X.X.X_MicroFocusOvInstaller.txt （Xには1桁以上の数字が入ります）
%TEMP%¥nmscreatedb.log
%TEMP%¥nnm_hotfixes.log
%TEMP%¥nnm_installconfig_vbs.log
%TEMP%¥nnm_premigration.log
%TEMP%¥nnm_preinstallcheck_phaseI.log
%TEMP%¥nnm_preinstallcheck_phaseII.log
```

```
%TEMP%¥ovRemoveDir.exe
%TEMP%¥ovDetach.exe
%TEMP%¥ovinstallparams.ini
%TEMP%¥ovCleanUp.bat
%TEMP%¥persistent_state
%TEMP%¥preinstallcheck
%TEMP%¥JP1NNMiInstaller.log
%TEMP%¥JP1NNMiPostinstaller.log
%TEMP%¥InstallScript.iap_xml
%TEMP%¥nnm_preupgrade.log
%TEMP%¥nnm_pre_dialogcheck.log
%TEMP%¥OvLauncher.log
%TEMP%¥nnm_pre-uninstall.log
%TEMP%¥nnmilog¥
<drive>:¥ProgramData¥apregid.com.hpe
```

7. 環境変数を削除する。

   NNMi をアンインストールしても環境変数「OVCSL_LOG」,「OVCSL_LOG_APPLICATION」,「OVCSL_LOG_FILE」および NNMi インストール時に環境変数「PATH」に追加された「<*NnmInstallDir*>bin¥」は，削除されません。手動で削除してください。

   <*NnmInstallDir*>は環境変数「NnmInstallDir」に設定された値です。

2.4.2 NNMi をアンインストールする（Linux の場合）

1. NNMi をアンインストールするシステムに，root 権限を持つユーザーでログインする。
2. NNMi のサービスをすべて停止する。
3. NNMi のアンインストーラを起動する。

   次のコマンドを実行して，Hitachi PP Installer を起動します。

```
# /etc/hitachi_x64setup
```

4. 指示に従って NNMi のアンインストールを選択し，アンインストールを実行する。
5. アンインストールが終了したら，NNMi のインストールディレクトリとデータディレクトリを削除する。

   アンインストールを実行しても，NNMi のインストールディレクトリやデータディレクトリなどが削除されない場合は，手動で削除します。

   次のディレクトリを削除します。

   - インストールディレクトリ

```
/opt/OV
```

   - データディレクトリ

```
/var/opt/OV
```

6. 一時ディレクトリおよび一時ファイルなどを削除する。

   NNMi が作成する次の一時ディレクトリおよび一時ファイルなどを削除します。

存在できるものをすべて列挙しているため，存在しなくても問題ありません。なお，アンインストールのログ出力ファイル（NNMUninstall.log）は，必要に応じてコピーを取得してから削除してください。

```
/var/tmp/HPOvPerlA-install.log
/var/tmp/jp1nnmi
/var/tmp/JP1NNMiInstaller.log
/var/tmp/JP1NNMiPostinstaller.log
/var/tmp/rpm-tmp.xxx（xxxには1桁以上の英数字が入ります）
/tmp/install.dir.xxx（xxxには1桁以上の数字が入ります）
/tmp/ia_remove.shxxx.tmp（xxxには1桁以上の数字が入ります）
/tmp/HPOvInstaller
/tmp/MicroFocusOvInstaller
/tmp/NNMUninstall.log
/tmp/NNM_X.X.X_HPOvInstaller.txt（Xには1桁以上の数字が入ります）
/tmp/NNM_X.X.X_MicroFocusOvInstaller.txt（Xには1桁以上の数字が入ります）
/tmp/debug
/tmp/JP1NNMiMIBLoad.log
/tmp/hsperfdata_bin
/tmp/hsperfdata_nmsdbmgr
/tmp/hsperfdata_nmsproc
/tmp/hsperfdata_root
/tmp/nnm-premigration.log
/tmp/nnm_preinstallcheck_phaseI.log
/tmp/nnm_preinstallcheck_phaseII.log
/tmp/ovinstallparams.ini
/tmp/persistent_state
/tmp/postInstall
/tmp/postRemove
/tmp/preInstall
/tmp/preRemove
/tmp/preinstallcheck
/tmp/nnm-preupgrade.log
/tmp/nnm_pre_dialogcheck.log
/usr/share/apregid.com.hpe
```

2.5 インストールおよび初期スタートアップのトラブルシューティング

2.5.1 インストールの問題

(1) 問題：NNMi のインストールに，現在のホストシステム上の空き領域以上のディスクの容量が必要である（Linux の場合）

(a) 解決方法

Linux に NNMi をインストールする場合，バイナリをインストールする場所（$OV_INST_DIR）やデータファイルを保存する場所（$OV_DATA_DIR）を選択できません。初期設定でのこれらの場所は，次の設定となります。

- OV_INST_DIR=/opt/OV
- OV_DATA_DIR=/var/opt/OV

/opt/OV または/var/opt/OV のどちらかのディスク容量が十分でない場合は，下記の回避方法を使用して状況を改善してください。

1. 必要な場合は，NNMi をアンインストールする。
2. インストールターゲットから，バイナリをインストールしてデータファイルを保存するのに十分なディスク容量がある大きなパーティションへの，シンボリックリンクを作成する。

   シンボリックリンクを作成するための構文は次のとおりです。

```
ln -s <large disk> /opt/OV
ln -s <large disk> /var/opt/OV
```

   重要

   - インストール先の上位ディレクトリのアクセス権は 555 以上にしてください。

3. NNMi をインストールする。

(2) 問題：インストール時に，プレインストール手順（フェーズ II）に失敗し，/tmp/nnm_preinstall_phaseII.log ファイルで詳細を確認する必要があることを示すメッセージが表示される（Linux の場合）

(a) 解決方法

NNMi インストールスクリプトは，2 つのグループ（nmsgrp と nmsdb）と 2 つのユーザー（nmsproc と nmsdbmgr）および対応する$HOME ディレクトリを自動的に作成します。これらの操作は，次の理由によって失敗することがあります。

- IT 部門でuseradd またはgroupadd コマンドを無効にしたため，ユーザーおよびグループを作成できない。
- $HOME ディレクトリが NFS 上に存在する場合に，ルートユーザーが$HOME ディレクトリを作成できない。

NNMi インストーラがこれらのグループ，ユーザー，またはディレクトリの作成に失敗すると，インストールが中止されます。この場合は，ユーザーを手動で作成してインストールを再開できます。

1. nmsgrp グループで nmsproc ユーザーを作成する。
   $HOME ディレクトリを任意のディレクトリに設定しますが，そのディレクトリは存在する必要があります。
2. nmsdb グループで nmsdbmgr ユーザーを作成する。
   $HOME ディレクトリを任意のディレクトリに設定しますが，そのディレクトリは存在する必要があります。

これらの操作に失敗することがわかっていて，ユーザー ID，グループ ID，または$HOME の場所を制御する必要がある場合は，グループ，ユーザー，および$HOME ディレクトリを作成してからインストーラを起動できます。

useradd コマンドによってユーザーを作成した場合，デフォルトでは/home/<ユーザー名>がホームディレクトリになります。

2.5.2 初期スタートアップの問題

(1) 問題：NNMi コマンドラインツールを Linux の NNMi 管理サーバーで実行できない

(a) 解決方法

システム環境変数PATH に/opt/OV/bin が含まれていることを確認します。含まれていない場合は，システム環境変数PATH に/opt/OV/bin を追加します。

(2) 問題：JBoss ポートの競合

(a) 解決方法

デフォルトでは，JBoss アプリケーションサーバーは，NNMi との通信に複数のポートを使用します。通常これらのポートは，JBoss 以外のアプリケーションにも使用されます。

ポートの競合を解決するには，次の手順を実行します。

1. 管理者権限（Windows の場合）または root 権限（Linux の場合）のあるユーザーとして，テキストエディタで次のファイルを開く。
   - Windows：`%NnmDataDir%Conf¥nnm¥props¥nms-local.properties`
     `%NnmDataDir%`は，インストール時に指定するデータディレクトリです。
   - Linux：`/var/opt/OV/conf/nnm/props/nms-local.properties`
2. 既存のエントリを修正し，競合しているポート番号を使用できるポート番号に変更する。
3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop -c
ovstart -c
```

メモ

Windows の場合，［`ovstop`］と［`ovstart`］のコマンドは，［**スタート**］メニューからも実行できます。NNMi が使用するポートの詳細については，`nnm.ports` リファレンスページを参照してください。

(3) 問題：NNMi がノードを検出しない

(a) 解決方法

1. **ワークスペース**のナビゲーションパネルで［**設定**］ワークスペースから［検出］を開く。
2. ［**シード**］ビューを開く。
3. ［**検出シードの結果**］列の値を確認する。

   検出されたノードの大部分のステータスが，［**ノードが作成されました**］以外の場合は，NNMi の検出プロセスが正常に動作していなかったということです。

   ステータスが［**ノードが作成されました（非 SNMP デバイス）**］の場合は，ノードに対して`ping` が可能であるか，また，`nnmsnmpwalk.ovpl` コマンドを実行してノードから情報を取得できるかを確認します。詳細については，`nnmsnmpwalk.ovpl` リファレンスページを参照してください。これらのツールが実行できない場合は，次の事項を確認してください。

   a ノードに ping し，応答するか確認してください。

b　ノードで SNMP が有効になっているか確認してください。

c　ノードの SNMP エージェントのアクセスリストに，NNMi 管理サーバーが含まれていることを確認してください。

d　NNMi がノードを適切に検出できるよう，ノードの正しいコミュニティ文字列を設定していることを確認してください。この情報は，［通信の設定］フォームの［デフォルトの SNMPv1/v2 コミュニティ文字列］タブに表示されています。

e　ルータ，スイッチ，またはファイアウォールについて，検出を制限することのあるアクセス制御リストが設定されていないことを確認します。

詳細については，NNMi ヘルプの「検出を設定する」を参照してください。

(4)　問題：NNMi 管理サーバーにアクセスしていると，NNMi コンソールを起動できない（Windows の場合）

Web ブラウザで NNMi 管理サーバーをポイントしているときに，NNMi コンソールを起動できない場合，ファイアウォールが HTTP ポートをブロックしている可能性があります。この問題のトラブルシューティングを行うには，NNMi 管理サーバーでブラウザを実行します。このブラウザからは NNMi コンソールにアクセスでき，リモートのブラウザからはアクセスできない場合，ポートをチェックする必要があります。

この問題を解決するには，許可ポートリストに%NnmDataDir%Conf¥nnm¥props¥nms-local.properties ファイルに示されているnmsas.server.port.web.http 値を追加します。詳細については，nnm.ports リファレンスページを参照してください。

%NnmDataDir%は，インストール時に指定するデータディレクトリです。

(5)　問題：NNMi のインストールまたはアップグレードが正常に終了した後，NNMi コンソールが開かない

incidentActions.*.*.log ファイル（NNMi 管理サーバーの/var/opt/OV/log/nnm/public ディレクトリから入手可能）の最新のコピーに次のエラーメッセージも表示される。

SEVERE: com.hp.ov.nms.events.action.log.ActionLogger createActionServer:

java.io.FileNotFoundExceptionが原因で/var/opt/OV/tmp/actionServer.portからポート番号を取得できませんでした。

/var/opt/OV/tmp/actionServer.port.lock（権限拒否）: java.io.FileNotFoundException:

/var/opt/OV/tmp/actionServer.port.lock（権限拒否）

(a)　解決方法

1. NNMi 管理サーバーで次のコマンドを実行します。

   a. /opt/OV/bin/ovstop

b. `chown root:root /var/opt/OV/tmp`

c. `chmod 777 /var/opt/OV/tmp`

d. `chmod g+s /var/opt/OV/tmp`

2. `/var/opt/OV/tmp` ディレクトリで`actionServer.port` ファイルと`actionServer.port.lock` ファイルを探します。 それらのファイルが存在する場合は削除します。

3. NNMi 管理サーバーで次のコマンドを実行します。

   `/opt/OV/bin/ovstart`

3 NNMi 入門

ここでは，NNMi でネットワーク管理を始める上で必要な，NNMi へのアクセス方法，ネットワークを検出するための設定方法の概要について説明しています。オペレータおよび管理者用の詳細情報は，NNMi ヘルプに記載されています。

3.1　NNMi へアクセスする

NNMi をインストールし，インストール後の設定作業を完了すると，ネットワークの管理を開始できます。ネットワークのモニタリングやイベント処理のタスクについては，Web ブラウザのウィンドウで開く NNMi コンソールからアクセスできます。

NNMi コンソールにアクセスするには，次の手順に従います。

1. 対応 Web ブラウザを使用していることを確認する。

   「1.1　ハードウェアおよびソフトウェアを確認する」を参照してください。

2. Web ブラウザで JavaScript，NNMi 管理サーバーからのポップアップウィンドウを有効にし，ブラウザが NNMi 管理サーバーからの Cookie を受け入れるようにする。

   「1.5.3　NNMi コンソール用の Web ブラウザの有効化」を参照してください。

3. 次の URL を Web ブラウザのアドレス入力用のウィンドウに入力する。

   http://*<fully_qualified_domain_name>*:*<port>*/nnm/

   *<fully_qualified_domain_name>*は，NNMi 管理サーバーの完全修飾ドメイン名を表し，*<port>*は，JBoss アプリケーションサーバーが NNMi コンソールとの通信で使用するポートを表します。

   使用している NNMi 管理サーバーに複数のドメイン名がある場合は，NNMi では，インストール時にその中から 1 つを選択します。NNMi が使用している完全修飾ドメイン名を判断するには，`nnmofficialfqdn.ovpl` スクリプトを実行します。詳細については，`nnmofficialfqdn.ovpl` リファレンスページを参照してください。

   ブラウザで Windows オペレーティングシステムにインストールされている NNMi 管理サーバーを指定しても NNMi コンソールを起動できない場合，NNMi 管理サーバーで Windows ファイアウォールが http ポートをブロックしているおそれがあります。「(4)　問題：NNMi 管理サーバーにアクセスしていると，NNMi コンソールを起動できない（Windows の場合）」を参照してください。

4. NNMi サインイン用ウィンドウで，ユーザーのアカウント名とパスワードを入力したあと［**サインイン**］をクリックする。

   詳細については，次の「ユーザーのアカウントとロール」を参照してください。

ユーザーのアカウントとロール

インストール後の NNMi への初回アクセスのために，NNMi は特別の**システムアカウント**を提供します。通常は，このシステムアカウントは使用しないでください。

通常のご使用のために，NNMi 管理者は各ユーザー（またはユーザーのグループ）のアカウントを設定し，各アカウントに対し定義済みのユーザーロールを割り当てます。ユーザーロールによって，NNMi コンソールにアクセスできるユーザーと，各ユーザーが使用できるワークスペースとアクションが決まります。NNMi では，NNMi コンソールへのアクセスに対して次のユーザーロールが用意されています。これらのロールは，プログラムによってあらかじめ定義されていて修正はできません。

- 管理者
- オペレータレベル 2
- オペレータレベル 1

- ゲスト

チームの NNMi サインインのアクセス設定を行う前に，各チームのメンバに，どの定義済みの NNMi ロールを割り当てるのがふさわしいかを判断します。ロールは階層的です。つまり，階層内で高位のロールは下位のロールの特権をすべて含みます（管理者が最高位で，ゲストが最低位です）。

コマンドラインへのアクセスと同様，ユーザーのアカウントとロールは，NNMi コンソールで設定します。詳細については，NNMi ヘルプの「セキュリティの設定」を参照してください。

NNMi には，インストール時に作成された自己署名証明書を使用してそのまま使用できる https 設定があります。自己署名証明書の代わりに認証機関による署名入り証明書を使用する場合の詳細については，「10.　NNMi での証明書の使用」を参照してください。

3.2 NNMi ヘルプへアクセスする

NNMi ヘルプには，NNMi コンソールの使用方法が記載されています。

NNMi のヘルプにアクセスするには，NNMi コンソールメニューバーの［ヘルプ］をクリックし，メニューにある最初の区切りラインの上の項目の 1 つをクリックしてください。

メモ

NNMi コンソールには，情報入力フォームが含まれています。フォーム名は，ウィンドウの左上に表示されます。どの NNMi フォームからでも，フォームのヘルプ情報にアクセスできます。［ヘルプ］メニューで，［*<xyz>*フォームの使用法］（*<xyz>*は現在のフォームのタイトル）をクリックしてください。

3.3　ネットワーク検出を設定する

NNMi を使ってネットワークの検出や管理を開始するときは，テスト用ネットワークから始め，ごくわずかのインタフェースしか持たない少数のノードを検出，管理するように NNMi を設定することをお勧めします。**クイックスタート設定ウィザード**（「2.2　クイックスタート設定ウィザードを使用する」を参照）を使用すると，このような小さな構成が簡単に設定できます。NNMi のインストール直後は，**クイックスタート設定ウィザード**を使用することを推奨します。

NNMi の操作に慣れると，その豊富な機能がどのようにネットワークの管理に使われているのかを理解できるようになります。NNMi で管理するネットワークトポロジは，検出規則や管理領域を系統的に追加すれば，次第に拡張できます。

ここでは検出プロセスを開始する前に必要となる設定作業について，簡単に概要を説明します。次の表のチェックリストでは，これらの作業についてまとめてあります。

表 3-1　検出設定チェックリスト

チェック欄（はい/いいえ）	タスク
	検出するノードのすべてがネットワークに接続され，それに対応している SNMP のバージョン（SNMPv1，SNMPv2c，または SNMPv3）の設定がされているかを検証します。
	ネットワーク管理者より，管理するノードの読み取り専用コミュニティ文字列を入手します。
	NNMi コンソールを使用して，「3.3.1　コミュニティ文字列を設定する」に記載されている手順でコミュニティ文字列を設定します。
	NNMi コンソールを使用して，「3.3.2　自動検出ルールを設定する」に記載されている手順でスパイラル検出プロセスを設定します。
	NNMi コンソールを使用して，「3.3.3　検出の進行状況を確認する」に記載されている手順でスパイラル検出プロセスをチェックします。

検出プロセスの詳細については，NNMi ヘルプの「ネットワークの検出」を参照してください。

3.3.1　コミュニティ文字列を設定する

コミュニティ文字列を使用して NNMi を設定するには，次の手順に従います。

1. ワークスペースのナビゲーションパネルで，**［設定］** ワークスペースを選択する。
2. 次のように，**［通信の設定］** フォームを開く。

3. ［デフォルトの SNMPv1/v2 コミュニティ文字列］タブで，［新規作成］アイコンをクリックする。
4. ［デフォルトの読み取りコミュニティ文字列］フォーム上の，［読み取りコミュニティ文字列］のボックスに，検出範囲内の特定のノードのコミュニティ文字列を入力して，［保存して新規作成］アイコンをクリックする。
5. 手順 4 を繰り返し実行し，検出範囲内のノードのコミュニティ文字列をすべて入力してから，［保存して閉じる］アイコンをクリックする。
6. ［通信の設定］フォームで，［保存して閉じる］アイコンをクリックする。

デバイスのコミュニティ文字列の設定やファイルからのコミュニティ文字列のロードの詳細については，NNMi ヘルプの「通信プロトコルの設定」を参照してください。

3.3.2　自動検出ルールを設定する

ネットワーク管理で最も重要な作業の 1 つは，常に最新のネットワークトポロジを把握しておくことです。NNMi は，ネットワークノードの継続**検出**によってこのトポロジを維持します。NNMi の検出プロセスは，根本原因解析やトラブルシューティングのツールが，インシデント解決のための正確な情報を提供することを保証します（メモの「ネットワーク検出」を参照）。

自動検出ルールを設定するには，次の手順に従います。

1. ワークスペースのナビゲーションパネルで［設定］ワークスペースから［検出］を開く。
2. ［検出の設定］フォームを開く。

3. ［自動検出ルール］タブをクリックし，次に［新規作成］アイコンをクリックする。
4. ［自動検出ルール］フォームの［基本］に，ルールの名前および順序の情報を入力する。
   この順序は，ほかの自動検出ルールに対するこのルールの優先度を示す数値です。詳細については，［ヘルプ］＞［自動検出ルール フォームの使用法］の順にクリックします。
5. ［このルールの自動検出開始ポイント］で，この規則に対する適切な自動検出アクションを選択する。
6. ［このルールの IP アドレス範囲］で，［新規作成］アイコンをクリックする。
7. ［IP の自動検出範囲］フォームで，［IP の範囲］を入力し，［範囲のタイプ］は［ルールに含める］という設定のままにして，［保存して閉じる］アイコンをクリックする。
8. ［自動検出のルール］フォームで，［保存して閉じる］アイコンをクリックする。
9. 手順 3 から手順 8 までを繰り返し実行し，使用するすべてのルールを追加する。
10. ［検出の設定］フォームで，［保存して閉じる］アイコンをクリックし，すべての新しい自動検出ルールを NNMi データベースに保存する。
11. ［設定］ワークスペースから［検出］を開き，［シード］をクリックする。
12. ［新規作成］アイコンをクリックする。
13. ［検出シード］のフォームで，ホスト名または IP アドレスを入力し，［保存して閉じる］アイコンをクリックする。
14. 手順 12 および手順 13 を繰り返して，検出シード用のすべてのホスト名または IP アドレスを追加する。

検出の進行状況を監視する方法は，「3.3.3　検出の進行状況を確認する」を参照してください。

検出の設定の詳細については，NNMi ヘルプの「検出を設定する」を参照してください。

メモ

ネットワーク検出

NNMi は，ネットワークにあるデバイス（スイッチやルータなど）に関する情報を収集したり，ユーザーやチームにとって重要なデバイスの管理を積極的に行ったりします。検出モードは，次の 2 つから選ぶことができます。

- **検出シード**：ユーザーが，デバイスのリストを提供して，NNMi の検出やモニタリングの対象となるデバイスを包括的に管理します。
- **自動検出ルール**：ユーザーが，検出シードとなるアドレスやホスト名のリストを提供し，NNMi は，この情報を包括的な自動検出用の開始ポイントとして使用します。さらに，ユーザーは，IPv4 アドレス範囲や MIB II sysObjectID を提供すれば，NNMi の検出プロセスに制限がかけられます。

検出モードの選択が済むと，**NNMi スパイラル検出**を行います。NNMi は，さまざまなプロトコルや技術を利用して，ネットワークインベントリについての豊富な情報を収集し，デバイス（サブネットや VLAN）間の関係を確認し，デバイス間の接続関係を正確に描き出します。

NNMi Causal Engine は，各デバイス（および，デバイスに関連する各インタフェースやアドレス）の現在のステータスを判定し，発生した問題や潜在的な問題を検出した場合には，積極的に通知を行います。

ダイナミック検出プロセスは，長期的に継続されます。ネットワーク管理ドメインの中で変更があった場合は，NNMi スパイラル検出が自動的に情報を更新します。

ネットワーク検出の詳細については，NNMi ヘルプの「ネットワークの検出」を参照してください。

3.3.3 検出の進行状況を確認する

スパイラル検出プロセスの起動後，そのプロセスが正しく実行されているか検証します。

メモ

スパイラル検出は動的であるため，NNMi は継続的にネットワークノードを検出します。NNMi は，検出ルールに新しいノードが追加されるたびに，そのノードを検出し，ノードに関するトポロジ情報を収集し，ノードのモニタリングを開始します。

検出の進行状況の測定には幾つかの方法があります。検出の進行状況を調べるには，次のどれかの処理を実行します。

- 検出中に，［設定］＞［検出］＞［シード］の順にクリックして，シードのステータスをチェックします。［検出シードの結果］列のステータス情報を確認します。検出が終わりに近づくと，ノードの大半が「ノードが作成されました」のステータスになります。
- 検出中に，［ヘルプ］＞［システム情報］の順にクリックして，［データベース］タブから検出の進行状況を確認します。［データベースのオブジェクト数］を 1 時間に数回確認します。ノード，SNMP エージェント，インタフェース，IP アドレス，L2 接続のフィールドの数は，やがて一定になります。サンプリング周期を通して，この数字の増加がなければ，検出は完了です。
- 検出中に，NNMi コンソールで，［インベントリ］ワークスペースから［ノード］を選択します。［合計］フィールドの値を 1 時間に数回確認します。サンプリング周期を通して，この値が増加していなければ検出は完了です。
- 検出中に，NNMi コンソールで［ツール］＞［NNMi セルフモニタリングのグラフ］＞［検出の進行状況］をクリックして，検出の進行状況を確認します。
- 検出中に，NNMi コンソールで［ツール］＞［ステータス分布グラフ］＞［ノードステータス］をクリックして，検出の進行状況を確認します。
- 検出中に，NNMi コンソールの［トポロジマップ］ワークスペースの［ネットワークの概要］をクリックします。マップの複雑性の成長を 1 時間監視します。マップの成長が鈍化し，サンプリング周期を通してこの成長が止まれば，検出は完了です。

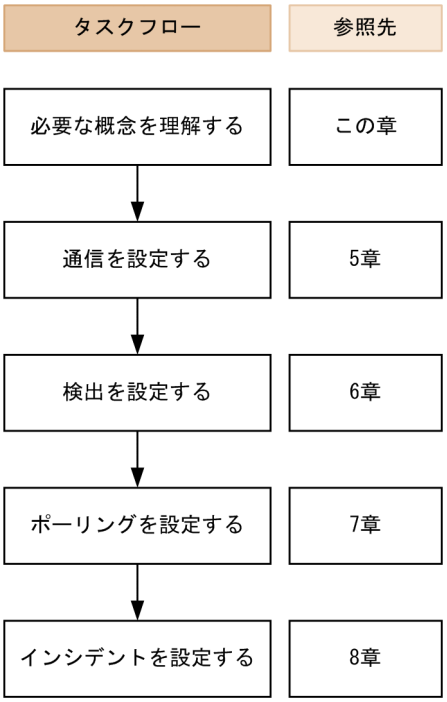
メモ

検出で問題が発生する場合は，「(3)　問題：NNMi がノードを検出しない」を参照してください。

4 設定の一般概念

この章では設定の概念を説明しています。詳細については，このマニュアルの 5 章以降で説明しています。この章では，すべての NNMi 設定領域に適用されるベストプラクティスについても記載しています。

4.1　タスクフローモデル



このマニュアルの設定編では，次のタスクフローに役立つ情報を記載しています。

1. **概念**－設定領域の概略を理解できます。このマニュアルの情報は，NNMi ヘルプの情報を補足しています。
2. **計画**－設定にどのように取り組むかを決定します。これは，会社のネットワーク管理の文書化を開始または更新する良い機会です。
3. **設定**－NNMi コンソール，設定ファイル，コマンドラインインタフェースの組み合わせを使用して，NNMi に設定します。具体的な手順については，NNMi ヘルプを参照してください。
4. **評価**－NNMi コンソールで，設定結果を確認します。設定を最適なものにするために，必要に応じて調整します。
5. **調整**－（任意）設定を調整して，NNMi のパフォーマンスを向上します。

4.2　ベストプラクティス：既存の設定を保存する

大きな設定変更を行う前には，既存の設定内容のコピーを保存しておくことをお勧めします。設定内容を元に戻したい場合に，簡単に戻すことができます。

nnmconfigexport.ovpl コマンドを使用して，現在の設定内容を保存します。保存した設定内容を復元するには，nnmconfigimport.ovpl コマンドを使用します。

これらのコマンドの使用方法の詳細については，該当するリファレンスページを参照してください。

nnmconfigexport.ovpl コマンドでは SNMPv3 資格情報は保持されません。詳細については，nnmconfigexport.ovpl コマンドのリファレンスページを参照してください。

4.3　ベストプラクティス：作成者属性を使用する

多くの NNMi 設定フォームには，**作成者**属性が含まれています。

これらのフォーム上で設定を作成，または変更する場合，［作成者］属性に作成者の組織を識別する値を設定してください。NNMi 設定をエクスポートするときに，作成者値を指定して作成者の組織がカスタマイズした項目だけを引き出すことができます。

NNMi をアップグレードする際，作成者の属性値が，ユーザーが作成した作成者になっている設定は上書きされません。

4.4 ユーザーインタフェースモデル

NNMi コンソールフォームの一部では，データベースの更新にトランザクションアプローチが使用されます。NNMi コンソールのフォームで行った変更は，フォームを保存して閉じる操作が NNMi コンソールで行われないと有効になりません。保存されていない変更が含まれるフォームを閉じると，NNMi によって保存されていない変更があるため，終了を続行するか確認するメッセージが表示されます。

4.5 順序

幾つかの NNMi コンソール設定フォームには，設定を適用する優先順位を設定する**順序**属性が含まれています。ある設定領域で，NNMi は設定内容に対して各項目を，順序番号が最も小さい（低い）ものから大きいものへの順に，NNMi が一致するまで評価し続けます。一致した時点で，NNMi は一致する設定の情報を使用し，これ以上探すのをやめます（通信設定は例外です。NNMi は通信設定を完了するために，そのほかのレベルで情報の検索を続行します）。

順序属性は，NNMi の設定で重要な役割を果たします。予想外の検出結果やステータス結果が出た場合は，その領域の設定の順序を確認してください。

順序番号は次の個所でも使用されますが，その意味は異なります。

- メニューおよびメニュー項目の順序は，関連するメニューのローカルコンテキスト内の項目の順序を設定します。
- **［ノードグループマップの設定］**フォームのトポロジマップ順序で，**［トポロジマップ］**ワークスペースの項目の順序が設定されます。

順序属性が指定の設定領域にどのように影響するかの情報については，その領域の NNMi ヘルプを参照してください。

ヒント

- 各設定領域で，小さい順序番号は最も限定的な設定に適用し，大きな順序番号は限定度の低い設定に適用します。
- 各設定領域で，すべての順序番号を一意にしてください。初期設定時は，通常の間隔の順序番号を使用して，将来設定を変更できるような柔軟性を確保しておいてください。例えば，1 番目から 3 番目の設定には 100，200，300 の順序番号を付けます。

4.6　ノードグループおよびインタフェースグループ

ノードグループやインタフェースグループに対し，ビューに表示する内容を絞り込むためのフィルタを設定できます。ノードグループに「重要な Cisco ルーター」を設定した場合を例にとると，「重要な Cisco ルーター」だけをフィルタに設定すれば，目的のルーターだけをビューに表示できます。

ノードグループは，次のどれか，またはすべての目的に使用できます。

- モニタリングの設定
- インシデントペイロードのフィルタリング
- テーブルフィルタリング
- マップビューのカスタマイズ
- グローバルネットワーク管理機能のリージョナルマネージャーからグローバルマネージャーに渡されたノードのフィルタリング

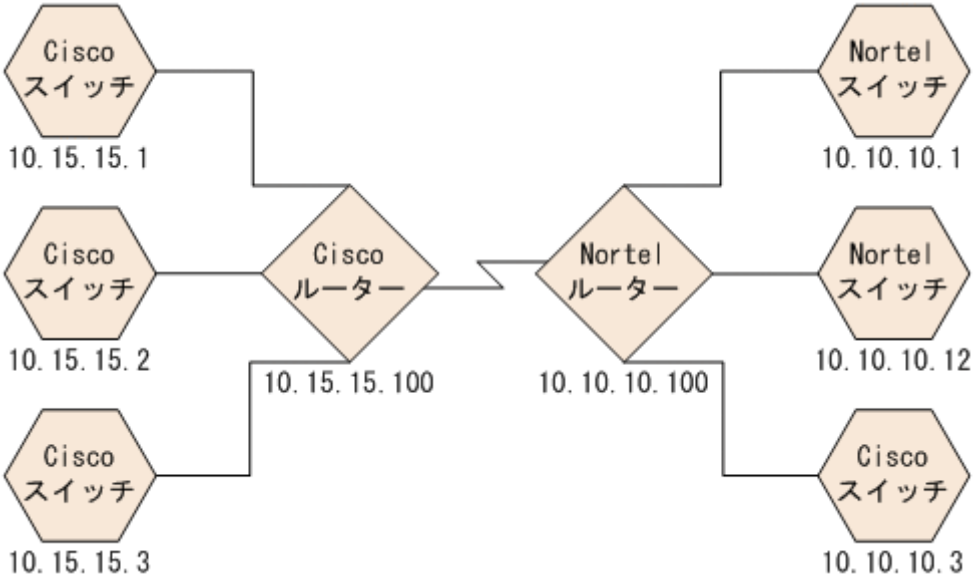
インタフェースグループは，次のどれか，またはすべての目的に使用できます。

- 検出からのインタフェース除外
- モニタリングの設定
- インシデントペイロードのフィルタリング
- テーブルフィルタリング

任意のフィルタリング可能な属性に基づきノードグループの階層を作成し，マップビューのドリルダウン，監視，またはその両方の設定の継承を管理できます。

4.6.1　グループの重複

グループ定義をどのように使用するかに関係なく，最初のステップでは，どのノードまたはインタフェースをグループのメンバーにするかを定義します。さまざまな目的でグループが作成されるため，それぞれの対象が複数のグループに含まれる可能性があります。次の例を考えてみます。



- 監視を目的とした場合，ベンダーや場所を問わずすべてのスイッチに 3 分間のポーリング間隔を設定するのがよいでしょう。この場合は，デバイスカテゴリフィルタを使用します。
- 保守を目的とした場合は，すべての Cisco スイッチを 1 つのグループにして，IOS のアップグレードのときに，このグループをまとめてサービス停止にできるようにするのがよいでしょう。この場合は，ベンダーフィルタを使用します。
- 可視化の場合は，10.10.*.*サイト上のすべてのデバイスを，ステータスを反映したコンテナにグループ化するのがよいでしょう。この場合は，IP アドレスフィルタを使用します。

IP アドレスが 10.10.10.3 の Cisco スイッチはこの 3 つのグループすべてに適しています。

設定や表示に便利なようにグループセットを豊富にするのもよいですが，使用されることのない必要以上のエントリを一覧に詰め込み過ぎることのないよう，バランスをとってください。

4.6.2 ノードグループのメンバーシップ

NNMi は，設定されたノードグループと検出したノードを比較して，ノードグループのメンバーシップを判断します。

- **[追加のノード]** タブで指定したノードは，すべてそのノードグループのメンバーです。

> **メモ**
>
> NNMi 管理サーバーのリソースを大きく消費するため，**[追加のノード]** タブを使用したノードグループへのノード追加は極力避けてください。

- **[子ノードグループ]** タブで指定した少なくとも 1 つのノードグループのメンバーになっているノードは，すべてそのノードグループのメンバーです。

- ［デバイスフィルター］タブの 1 つ以上のエントリ（存在する場合），および［追加のフィルター］タブで指定したフィルタに一致するノードは，すべてそのノードグループのメンバーです。

(1) 階層/包含

単純で再利用可能な小さいグループを作成し，これらを監視や可視化のために階層的に組み合わせることができます。階層的なノードのコンテナを使用すると，障害時にオブジェクトの場所やタイプに関する手がかりが得られるような，より良いマップビューを作成できます。NNMi によって，グループの定義とそのドリルダウンの順序を完全にコントロールできます。

単純で再利用可能な小さいグループを最初に作成し，そのあとでより大きなグループを作成するときに，これらを子グループとして指定します。また，最初にいちばん大きな親グループを指定し，それから子グループを作成していくこともできます。

例えば，ネットワークが Cisco スイッチ，Cisco ルーター，Nortel スイッチ，Nortel ルーターで構成されているとします。Cisco デバイスの親グループとすべてのスイッチの親グループを作成できます。親を作成してその子を指定するときに階層が定義されるので，Cisco スイッチのようなそれぞれの子グループには複数の親ができる可能性があります。

階層は，次の状況で使用すると効果的です。

- 監視ニーズが類似したノードのタイプ
- ノードの地理的な配置
- まとめてサービスを停止にするノードのタイプ
- オペレータの職務別によるノードのグループ

マップビューおよびテーブルビューでグループを使用すると，伝播された（設定可能な）グループのステータスが表示されます。

メモ

グループ定義を使用して監視設定を指定する際に，階層は設定の順序を示すものではないことを留意してください。小さい順序番号の設定は，ノードに適用されます。順序番号を注意深く増やすことで，設定の継承概念を真似ることができます。

子ノードグループに循環参照となるノードグループを設定して保存すると，警告が表示され，保存に失敗します。

(2) デバイスフィルター

検出中，NNMi は直接情報を SNMP クエリーで収集し，そこからほかの情報を，デバイスプロファイルを通じて導き出します。詳細については，「6.1.1 デバイスプロファイルとデバイスの属性」を参照してください。システムオブジェクト ID を収集することによって，NNMi は該当するデバイスプロファイルを検索し，次の情報を導き出します。

- ベンダー
- デバイスカテゴリ
- カテゴリ内のデバイスファミリ
- デバイスのプロファイル

導き出されたこれらの値は，デバイスプロファイルそのものとともに，フィルタとして使用できます。

例えば，あるベンダー製のすべての対象物を，デバイスタイプやファミリに関係なくグループ化できます。また，ある種類のデバイス（例えばルーター）をすべて，ベンダーを問わずにまとめることができます。

(3) 追加のフィルター

追加のフィルターエディタを使用すると，次のようなフィールドに一致するカスタム論理を作成できます。

- hostname（ホスト名）
- mgmtIPAddress（管理アドレス）
- hostedIPAddress（アドレス）
- sysName（システム名）
- sysLocation（システムのロケーション）
- sysContact（システムの連絡先）
- capability（ケーパビリティの一意キー）
- customAttrName（カスタム属性名）
- customAttrValue（カスタム属性値）
- isSnmpNode（エージェント有効）
- isNnmSystemLocal（NNMi 管理サーバー）
- sysOidNode（システムオブジェクト ID）
- devCategoryNode（デバイスのカテゴリ）
- devVendorNode（デバイスのベンダー）
- devFamilyNode（デバイスのファミリ）
- nnmSystemName（ホスト名，大文字と小文字を区別）
- nodeName（ノード名）
- securityGroupName（セキュリティグループ名）
- securityGroupUuid（セキュリティグループの UUID）
- tenantName（テナント名）
- tenantUuid（テナントの UUID）

フィルタには，AND，OR，NOT，EXISTS，NOT EXISTS，およびグループ化（括弧）操作を含めることができます。詳細については，NNMi ヘルプの「ノードグループの追加のフィルターを指定する」を参照してください。

ケーパビリティは，すでに検出されたデバイスからノード詳細を調べることによって，確認できます。

(4) 追加のノード

ノードグループに対してノードを限定するには，［追加のフィルター］を使用することをお勧めします。フィルタを使用して指定することが難しい重要なデバイスがネットワークに含まれている場合，それらのデバイスは個々のホスト名でグループに追加できます。ホスト名ごとにノードをノードグループに追加するのは，ほかに手段がない場合だけにしてください。

メモ

NNMi 管理サーバーの使用リソースが増加するため，［追加のノード］タブを使用してノードグループにノードを追加することはほとんどありません。

4.6.3 ノードグループのステータス

次のどちらかのアルゴリズムを使用して NNMi によってノードグループのステータスが決定されます。

- ノードグループの任意のノードの最も重大なステータスと一致するようにノードグループを設定します。このアプローチを使用するには，［ステータスの設定］フォームの［最も重大なステータスを伝達］チェックボックスをオンにします。
- 各ターゲットステータスに設定されたしきい値を使用してノードグループのステータスを設定します。例えば，警戒域のターゲットステータスのデフォルトしきい値は 20%です。NNMi では，ノードグループ内のノードの 20%（または，それ以上）が警戒域ステータスになると，ノードグループのステータスが警戒域に設定されます。このアプローチを使用するには，［ステータスの設定］フォームの［最も重大なステータスを伝達］チェックボックスをオフにします。ターゲットしきい値のパーセントしきい値は，このフォームの［ノードグループのステータス設定］タブで変更できます。

大きなノードグループのステータス計算には大量のリソースが必要になるため，新規インストール時にはノードグループのステータス計算は NNMi のデフォルトでオフに設定されます。ステータスの計算は，各ノードグループの［ノードグループ］フォームの［ステータスの計算］チェックボックスで有効にできます。

4.6.4 インタフェースグループ

インタフェースグループは，ノード内のインタフェースを，ifType 別に，または ifAlias，ifDesc，ifName，ifIndex，IP アドレスなどほかの属性別にフィルタリングします。インタフェースグループは階

層も包含もありませんが，インタフェースをホストしているノードのノードグループに基づいてメンバーシップをさらに限定できます。

インタフェースグループを，ノードグループと同様のカスタムケーパビリティおよび属性でフィルタリングできます。

インタフェースグループの制限は，タブ内およびタブ間でまとめて AND を適用します。

重要

インタフェースグループのインタフェースは，次の条件での検出中に必ずしも最初から除外されるわけではありません。

- インタフェースグループは，インタフェースグループ定義で 1 つ以上のインタフェース機能をフィルタリングして作成されます。
- インタフェースグループは，［除外対象インタフェース］検出の設定オプションで指定されます。インタフェース機能はインタフェースグループのインタフェースに適用されたあとに，再検出中に除外フィルタが再適用されると除外されます。

NNMi で提供されるインタフェース機能と［除外対象インタフェース］検出の設定オプションの詳細については，NNMi ヘルプ「管理」を参照してください。

4.7 ノード／インタフェース／アドレス階層

NNMi はモニタリングの設定を，次のように適用します。

1. **インタフェースの設定**－NNMi は，最初に一致した**インタフェースの設定**定義に基づき，各ノードのインタフェースと IP アドレスに一致するものがないか，照合する。

   照合するときに最初に適用されるのは，順序番号が最も小さい**インタフェースの設定**定義です。

2. **ノードの設定**－1.の処理で一致しなかった各インタフェースまたは IP アドレスは，**ノードの設定**定義に基づき照合される。

   このとき，最初に適用されるのは，順序番号が最も小さい**ノードの設定**定義です。

   > **メモ**
   >
   > 子ノードグループは，順序階層に含まれます。親ノードグループの順序番号のほうが小さい場合（例えば，親=10，子=20），親ノードグループに指定された監視設定は子ノードグループ内のノードにも適用されます。親ノードグループ監視設定を上書きするには，子ノードグループの順序番号を親よりも小さな番号に設定します（例えば，親=20，子=10）。

3. **デフォルト設定**－1.または 2.の照合でノード，インタフェース，または IP アドレスが一致しなかった対象については，デフォルトの監視設定が適用される。

4.8 NNMi 設定およびデータベースのリセット

検出を完全に再スタートして NNMi 設定をすべてやり直したい場合，または NNMi データベースが破損した場合は，NNMi 設定およびデータベースをリセットできます。このプロセスで，NNMi 設定，トポロジ，およびインシデントのすべてが削除されます。

次の手順で説明しているコマンドの詳細は，該当するリファレンスページを参照してください。

次の手順に従ってください。

1. （任意）現在の NNMi 設定をとっておきたい場合は，`nnmconfigexport.ovpl` コマンドを使用して NNMi 設定を XML ファイルに出力する。

   `nnmconfigexport.ovpl` コマンドでは SNMPv3 資格情報は保持されません。詳細については，`nnmconfigexport.ovpl` コマンドのリファレンスページを参照してください。

2. （任意）`nnmtrimincidents.ovpl` コマンドを使用して，NNMi インシデントをアーカイブする。

   `nnmtrimincidents.ovpl` コマンドのデフォルトではインシデントはアーカイブされないため，`-archiveOnly` オプションを付与して実行します。詳細については，`nnmtrimincidents.ovpl` コマンドのリファレンスページを参照してください。

3. NNMi サービスを，次のコマンドを使用して停止する。

```
ovstop -c
```

4. （任意）既存のデータベースをバックアップする。

   この手順によってデータベースが削除されます。実行する前に次のコマンドで既存のデータベースをバックアップしておくことをお勧めします。

```
nnmbackup.ovpl -type offline -target <backup_directory>
```

5. NNMi データベースを削除して再作成する。

```
nnmresetembdb.ovpl -nostart
```

6. NNMi サービスを，次のコマンドを使用して開始する。

```
ovstart -c
```

   これで，NNMi を新しいシステムにインストールしたときと同じ，デフォルト設定だけの状態となります。

7. 次のどれかの方法で，NNMi の設定を開始する。
   - 「クイックスタート設定ウィザード」を使用する。
   - NNMi コンソールの［設定］ワークスペースで情報を入力する。
   - `nnmconfigimport.ovpl` コマンドを使用して，手順 1.で保存した NNMi 設定の一部またはすべてをインポートする。

重要

nnmconfigimport.ovpl コマンドを使用して大量の設定をインポートする場合（9,500 個のノードグループや 10,000 個のインシデントの設定など），-timeout オプションを使用して，インポートトランザクションのタイムアウトをデフォルト値の 60 分（3,600 秒）よりも長くなるように調整することを検討してください。詳細については，nnmconfigimport.ovpl コマンドのリファレンスページを参照してください。

4.9 NNMi 設定ファイルのモデルファイル

NNMi のインストール時の設定ファイルの内容は，モデルファイルとして以下のフォルダにコピーされます。

- Windows の場合：`%NnmInstallDir%support¥JP1Model_13-00`
- Linux の場合：`/opt/OV/support/JP1Model_13-00`

モデルファイルには，「.model」という拡張子が付いています。

設定ファイルをカスタマイズしたあと，デフォルトの設定に戻したい場合は，モデルファイルをコピーして，拡張子の「.model」を削除してください。

メモ

- NNMi のモデルファイルでは，新規インストールの場合は初期設定値が保存されますが，バージョンアップの場合，バージョンアップ前に変更した設定はそのままになります。
- NNMi のサポートする server.properties は 2 種類あるため，以下のようにコピーします。

ファイルパス	コピー名
• Windows の場合：`%NnmDataDir%nmsas¥NNM¥server.properties` • Linux の場合：`/var/opt/OV/nmsas/NNM/server.properties`	server.properties.NNM.model
• Windows の場合：`%NnmDataDir%nmsas¥nms¥server.properties` • Linux の場合：`/var/opt/OV/nmsas/nms/server.properties`	server.properties.nms.model

5 NNMi 通信

NNMi は，Simple Network Management Protocol（SNMP）と Internet Control Message Protocol（ICMP ping）の両方のプロトコルを使用して，デバイスを検出し，デバイスのステータスと稼働状態を監視します。お使いの環境で実行可能な通信を確立するには，ネットワークのさまざまなデバイスとエリアについて，アクセス認証，適切なタイムアウトと再試行回数を NNMi に設定します。トラフィックを削減するためやファイアウォールを考慮するために，ネットワークの幾つかの領域でプロトコルを無効にできます。通信の設定の値は，NNMi の検出およびステータスポーリングの基礎となります。NNMi は，検出またはポーリングのクエリーを作成するときに，各デバイスに該当する値を適用します。そのため，ネットワークの幾つかの領域との SNMP 通信を無効にするよう NNMi を設定すると，NNMi 検出と NNMi 状態ポーリングはどちらも，SNMP 要求をその領域には送信できません。

5.1　通信の概念

NNMi は，主に要求と応答の方式で SNMP と ICMP を使います。ICMP ping 要求への応答で，アドレスの応答性を確認します。特定の MIB オブジェクトに対する SNMP 要求への応答で，ノードに関するより総合的な情報を取得します。

> **メモ**
>
> （SNMP エージェントに加えて）Web エージェントが設定されている場合，NNMi は追加のプロトコルを使用できます。例えば，VMware 環境用の SOAP プロトコルなどです。

次の概念が NNMi 通信設定に適用されます。

- 通信の設定レベル
- ネットワーク待ち時間とタイムアウト
- SNMP アクセス制御
- SNMP バージョンの優先
- 管理アドレスの優先
- ポーリングプロトコル
- `nnmsnmp*.ovpl` コマンドの動作

5.1.1　通信の設定レベル

NNMi 通信設定には，次のレベルがあります。

- 特定のノード
- 領域
- グローバルなデフォルト

各レベルで，アクセスの資格情報，タイムアウトと再試行の値，管理プロトコルの有効化（ICMP や SNMP など），管理プロトコルのアクセス設定（SNMP など）を指定できます。あるレベルで設定をブランクにしておくと，NNMi は次のレベルのデフォルトを適用します。

指定ノードと通信するとき，NNMi は設定を次のように適用します。

1. ノードが特定のノードの設定と一致する場合，NNMi はその設定に含まれている通信の値をすべて利用する。
2. 1.の特定ノードの設定に当てはまるものがなければ，NNMi はノードがどの領域に属するか判断する。

領域は重なる可能性があるため，NNMi では順序番号が最小のものと一致する領域が使用されます。NNMi は，その領域に対して指定された値を，設定が一致したノードに適用します。領域の設定が一致した場合，それ以降の順序番号の大きな領域設定は使用されません。

3. 1.と 2.に当てはまる設定がなければ，NNMi はグローバルなデフォルト設定を使用して，残りの空白の設定に取り込む。

特定のデバイスとの管理プロトコル通信に使用される値は，必要な設定がすべて決まるまで，累積的に構築されます。

5.1.2 ネットワーク待ち時間とタイムアウト

通常のネットワーク遅延は，NNMi 管理サーバーが ICMP クエリーと SNMP クエリーへの応答を得るための待ち時間に影響を与えます。一般に，ネットワークのエリアが異なれば，応答が返る時間も異なります。例えば，NNMi 管理サーバーが置かれているローカルネットワークからは，ほぼ即時の応答が返り，ダイヤルアップワイドエリアリンク経由でアクセスする遠隔地にあるデバイスからの応答は，通常はるかに長く時間が掛かります。

さらに，負荷が大きいデバイスは処理量が多いため ICMP クエリーまたは SNMP クエリーにただちに応答できません。タイムアウトと再試行の設定を決定するときには，こうした遅延に関する事項を考慮してください。

ネットワーク領域と特定のデバイスの両方について，固有のタイムアウトと再試行の設定を行うことができます。設定によって，応答がない場合に要求を破棄するまでの，NNMi の応答待ち時間，NNMi がデータを要求する回数が決まります。

要求を再試行するたびに，NNMi は設定したタイムアウト値をそれまでのタイムアウト値に加算します。そのため，再試行するごとに停止時間が長くなります。例えば，NNMi の設定を 5 秒でタイムアウト，再試行は 3 回とすると，NNMi は最初の要求への応答を 5 秒待ちます。応答がない場合は再試行 1 回目の要求への応答は 10 秒待ち，2 回目の要求への応答は 15 秒待ち，3 回目の要求の応答は 20 秒待ってから次のポーリングサイクルに移ります。

5.1.3 SNMP アクセス制御

管理対象デバイス上の SNMP エージェントとの通信には，アクセス制御資格情報が必要です。

- SNMPv1 と SNMPv2c

  各 NNMi 要求内のコミュニティ文字列は，応答する SNMP エージェントで設定されているコミュニティ文字列と一致する必要があります。通信はすべて，クリアテキスト（暗号化なし）でネットワークを通過します。

- SNMPv3

SNMP エージェントとの通信は，ユーザーベースのセキュリティモデル（USM）に従います。各 SNMP エージェントには，設定済みのユーザー名とそれに関連する認証要件のリストがあります（認証プロファイル）。すべての通信のフォーマットは，設定によって制御されます。NNMi SNMP 要求は，有効なユーザーを指定し，そのユーザーに対して設定されている認証とプライバシの制御に従う必要があります。

- 認証プロトコルは，メッセージ認証を使用しないか，HMAC-MD5-96，HMAC-SHA-1，HMAC128-SHA-224，HMAC192-SHA-256，HMAC256-SHA-384，または HMAC384-SHA-512 のどれかを選択したものの，ハッシュベースのメッセージ認証コードを使用します。
- プライバシプロトコルは，暗号化を使用しないか，DES-CBC，TripleDES，AES-128，AES-192 または AES-256 のどれかを選択したものの，対称暗号化プロトコルを使用します。

  DES-CBC は弱い暗号と考えられています。そのため，DES-CBC を使用する場合は，より強い暗号を選択することをお勧めします。NNMi が管理するノードで SNMPv3 通信を設定する場合は，DES-CBC の使用はお勧めしません。

暗号の選択を変更する場合は，次の手順で実施します。

1. NNMi コンソールから，［**設定**］ワークスペースをクリックする。
2. ［**インシデント**］フォルダを展開する。
3. ［**トラップサーバー**］フォルダを展開する。
4. ［**トラップ転送設定..**］をクリックする。
5. ［**プライバシプロトコル**］リストで，より強い暗号を選択する。

NNMi は，（IP アドレスフィルタやホスト名フィルタ経由で定義された）ネットワークの領域のマルチ SNMP アクセス制御資格情報の仕様をサポートします。NNMi は，設定したすべての値を，所定の SNMP セキュリティレベルで並行して試し，その領域内のデバイスと通信しようとします。NNMi がその領域で使用する最小限の SNMP セキュリティレベルを指定できます。NNMi は，各ノードから返される最初の値（デバイスの SNMP エージェントからの応答）を検出と監視の目的で使用します。

デフォルトの HA 環境では，SNMP ソースアドレスは物理クラスタノードアドレスに設定されます。SNMP ソースアドレスを NNM_INTERFACE（仮想 IP アドレスに設定される）に設定するには，ov.conf ファイルを編集して，IGNORE_NNM_IF_FOR_SNMP の値を OFF に設定する必要があります（デフォルトでは，これは ON に設定されます）。

5.1.4 SNMP バージョンの優先

SNMP プロトコルはバージョン 1 からバージョン 2（c）へと長年をかけて発展したもので，現在はバージョン 3 です。この間，とりわけセキュリティ機能は強化されてきました。NNMi は，どのバージョンでも処理できますし，全バージョンが混在した環境でも処理できます。

NNMi が特定のノードについて受信する最初の SNMP 応答によって，そのノードとの通信に NNMi が使用する通信の資格情報と SNMP バージョンが決まります。

ノードの SNMP バージョンによって，NNMi でのノードからのトラップの受け入れが，次のように異なります。

- NNMi が SNMPv3 を使用して受信トラップのソースノードやソースオブジェクトを検出すると，NNMi は受信する SNMPv1，SNMPv2c，および SNMPv3 のトラップを受け入れます。
- NNMi が SNMPv1 または複数の SNMPv2c を使用して受信トラップのソースノードやソースオブジェクトを検出すると，NNMi は受信する SNMPv3 トラップを廃棄します。

  このトラップを受信する必要がある場合は，「21.8.1　SNMPv3 以外で管理されているノードまたは監視対象外のノードの SNMPv3 トラップを認証するための NNMi の設定」の手順に従います。

SNMP バージョンと，ネットワークの各領域で受け入れられる最小レベルのセキュリティ設定を指定します。**[SNMP 最小セキュリティレベル]** フィールドのオプションは，次のとおりです。

- **コミュニティのみ（SNMPv1）**

  NNMi は，コミュニティ文字列，タイムアウト，および再試行用に設定した値で SNMPv1 を使って更新を試みます。NNMi は，SNMPv2c や SNMPv3 の設定は試みません。
- **コミュニティのみ（SNMPv1 または v2c）**

  NNMi は，コミュニティ文字列，タイムアウト，および再試行用に設定した値で SNMPv2c を使って更新を試みます。SNMPv2 を使ったコミュニティ文字列への応答がない場合は，NNMi はコミュニティ文字列，タイムアウト，および再試行用に設定した値で SNMPv1 を使って通信を試みます。NNMi は，SNMPv3 の設定は試みません。
- **コミュニティ**

  NNMi は，コミュニティ文字列，タイムアウト，および再試行用に設定した値で SNMPv2c を使って更新を試みます。SNMPv2 を使ったコミュニティ文字列への応答がない場合は，NNMi はコミュニティ文字列，タイムアウト，および再試行用に設定した値で SNMPv1 を使って通信を試みます。機能するものがない場合，NNMi は SNMPv3 を試みます。
- **認証なし，プライバシなし**

  認証もプライバシもないユーザーについて，NNMi はタイムアウトと再試行用に設定した値で SNMPv3 を使って通信を試みます。機能するものがない場合，必要に応じて，NNMi は認証はあるがプライバシがないユーザー，次に認証とプライバシがあるユーザーを試みます。
- **認証，プライバシなし**

  認証はあるがプライバシはないユーザーについて，NNMi はタイムアウトと再試行用に設定した値で SNMPv3 を使って通信を試みます。機能するものがない場合，NNMi は認証とプライバシのあるユーザーを試みます。
- **認証，プライバシ**

  認証もプライバシもあるユーザーについて，NNMi はタイムアウトと再試行用に設定した値で SNMPv3 を使って通信を試みます。

5.1.5　管理アドレスの優先

ノードの**管理アドレス**とは，NNMi がノードの SNMP エージェントと通信する場合に使用するアドレスです。ノードの管理アドレスを指定するか（特定ノードの設定で），またはノードに関連する IP アドレスの中から NNMi がアドレスを選択するようにできます。検出設定で検出から特定のアドレスを除外することによって，この動作を微調整できます。NNMi が管理アドレスを決定する方法については，NNMi ヘルプの「[ノード] フォーム」を参照してください。

> **メモ**
>
> ハイパーバイザー NNMi を検出するには，管理アドレスではなくノード名が必要です。

NNMi は，デバイスの検出と監視を継続的に行います。最初の NNMi 検出サイクルのあと，以前検出した SNMP エージェントが応答しない場合（例えば，デバイスの SNMP エージェントを再設定した場合など）は，**[SNMP アドレス再検出を有効にする]** フィールドの設定によって NNMi の動作が制御されます。

- **[SNMP アドレス再検出を有効にする]** チェックボックスがオンになっている場合，NNMi は機能するアドレスの検索で設定した値を再試行します。
- **[SNMP アドレス再検出を有効にする]** チェックボックスがオフになっている場合，NNMi はデバイスが「停止中」であると報告し，そのデバイスについて別の通信設定を試みません。

[SNMP アドレス再検出を有効にする] チェックボックスは，通信設定のすべてのレベルで使用できます。

自動検出ルール設定フィールドの **[SNMP デバイスの検出]** と **[非 SNMP デバイスの検出]** は，NNMi の SNMP 使用方法に影響します。詳細については，NNMi ヘルプの「自動検出ルールの基本設定を設定する」を参照してください。

5.1.6　SNMPv3 トラップと通知

デバイスと通信するために NNMi で SNMPv3 を使用する場合，検出プロセスを使用して，デバイスのエンジン ID，ブートカウント，エンジン時間が識別されます。NNMi は，ユーザーおよびプロトコルに関する設定済みの詳細とこの情報を併用して，デバイスへのメッセージ送信を開始します。

デバイスから NNMi にトラップを送信する場合，トラップは単一パケットのトランザクションであり必要な情報を取得する手段がないため，デバイスに NNMi 情報が存在しないことがあります。したがって，デバイス自体のエンジン ID，ブートカウント，エンジン時間が，ユーザー名およびプロトコルの詳細とともにトラップで使用されます。デバイスの詳細については，NNMi でデバイス用に設定された内容と同じである必要があります。NNMi では，デバイスごとに複数の SNMPv3 ユーザーを設定できません。

通知は確認済みのパケットであるため，最初のパケットを開始するデバイス，および確認に応答する NNMi は対象外となります。このため，NNMi のエンジン ID，ブートカウント，エンジン時間を取得するために，デバイスから NNMi に対して検出が実行されます。デバイスで使用されるユーザー名およびプロトコ

ルの設定は，NNMi トラップ転送の設定（つまり，NNMi の SNMPv3 エージェント設定）の内容と一致する必要があります。

5.1.7　ポーリングプロトコル

ネットワークの一部で NNMi が SNMP または ICMP 用を使用しないようにできます。例えば，インフラストラクチャ内のファイアウォールが ICMP または SNMP トラフィックを制限する場合などです。

ネットワークのある領域にあるデバイスへの ICMP トラフィックを無効にすると，NNMi では次のような結果になります。

- オプションの自動検出ルール Ping スイープ機能は，ネットワーク領域内で追加ノードを見つけられません。すべてのノードが，シードからとして追加されるか，または近隣 ARP キャッシュ，Cisco Discovery Protocol（CDP），または Extreme Discovery Protocol（EDP）など，MIB オブジェクト要求への応答を通して使用できる必要があります。広域ネットワークデバイスは，すべてシードから追加されるようにしておかないと，監視できない場合があります。
- State Poller は，SNMP 要求に応答するように設定されていないデバイスは監視できません。
- オペレータはトラブルシューティングの間は，［アクション］＞［ノードアクセス］＞［Ping］を使ってデバイス到達可能性をチェックできません。

ネットワークのある領域にあるデバイスへの SNMP トラフィックを無効にすると，NNMi では次のような結果になります。

- 検出では，デバイスが存在すること以外の情報は収集できません。すべてのデバイスで「No SNMP」デバイスプロファイルを適用します。
- 検出では，クエリーによって追加の近隣デバイスを見つけることができません。すべてのデバイスをシードに直接追加する必要があります。
- 検出では，デバイスから接続情報を収集できないため，デバイスは NNMi マップには未接続として示されます。
- 「No SNMP」デバイスファイルを持つデバイスについては，State Poller は ICMP（ping）だけを使用するデバイスの監視のデフォルトが優先されます。
- State Poller は，コンポーネントの稼働状態やパフォーマンスデータをデバイスから収集できません。
- Causal Engine は，近隣接続分析や，インシデントの根本原因を特定するために，デバイスと通信することができません。

5.1.8 nnmsnmp*.ovpl コマンドの動作

nnmsnmp*.ovpl コマンドは，NNMi データベースで指定されていないデバイス通信設定の値を検索します。この方法ではovjboss プロセスが動作している必要があります。ovjboss プロセスが動作していない場合，nnmsnmp*.ovpl コマンドは次のように動作します。

- SNMPv1 エージェントと SNMPv2c エージェントの場合，コマンドは未指定通信設定にデフォルト値を使用します。
- SNMPv3 エージェントの場合，ユーザー ID とパスワードを指定すると，コマンドは未指定通信設定にデフォルト値を使用します。ユーザー ID とパスワードを指定しないとき，コマンドはエラーになります。

5.2　通信の計画作成

次の項目を検討し，通信の計画を作成します。

- デフォルトの通信設定
- 通信設定領域
- 特定のノードの設定
- 再試行とタイムアウトの値
- アクティブなプロトコル
- 複数のコミュニティ文字列または認証プロファイル

5.2.1　デフォルトの通信設定を計画する

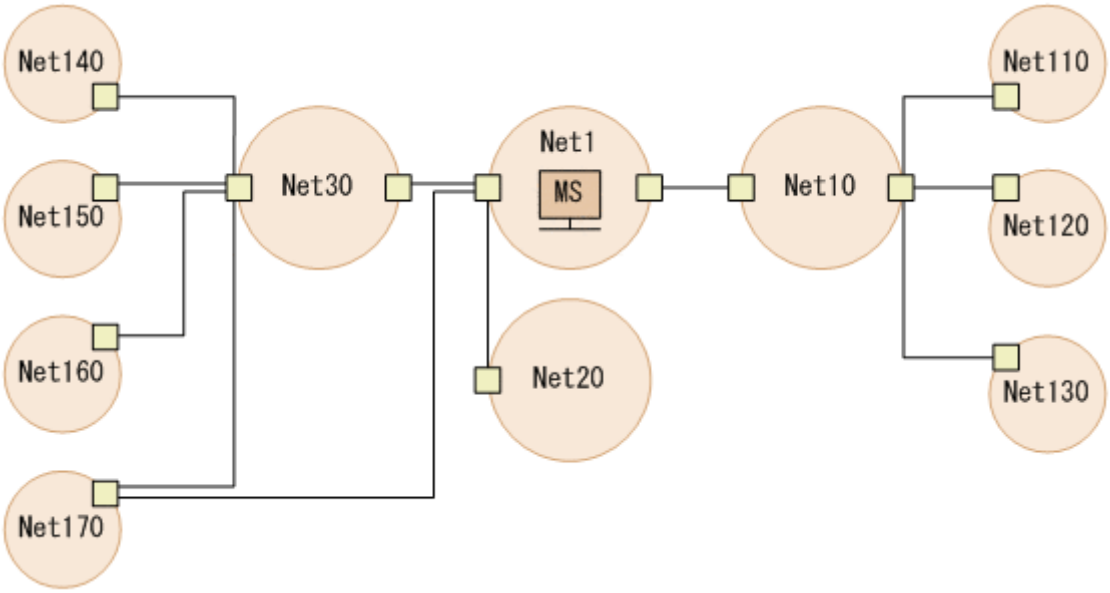
NNMi は，該当する領域や特定のノードで指定しなかった設定を，デフォルト値を使用して完成させるため，大半のネットワークで妥当なものになるようデフォルトを設定します。

- NNMi が試す必要のある一般に使われるコミュニティ文字列がありますか？
- ネットワークではどのようなタイムアウトと再試行のデフォルト値が合理的でしょうか？
- ネットワークデバイスから NNMi にどの SNMP トラップを送信するのですか？

5.2.2　通信設定領域を計画する

領域とは，ネットワーク内で同じ通信設定を適用するのが妥当なエリアのことです。例えば，NNMi 管理サーバーの近くにあるローカルネットワークからは，通常はすぐに応答が戻ってきます。複数ホップ離れたネットワークエリアなら応答にもっと時間がかかるのが普通です。

ネットワークのサブネットやエリアを個別に設定する必要はありません。ラグタイムが近い複数のエリアを 1 つの領域にまとめることができます。次のネットワークマップについて考えてみてください。



タイムアウトと再試行を考慮した場合，次のように領域を設定できます。

- 領域 A　Net 1
- 領域 B　Net 10，Net 20，および Net 30 を含める
- 領域 C　さらに遠くにある外部のネットワーク

NNMi 管理サーバーから 1 ホップまたは 2 ホップのパスのどちらを優先するようトラフィック管理構成が設定されているかに従って，Net 170 をグループにまとめる最良の方法を決定します。

また，類似したアクセス資格認定を使用するデバイスをグループにまとめる場合にも領域を使用します。ネットワークのすべてのルーターで同じコミュニティ文字列（または数種類のコミュニティ文字列の一部）が使用されていて，命名規約（rtrnnn.yourdomain.com など）でルーターを識別できる場合は，すべてのルーターを 1 つの領域に設定すれば，すべてのルーターが同じように処理されます。ワイルドカードを使ってデバイスをグループにまとめられない場合は，各デバイスを特定のノードとして設定できます。

同じタイムアウト／再試行の値とアクセス資格証明設定を 1 つの領域のすべてのノードに適用できるように，領域設定を計画してください。

領域定義は重複することがあり，1 つのデバイスが複数の領域の定義にあてはまることもあります。NNMi は，順序番号が最も小さくて，ほかに一致する領域がない領域から設定を適用します。

5.2.3　特定のノードの設定を計画する

固有の通信設定要件を持つデバイスの場合，特定ノードの設定を使用して，そのノードの通信設定を指定します。特定ノードの設定の使用例として，次の例があります。

- SNMPv2c/SNMPv3 GetBulk 要求に適切に応答しないノード
- ほかの類似ノードと名前のパターンが一致しないノード

特定のデバイスの SNMP 通信を有効または無効にできます。NNMi ヘルプの「[特定ノードの設定] フォーム (通信設定)」を参照してください。

5.2.4　再試行とタイムアウトの値を計画する

タイムアウトの時間を長く，再試行の回数を多く設定すると，ビジー状態であるか，離れたところにあるデバイスからより多くの応答が集められます。このように応答率が高まると，誤ったダウンメッセージを除外できます。しかし，実際にダウンしているデバイスに気づくのに時間がかかるようにもなります。ネットワークの各領域のバランスを見出すことは重要であり，このためにお使いの環境で値のテストと調整の期間が必要になるかもしれません。

各ホップの現在のタイムラグに関するヒントを得るには，次のコマンドを実行します。

- Windows：それぞれのネットワークエリア内のデバイスに対して`tracert` を実行する。
- Linux：それぞれのネットワークエリア内のデバイスに対して`traceroute` を実行する。

5.2.5　アクティブなプロトコルを計画する

通信の設定と監視の設定を使用して，ネットワーク内でデバイスと通信を行うときに NNMi が生成するトラフィックの種類を制御できます。インフラストラクチャのファイアウォールで，ICMP または SNMP のトラフィックが許可されていない場合は，通信の設定を使用します。デバイスに関するデータの特定のサブセットが必要ない場合は，監視の設定を使用してプロトコルの使用を微調整します。通信または監視の設定のどちらかによってデバイスのプロトコルが無効にされると，NNMi はその種類のトラフィックをデバイスに送信しません。

メモ

SNMP 通信を無効にするとデバイスの詳細な情報が得られないため，障害対処など機器の管理が困難になります。

各領域または特定のデバイスは ICMP トラフィックを受信する必要があるか注意してください。

アクセスクレデンシャルを与えないデバイスとの SNMP 通信を明示的に無効にする必要はありません。デフォルトで，NNMi はこれらのデバイスを「`No SNMP`」デバイスプロファイルに割り当て，ICMP だけを使ってデバイスを監視します。

（SNMP エージェントに加えて）Web エージェントが設定されている場合，NNMi は追加のプロトコル（例えば，VMware 環境用の SOAP プロトコル）を使用できます。

5.2.6 コミュニティ文字列と認証プロファイルを計画する

ネットワークの各エリアで試みるコミュニティ文字列と認証プロファイルの計画を作成します。デフォルト設定と領域設定については，並行して試みる複数のコミュニティ文字列と認証プロファイルを設定できます。

> **メモ**
>
> 可能性のあるコミュニティ文字列を試す間に，NNMi クエリーによってデバイスで認証失敗（authentication failure）が生成されることがあります。NNMi が初期検出を完了する間に出された認証失敗は，無視しても問題ないことをオペレータなどに知らせてください。または，領域と試行する関連コミュニティ文字列と認証プロトコルをできる限り厳しく設定して，認証失敗の数を削減することもできます。

環境で SNMPv1 または v2 と SNMPv3 が使用されている場合は，各領域で受け入れられる最低のセキュリティレベルを決定してください。

(1) SNMPv1 と SNMPv2 のコミュニティ文字列

SNMPv1 または SNMPv2c アクセスが可能な領域では，領域内で使用されるコミュニティ文字列と特定のデバイスで必要とされるコミュニティ文字列を集めます。

(2) SNMPv3 の認証プロファイル

SNMPv3 アクセスが可能なデバイスを含む領域では，受け入れられる最小限のデフォルト認証プロファイル，各領域に適した認証プロファイル，および特定のデバイスで使用される固有の認証資格証明を決定します。また，ネットワーク内で使用中の認証プロトコルとプライバシプロトコルも判断します。1 つの特定ノードの設定，または領域の設定に対して，認証プロトコルとプライバシプロトコルを 1 つずつ設定することもできます。

NNMi がサポートする SNMPv3 通信の認証プロトコル

- HMAC-MD5-96
- HMAC-SHA-1
- HMAC128-SHA-224
- HMAC192-SHA-256
- HMAC256-SHA-384
- HMAC384-SHA-512

NNMi がサポートする SNMPv3 通信のプライバシプロトコル

- DES-CBC
- TripleDES

- AES-128
- AES-192
- AES-256

5.3　通信の設定

この節では，次の項目について説明しています。

- SNMP プロキシを設定する
- NETCONF を使用するデバイスのサポート
- VMware ハイパーバイザーベースの仮想ネットワークの検出と監視
- Cisco ACI ネットワークの検出と監視
- マルチホーム NNMi 管理サーバー

この節を読んだあと，詳細な手順については，NNMi ヘルプの「通信プロトコルの設定」を参照してください。

メモ

大幅な設定変更を行う前には，既存の設定のコピーを保存しておくことをお勧めします。詳細については，「4.2　ベストプラクティス：既存の設定を保存する」を参照してください。

通信の次の領域を設定してください。

- デフォルト設定
- 領域定義とその設定
- 特定のノードの設定
  特定のノードについて，NNMi コンソールまたは構成ファイルで，ノードの設定ができます。

ヒント

定義した領域の順序番号をダブルチェックします。ノードが複数の領域を認証する場合，NNMi はそのノードの順序番号の最も小さい領域の設定を適用します。

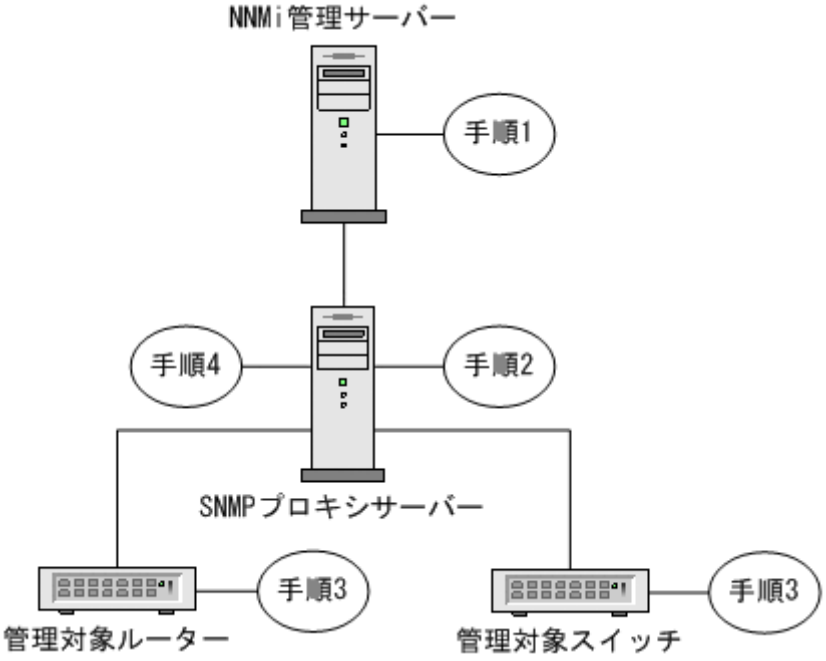
5.3.1　SNMP プロキシを設定する

一部のネットワークでは，ネットワークデバイスとの通信に SNMP プロキシエージェントを使用します。次の図に，NNMi コンソールから［設定］＞［通信の設定］を使用して［SNMP プロキシアドレス］と［SNMP プロキシポート］を設定した場合に，NNMi が使用する SNMP 通信手順を示します。

ヒント

コマンドラインから SNMP プロキシ設定を行う代わりの方法については，`nnmcommunication.ovpl` リファレンスページを参照してください。

図 5-1　プロキシサーバーの使用



1. NNMi 管理サーバーが SNMP プロキシアドレスと SNMP プロキシポートに SNMP 要求を送信し，管理対象ルーターと管理対象スイッチから情報を取得する。

   NNMi 管理サーバーが特殊なプロキシ varbind である SecurityPackAgentAddressOid OID（.1.3.6.1.4.1.99.12.45.1.1）で管理対象ルーターとスイッチのリモートアドレスおよびポートをエンコードし，この varbind を SNMP 要求に追加します。

2. SNMP プロキシサーバーがこの特殊なプロキシ varbind を読み取り，SNMP 要求の送信先を判別して，NNMi 管理サーバーによって要求された情報を取得するために管理対象ルーターとスイッチに SNMP 要求を送信する。

3. 管理対象スイッチとルーターが SNMP プロキシサーバーに応答し（SNMP プロキシアドレスと SNMP プロキシポートを使用），要求された情報を返す。

4. SNMP プロキシサーバーが NNMi 管理サーバーに応答する（設定された SNMP ポートを使用）。

   注：NNMi は，SecurityPackAgentAddressOid OID（.1.3.6.1.4.1.99.12.45.1.1）の使用をサポートする SNMP プロキシサーバーに対応しています。SNMP プロキシ設定を使用しているデバイスに対しては，次のプロパティに基づいてこの OID を SNMP 要求に含めます。

   ```
   com.hp.nnm.snmp.USE_PROXY_VARBIND=true
   ```

   このプロパティのデフォルトの設定は`false` です。

5. SNMP プロキシサーバーは，SNMP による管理対象デバイスからの通知やトラップを NNMi に転送する。

   NNMi は，SNMP プロキシから転送されてきたトラップのソースを判定するために，次の OID の使用をサポートしています。

   - TrapForwardingAddressTypeOid .1.3.6.1.4.1.11.2.17.2.19.1.1.2.0（HP）

- TrapForwardingAddressOid .1.3.6.1.4.1.11.2.17.2.19.1.1.3.0（HP）
- Rfc3584TrapAddressOid .1.3.6.1.6.3.18.1.3.0（RFC 3584）
- Rfc3584TrapCommunityOid .1.3.6.1.6.3.18.1.4.0（RFC 3584）

SNMP プロキシサーバーで NNMi を使用する場合，プロキシベンダーに連絡してこのリスト内の OID をサポートしているかどうかを確認してください。

5.3.2 NETCONF を使用するデバイスのサポート

NNMi は，主として SNMP を使用してサポート対象デバイスの管理情報を収集します。しかし，必要な管理情報が SNMP では報告されない一部のベンダーのデバイスについては，NETCONF を使用する場合もあります。

現在，NNMi で NETCONF を使用する場合にサポートされているデバイスは，Juniper Networks QFabric システムだけです。

ここでは，NETCONF について簡単に紹介し，NNMi で管理対象デバイスをサポートするために，デバイスおよび NNMi の両方で必要な設定について説明します。

(1) NETCONF とは何か

NETCONF は，SNMP と同様に，ネットワーク管理のための IETF（Internet Engineering Task Force）規格です。IETF RFC（Request for Comments）4741 および 4742（Version 1）で定義されており，のちに RFC6241 および 6242（Version 1.1）によって更新されました。

NETCONF は主としてデバイスの設定手段として使用されますが，監視，ポーリング，障害通知の目的では，SNMP が最も広く使用されています。どのプロトコルを使用しても，NNMi にとって有用な管理情報が収集できます。

NNMi は，検出または再検出の場合に NETCONF を使用してデバイス情報（つまり，読み出し専用の情報）を収集します。デバイスの設定を変更する，または状態やパフォーマンス測定指標を監視する目的では，NETCONF を使用しません。

NETCONF は，XML 形式のコマンド応答プロトコルであり，主として SSH（Secure Shell）トランスポート層で動作します。NETCONF プロトコルは，幾つかの点で従来のデバイスコンソールで使用されるコマンドラインインタフェース（CLI）に似ています。しかし，XML 形式のコマンドと結果は，人間とデバイスとの間のインタラクションよりも管理アプリケーションでの使用を念頭に設計されています。

NETCONF は比較的新しい管理プロトコルです。そのため，使用できるデバイスベンダーは，SNMP と比較すると限定的です。

ベンダーが NNMi で管理されているデバイスに NETCONF を実装する場合，次の点に注意してください。

- NETCONF のコマンドは概してベンダー固有であることが多く，SNMP の多くの標準のベンダー固有の MIB ほどは知られていません。その結果，NNMi が NETCONF を活用できる範囲はきわめて限定的です。
- 特定のベンダーがデバイスに NETCONF を実装し，NNMi で必要な管理情報をレポートする場合，NNMi でそのデバイス固有の NETCONF に対応する必要があります。

詳細については，「(3)　管理対象デバイスでの NETCONF の有効化と設定」，および「(4)　NNMi に NETCONF デバイスの認証情報を設定する」を参照してください。

(2)　NETCONF プロトコルの運用

NNMi と管理対象デバイスとの間の NETCONF 通信の詳細なやり取りは，NNMi のユーザーに対して透過的です。しかし，トラブルシューティングには，次の手順が有効な場合があります。

1. NETCONF クライアント（NNMi などの管理アプリケーション）は，管理対象デバイス上の NETCONF サーバー（サブシステム）との間で SSH 接続を確立する。

   有効な SSH ユーザー名およびパスワードの認証情報は，クライアントが指定し，デバイスによって認証される必要があります。
2. クライアントアプリケーションとデバイスは，<hello>メッセージの形式で機能を交換する。
3. クライアントは，標準の<get>または<get-config>演算，およびデバイスに定義されたベンダー固有の演算など，RPC（Remote Procedure Call）メッセージ形式によってデバイスに対して要求を開始する。
4. デバイスは，演算の結果を RPC 応答メッセージの形式で返す。
5. クライアントアプリケーションは，要求の送信および応答の処理を終了したときには，デバイスに<close-session>RPC メッセージを送信する。
6. デバイスは，<ok>RPC 応答メッセージによって受信を確認する。
7. 最後に，双方が SSH 接続を終了する。

(3)　管理対象デバイスでの NETCONF の有効化と設定

NNMi が管理対象デバイスと通信できるようにするために，場合によってはデバイスで明示的に NETCONF を有効化し，設定する必要があります。具体的な設定方法については，デバイスのベンダーが提供するマニュアルを参照してください。

一般に，管理対象デバイスは，次の前提要件を満たす必要があります。

- デフォルトの NETCONF TCP ポート 830，または標準的な SSH TCP ポート 22 のどちらかで，NETCONF を有効化する。
- NETCONF 通信でアクセスできるように，SSH のユーザー名とパスワードの認証情報をデバイスに設定する。

NNMi に対しては，読み出し専用のアクセス権だけが必要です。

(4) NNMi に NETCONF デバイスの認証情報を設定する

NNMi が，NETCONF を使用する管理対象デバイスと通信できるようにするには，デバイスで設定されているのと同じ NETCONF SSH の認証情報を NNMi に設定する必要があります。

デバイスに適切な NETCONF 認証情報が設定されていなくても，NNMi の検出（SNMP を使用した検出だけ）は実行されますが，NNMi に報告されたデバイスの管理情報は完全なものではないことがあります。

NNMi コンソールを使用して，［通信の設定］で［特定ノードの設定］，［領域］または［デフォルト設定］のどれかを選択し，［デフォルトデバイスの資格証明］タブに NETCONF デバイスの認証情報を設定してください。

重要

各管理対象デバイスには，SSH ユーザーおよびパスワードを 1 つだけ設定できます。これは，そのデバイスに対する正規の SSH セッション，および NETCONF セッションに対して，同じ資格情報の組み合わせが使用されることを意味します。

いったん設定されると，NNMi は，指定されたデバイス（ノード）に対して次の検出サイクルの間に新しい資格情報を使用します。

NNMi の［通信の設定］フォームの編集方法の詳細な手順については，NNMi ヘルプ「管理」を参照してください。

5.3.3 VMware ハイパーバイザーベースの仮想ネットワークの検出と監視

(1) ハイパーバイザー上にホストされた仮想マシンを監視するための前提条件

NNMi では次の操作がサポートされます。

- サポート対象ハイパーバイザーの検出と監視。
  ハイパーバイザーのノードフォームでは，各仮想マシンは［ホスト対象ノード］タブに表示されます。
- 各仮想マシン（ルーター，スイッチ，ノードなど）の検出と監視。
  仮想マシンのノードフォームでは，［ホスト元ノード］属性にハイパーバイザーの名前が表示されます。

次の表に，ハイパーバイザーでホストされているハイパーバイザーと仮想マシンを検出するための前提条件を示します。

表 5-1　ハイパーバイザーとその VM を監視するための前提条件

検出対象	前提条件	詳細情報
ハイパーバイザー	ハイパーバイザーは SNMP 通信をサポートする必要があり，SNMP を使用して NNMi からアクセスできる必要があります。	該当しない
	NNMi は関連する SNMP エージェントと通信するように設定する必要があります（IP アドレスとコミュニティ文字列または SNMPv3 認証）。	NNMi ユーザーインタフェースを使用して設定するには，NNMi ヘルプ「管理」の「通信プロトコルの設定」に記載されているデフォルト，領域，または特定ノードについての SNMP の設定方法を参照してください。 コマンドラインインタフェース（CLI）を使用して設定するには，nnmcommunication.ovpl のリファレンスページを参照してください。
	NNMi は，HTTPS を使用してハイパーバイザーと通信するように設定する必要があります。 **注：**（VMware だけ）VMware のデフォルト証明書（localhost.localdomain）を，ESXi サーバーのホスト名を使用して生成された証明書と置き換える必要があります。詳細については，VMware のドキュメントを参照してください。	CLI を使用して設定するには，「(3)　ハイパーバイザーとの通信に HTTPS を使用するように NNMi を設定する」を参照してください。 NNMi ユーザーインタフェースを使用して設定するには，NNMi ヘルプ「管理」の「通信プロトコルの設定」に記載されているデフォルト，領域，または特定ノードについての信頼された証明書の設定方法を参照してください。
ハイパーバイザー上の仮想マシン	ハイパーバイザーの Web サービスで認証を行うには，ハイパーバイザーについて記載された SNMP 要件の他にハイパーバイザーデバイスの資格証明も NNMi に設定する必要があります。	NNMi ユーザーインタフェースを使用して設定するには，NNMi ヘルプ「管理」の「通信プロトコルの設定」に記載されているデフォルト，領域，または特定ノードについての資格証明の設定方法を参照してください。 CLI を使用して設定するには，nnmcommunication.ovpl のリファレンスページを参照してください。

(2)　VMware デフォルト証明書の置換

重要

自己署名または CA 署名証明書は，完全修飾ドメイン名を ESXi サーバーのホスト名として使用して生成する必要があります。

デフォルトでは，VMware 証明書は localhost.localdomain を ESXi サーバーのホスト名として使用します。

VMware のデフォルト証明書を，ESXi サーバーのホスト名を使用して生成された証明書と置き換える必要があります。手順の詳細については，VMware のドキュメントを参照してください。

(3) ハイパーバイザーとの通信に HTTPS を使用するように NNMi を設定する

このセクションでは，CLI を使用して証明書をアップロードする方法を説明します。NNMi ユーザーインタフェースを使用してアップロードする方法については，NNMi ヘルプ「管理」の「通信プロトコルの設定」を参照してください。

メモ

- ハイパーバイザーとの通信に HTTP を使用する必要がある場合は，「(4) ハイパーバイザーとの通信で HTTP を有効にする」も参照してください。
- ハイパーバイザー上でホストされている仮想マシン（VMware ESXi など）を，HTTPS プロトコルを使用して NNMi が監視できるようにするには，次のどちらかの方法でハイパーバイザーの信頼された証明書を NNMi にアップロードする必要があります。
  - NNMi ユーザーインタフェースを使用して信頼された証明書をアップロードする。
  - コマンドラインインタフェース（CLI）を使用して信頼された証明書をアップロードする。
- 信頼された証明書は，HTTPS プロトコルを使用してハイパーバイザーとの信頼性のある接続を確立するために NNMi が使用する SSL 証明書の 1 つです。デフォルトレベルと領域レベルでは，これは同じ CA によって発行された証明書を使用するハイパーバイザーを信頼するために NNMi が使用する CA 証明書を指します。ノードレベルでは，これは FQDN をサブジェクト名として使用して生成された，ハイパーバイザーの SSL 証明書（自己署名または CA 署名）のことです。

信頼された証明書を NNMi にアップロードするには，次の手順を実行します。

1. ハイパーバイザーの信頼された証明書を取得し，NNMi 管理サーバー上の一時的な場所にこれをコピーします。

   メモ

   （VMware だけ）VMware のデフォルト証明書（`localhost.localdomain`）を，ESXi サーバーのホスト名を使用して生成された証明書と置き換える必要があります。詳細については，VMware のドキュメントを参照してください。

2. 証明書がサポートされている形式であることを確認します。サポートされている信頼された証明書ファイルの拡張子は，`.pem`，`.crt`，`.cer`，および`.der` です。
3. 該当するコマンドを実行し，必要なレベルで証明書をアップロードします。次の表から，要件に合うコマンドを選択してください。

レベル	目的	コマンド
デフォルト（グローバル）	同じ CA によって署名された証明書をハイパーバイザー全体で使用する組織が，信頼された証明書をデフォルトレベルでアップロードするために使用します。	nnmcommunication.ovpl addCertificate -default -cert <fully qualified path to the certificate file>

レベル	目的	コマンド
領域	同じ CA によって署名された証明書を特定の領域のハイパーバイザーで使用する組織が，その領域の信頼された証明書をアップロードするために使用します。	nnmcommunication.ovpl addCertificate -region <region name or UUID> -cert <fully qualified path to the certificate file>
ノード	特定のハイパーバイザーで使用する SSL 証明書（CA 署名または自己署名証明書）をアップロードするために使用します。 注：自己署名または CA 署名証明書は，完全修飾ドメイン名（FQDN）をサブジェクト名として使用して生成する必要があります。	nnmcommunication.ovpl addCertificate -nodeSetting <node name or UUID> -cert <fully qualified path to the certificate file>

コマンド例：

- デフォルト：`nnmcommunication.ovpl addCertificate -default -cert /tmp/new.pem`
- 領域：`nnmcommunication.ovpl addCertificate -region region1 -cert /tmp/region1.der`
- ノード：`nnmcommunication.ovpl addCertificate -nodeSetting node1 -cert /tmp/node1.crt`

4. コマンドが正常に実行されると，コマンド出力に，アップロードされた証明書についての情報が表示されます。証明書の情報を確認します。

ヒント

- アップロードした証明書は，`listCertificates` コマンドを使用して表示でき，`removeCertificate` コマンドを使用して削除できます。詳細については，`nnmcommunication.ovpl` のリファレンスページを参照してください。
- ハイパーバイザーが検出された後，Web エージェント上で`updateWebAgentSettings` コマンドを使用して証明書を直接アップロード，置き換え，または削除できます。詳細については，`nnmcommunication.ovpl` のリファレンスページを参照してください。

(4) ハイパーバイザーとの通信で HTTP を有効にする

デフォルトでは，NNMi は HTTPS プロトコルを使用してハイパーバイザーと通信します。

HTTP を使用する必要がある場合は，次の手順で`server.properties` ファイルに必要なプロパティを追加します。

1. `server.properties` ファイルに移動します。

   Windows の場合：`%NnmDataDir%nmsas¥NNM¥server.properties`

   Linux の場合：`$NnmDataDir/nmsas/NNM/server.properties`

2. 次の行を追加します。

```
#VMware vSphere APIなどのSOAPエージェントとの通信でhttpを使用するかどうかを決定します。
#このプロパティはデモ環境またはテスト環境のみで有効にすること、およびHTTPSは本番環境の場合に
```

```
#設定することをお勧めします。
nms.comm.soap.targetconfig.HTTP_ENABLED=true
```

3. NNMi 管理サーバーを再起動します。

   NNMi 管理サーバーでovstop コマンドを実行します。

   NNMi 管理サーバーでovstart コマンドを実行します。

ハイパーバイザーとの通信で HTTP を無効にするには，次の手順を実行します。

1. server.properties ファイルに移動します。

   Windows の場合：%NnmDataDir%nmsas¥NNM¥server.properties

   Linux の場合：$NnmDataDir/nmsas/NNM/server.properties

2. HTTP_ENABLED プロパティ値を false に変更します。

```
nms.comm.soap.targetconfig.HTTP_ENABLED=false
```

3. NNMi 管理サーバーを再起動します。

   NNMi 管理サーバーでovstop コマンドを実行します。

   NNMi 管理サーバーでovstart コマンドを実行します。

4. 「(3)　ハイパーバイザーとの通信に HTTPS を使用するように NNMi を設定する」の手順を実行します。

5.3.4　Cisco ACI ネットワークの検出と監視

Cisco ACI で稼働しているネットワークを検出して監視する場合，次の追加タスクを実行する必要があります。

タスク 1：Cisco APIC コンソールで読み取り専用ユーザーを作成する

読み取り専用権限を持つ Cisco APIC ユーザーを作成します。このユーザーは，Cisco APIC のすべての REST API への読み取りアクセス権を持っている必要があります。このユーザーは，NNMi が Cisco APIC システムを検出するために使用します。ユーザーの作成中，［Create Local User］ページの［Security Domain］セクションで［all］を選択します。

タスク 2：Cisco APIC システムと通信するように NNMi を設定する

検出する Cisco APIC システムごとに，［デバイスの資格証明］フォームを使用してアクセス資格証明を指定します。これらの資格証明により，NNMi は Cisco APIC システムに接続できるようになります。NNMi と Cisco APIC システム間の HTTPS 通信をスムーズに行うには，Cisco ACI または CA によって信頼されている証明書を NNMi 管理サーバーにアップロードする必要があります。

また，Cisco ACI のノードを Cisco ACI のノードとして認識するには，各ノードが SNMP ノードとして検出されている必要があります。

このタスクを行うには，以下の手順を実行します。

1. 検出対象のクラスターごとに 1 つの Cisco APIC システムを特定します。SNMP エージェントが，検出対象のすべてのクラスター内のすべての Cisco APIC システムで有効であることを確認します。

NNMi は，APIC クラスター内の 1 つの Cisco APIC システムのみを検出することで，クラスター内のすべての Cisco APIC システムを最終的に検出できます。

2. Cisco APIC システムに使用するためのすべての信頼済み証明書を取得します。

   各証明書が特定の Cisco APIC システムに固有である証明書一式を使用することも，CA 署名証明書を使用することも，この 2 つの組み合わせを使用することもできます。

3. Cisco ACI API にアクセスして，Cisco APIC システムと通信するように NNMi を設定します。

   a.「タスク 1：Cisco APIC コンソールで読み取り専用ユーザーを作成する」で作成した資格証明を取得します。

   b. **[通信の設定]** フォームで，**[特定ノードの設定]** タブに移動します。

   > **ヒント**
   >
   > 領域の設定やデフォルトの設定でも動作します。

   c. 新しいノードを追加します。

      [特定ノードの設定] タブで，**[新規作成]** をクリックし，**[特定ノードの設定]** フォームで新しいノードを定義します。

      または，既存のノードをダブルクリックします。

   d. **[特定ノードの設定]** フォームで，**[デバイスの資格証明]** タブに移動します。

   e. **[新規作成]** をクリックします。

   f. **[SNMPv1/v2 コミュニティ文字列]** または **[SNMPv3 設定]** タブで，SNMP v1 もしくは v2c 通信文字列，または SNMPv3 資格証明を指定します。

   g. **[特定のノードデバイスの資格証明]** フォームで，**[タイプ]** に **[CiscoACI]** を選択し，「タスク 1：Cisco APIC コンソールで読み取り専用ユーザーを作成する」で作成した Cisco ACI ユーザーの資格証明を指定します。

   h. 信頼済み証明書をアップロードします。HTTP 通信を設定する場合は，この手順をスキップしてください。

      a. **[特定ノードの設定]** フォームで，**[信頼済み証明書]** タブに移動します。

      b. **[証明書のアップロード]** をクリックします。

         [開く] ウィンドウが表示されます。

      c. **[開く]** ウィンドウで，証明書を選択して **[開く]** をクリックします。

         以下のどれか 1 つを使用できます。

         - Cisco APIC システムと通信するための CA 署名証明書

           使用できるのは以下の証明書形式のみです。

           - `.pem`
           - `.crt`
           - `.cer`

- .der

i. [保存して閉じる] をクリックします。

タスク 3：検出を設定して実行する

環境内の Cisco APIC システムをシードするように NNMi を設定します。シードを設定するときには，検出対象の各クラスター内の 1 つの Cisco APIC システムの完全修飾ドメイン名または IP アドレスを指定します。

NNMi 検出によって情報が収集されるのを待機します。NNMi によって，APIC クラスターとすべての Cisco ACI リーフが検出され，検出された Cisco APIC システムが管理するスパインノードが検出されます。

メモ

上記のタスク 1 からタスク 3 までの設定を行わずに Cisco ACI のノードを検出している場合に，この機能を利用して Cisco ACI で稼働しているネットワークを監視するときは，Cisco APIC ノードに対して設定ポーリングを 2 回実行する必要があります。

5.3.5 マルチホーム NNMi 管理サーバー

複数の IP アドレスを持つように NNMi 管理サーバーを設定する場合，管理対象ノードは常に NNMi 管理サーバー上のオペレーティングシステムで設定された IP アドレスを使用します。管理対象ノードと通信およびデータ交換するときに，デフォルト以外の IP アドレスを使用するように NNMi を設定するには，次の手順を実行します。

1. NNMi 管理サーバーにログオンします。
2. テキストエディタで次のファイルを開きます。
   - Windows の場合：%NnmDataDir%shared¥nnm¥conf¥props¥nms-communication.properties
   - Linux の場合：/var/opt/OV/shared/nnm/conf/props/nms-communication.properties
3. デフォルト以外の IPv6 アドレスを設定するには，com.hp.ov.nms.comm.snmp.sourceAddress.IPv6 プロパティのコメントを解除して，このプロパティに選択した IPv6 アドレスを設定します。
4. デフォルト以外の IPv4 アドレスを設定するには，com.hp.ov.nms.comm.snmp.sourceAddress.IPv4 プロパティのコメントを解除して，このプロパティに選択した IPv4 アドレスを設定します。

重要

- 高可用性（HA）で設定を行う場合は，ov.conf ファイルのNNM_INTERFACE プロパティの値は，nms-communication.properties ファイルの com.hp.ov.nms.comm.snmp.sourceAddress.IPv4 プロパティの値と同じである必要があります。ov.conf ファイルの場所は，「19.9.1　NNMi HA 設定ファイル」を参照してください。

- 高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，`ovstop` コマンドおよび`ovstart` コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

5. 変更を保存します。
6. 次のコマンドを実行して，NNMi を再起動します。

```
ovstop -c
ovstart -c
```

5.4　通信の評価

この節では，通信設定の進行と成功を評価する方法を挙げます。多くの作業が完了するのは，検出が完了したあとです。

次について考えます。

- すべてのノードに対して SNMP の設定をしましたか？
  「5.4.1　ノードの SNMP の設定を確認する」を参照してください。
- 現在デバイスに対して SNMP アクセスは可能ですか？
  「5.4.2　SNMP アクセスを確認する」を参照してください。
- 管理 IP アドレスは正しいですか？
  「5.4.3　SNMP デバイスの管理 IP アドレスを確認する」を参照してください。
- NNMi は正しい通信設定を使っていますか？
  「5.4.4　通信設定を確認する」を参照してください。
- State Poller 設定は通信設定と一致していますか？
  「5.4.5　監視設定と通信設定の一致を確認する」を参照してください。

5.4.1　ノードの SNMP の設定を確認する

1. **[ノード]** インベントリビューを開く。
2. **[デバイスのプロファイル]** 列を，文字列「No SNMP」が含まれるようにフィルタリングする。
   - 管理するデバイスごとに，特定ノードの通信設定を行います。その代わりに，領域を拡張して，ノードを組み入れ，アクセスクレデンシャルを更新することもできます。
   - 通信設定が正しい場合は，デバイスの SNMP エージェントが実行中であり，適切に設定されていることを確認します（ACL を含みます）。

5.4.2　SNMP アクセスを確認する

1. インベントリビューでノードを選択する。
2. **[アクション]** ＞ **[ポーリング]** ＞ **[ステータスのポーリング]** または **[アクション]** ＞ **[ポーリング]** ＞ **[設定のポーリング]** を選択する。
   結果に SNMP の値が表示された場合，通信は動作中です。

コマンドラインから`nnmsnmpwalk.ovpl` コマンドで通信をテストすることもできます。詳細については，`nnmsnmpwalk.ovpl` のリファレンスページを参照してください。

5.4.3　SNMP デバイスの管理 IP アドレスを確認する

デバイスに対して NNMi が選択した管理アドレスを判定するには，次の手順を実行します。

1. インベントリビューでノードを選択する。
2. ［アクション］ ＞ ［設定の詳細］ ＞ ［通信設定］ を選択する。
3. ［通信設定］ ウィンドウで，［アクティブな SNMP エージェント設定］ リストにある SNMP エージェントの管理アドレスが正しいことを確認する。

5.4.4　通信設定を確認する

SNMP コミュニティ文字列が欠落しているか，または正しくない場合は，検出が不完全になる可能性があり，検出パフォーマンスに悪影響を及ぼす可能性もあります。

デバイスの通信設定を確認するには，`nnmcommconf.ovpl` コマンドを使用するか，次の手順を実行します。

1. インベントリビューでノードを選択する。
2. ［アクション］ ＞ ［設定の詳細］ ＞ ［通信設定］ を選択する。
   NNMi は，表示された値を求めるために，特定のノード一致，順序番号による領域設定，デフォルト設定をすべて評価します。
3. ［通信設定］ ウィンドウで，SNMP 設定テーブルにリストされた値が，NNMi でこのノードに使用する設定であることを確認する。
   通信設定が正しくない場合，問題解決の手始めとして，SNMP 設定テーブル内のソース情報を使用します。領域や特定ノードの設定や順序番号を変更する必要がでてくる場合もあります。

メモ

VMware 通信の場合，［Web エージェント］ フォームでアクティブ設定を確認するか，または `nnmcommunication.ovpl listWebAgentSettings` コマンドを使用します。

詳細については，NNMi ヘルプ「管理」を参照してください。

5.4.5　監視設定と通信設定の一致を確認する

通信設定によってネットワークの領域へのプロトコルトラフィックが許可される場合でも，その種類のトラフィックは監視設定で無効にされることがあります。設定が上書きされるかどうかを知る手順は次のとおりです。

1. インベントリビューでノードを選択する。
2. ［アクション］ ＞ ［設定の詳細］ ＞ ［モニタリングの設定］ を選択する。

監視設定または通信設定のどちらかによってある種類のデバイスへのトラフィックが無効にされる場合，そのトラフィックは NNMi から送信されません。

5.5 通信の調整

認証失敗の削減

検出の間に NNMi があまりにも多くの認証失敗トラップを生成している場合は，NNMi が試行するアクセスクレデンシャルのグループを小さくし，小さい領域または特定のノードに設定します。

タイムアウトと再試行の調整

NNMi がノード検出中に SNMP を使ってデバイス通信を試みるとき，通信の設定によって NNMi が必要なデバイス情報を収集できるかが決まります。通信の設定に正しい SNMP コミュニティ文字列が含まれていない場合，または NNMi が非 SNMP デバイスの検出をしている場合，NNMi は設定されている SNMP タイムアウトと再試行回数を使用します。この場合，タイムアウトの値が大きいか，または再試行の回数が多いと，検出の全般的パフォーマンスに悪影響が及ぶ可能性があります。SNMP/ICMP 要求に低速で応答することがわかっているデバイスがネットワークにある場合は，［通信の設定］フォームの［領域］タブまたは［特定ノードの設定］タブを使って，これらのデバイスについてだけタイムアウト値と再試行値を微調整することを考えてください。

デフォルトコミュニティ文字列の削減

デフォルトコミュニティ文字列が多数あると，検出パフォーマンスに悪影響が及ぶことがあります。多数のデフォルトコミュニティ文字列を入力する代わりに，［通信の設定］フォームの［領域］タブまたは［特定ノードの設定］タブを使って，ネットワークの特定エリアのコミュニティ文字列設定を微調整します。

6 NNMi 検出

ネットワーク管理で最も重要な作業の 1 つは，常に最新のネットワークトポロジを把握しておくことです。NNMi 検出によって，トポロジインベントリにネットワーク内のノードに関する情報が挿入されます。NNMi では，継続的なスパイラル検出によってこのトポロジ情報が維持されます。これによって，根本原因解析ツールとトラブルシューティングツールで，インシデントに関する正確な情報を把握できるようになります。

この章では，NNMi 検出を設定するために役立つ情報を記載しています。検出がどのようにして行われるのかと検出の設定方法については，NNMi ヘルプの「ネットワークの検出」を参照してください。

6.1　検出の概念

ルーターとスイッチだけを検出するNNMiのデフォルト動作によって，ネットワーク管理を最も重要なデバイスに集中させることができます。つまり，最初にネットワークの基幹をターゲットにします。一般に，末端ノード（例えばパソコンやプリンタ）を管理対象にするのは，それらを重大リソースと見なすのでないかぎり避けるべきでしょう。例えば，データベースやアプリケーションサーバーがクリティカルなリソースとして考えられます。

NNMiで検出するデバイスを管理してNNMiトポロジに加えるには，幾つかの方法があります。ネットワークをどのように構成するかやNNMiで何を管理するかによって，検出構成を単純にしたり，複雑にしたり，その間の適当なレベルに設定したりできます。

重要

NNMiは，デフォルトでの検出を実行しません。各種のデバイスがNNMiトポロジに現れる前に，検出の事前設定をする必要があります。

検出された各ノード（物理または仮想ホスト）は，NNMiがそのノードを能動的に管理しているかどうかに関わらず，ライセンスの限度までカウントします。所有しているNNMiライセンスの数は，検出方法にも影響を及ぼします。

ライセンス情報を追跡する際には，次の点に注意してください。

- **消費量：**NNMiは，NNMiのライセンス容量限界までノードを検出および管理します（切り上げ）。
- **VMware環境：**デバイスプロファイルがvmwareVMの各デバイスは，1/10のノードと同等です。ほかのすべてのデバイスは1つの検出されたノードと同等です。

検出されたノードの数がライセンスされた容量限界に到達したかまたは超えた場合，次のどちらかが行われないかぎり，新しいノードは検出されません。

- ライセンス拡張をインストールする。
- 設定を確認し，NNMi検出をネットワーク環境内の重要なノードだけに限定する。次にノードを削除し，NNMiの再検出でノードの管理対象インベントリをリセットする。

メモ

多数のノードを検出する設定については，NNMiヘルプを参照してください。

多数のノードを検出する設定については，NNMiヘルプを参照してください。

ステータス監視の考慮事項も，選択肢に影響を及ぼします。State Pollerは，デフォルトではNNMiが検出したデバイスに接続したインタフェースしか監視しません。ネットワークの幾つかの領域ではこのデフォルト設定を変更できるため，担当する範囲の先にあるデバイスの検出をすることも可能になります（State Pollerの詳細については，「7.　NNMiステータスポーリング」を参照してください）。

NNMi には，次の 2 つの基本的な検出設定モデルがあります。

- **リストベース検出**－NNMi に，リストのシードによってどのデバイスをデータベースに追加し，監視するかを明示的に指定します。
- **ルールベース検出**－NNMi に，ネットワークのどの領域とデバイスタイプをデータベースに追加するかを指定します。各領域の開始アドレスを指定することで，NNMi に定義済みのデバイスを検出させます。

リストベース検出とルールベース検出を自由に組み合わせて，NNMi の検出対象を設定できます。初回の検出によってこれらのデバイスが NNMi トポロジに追加され，スパイラル検出でネットワークが日常的に再検出されるため，トポロジは常に最新の状態が維持されます。

NNMi では，テナントを使用して重複アドレスドメインを含むネットワークに対応します。重複アドレスドメインは，ネットワーク管理ドメインの静的ネットワークアドレス変換（NAT），動的ネットワークアドレス変換（NAT），またはポートアドレス変換（PAT）領域内に存在することがあります。そのようなネットワークの場合，NNMi はシード検出を使用して重複アドレスドメインを異なるテナントに配置します。詳細については，NNMi ヘルプを参照してください。

重要

- NNMi を使用して VMware ハイパーバイザーベースの仮想ネットワークを管理する場合は，NNMi ヘルプ「管理」の「仮想環境内のテナント」のヘルプトピックを参照してください。
- マルチテナントを設定する場合は，ネットワーク検出を開始する前に，テナントを設定してください。

6.1.1 デバイスプロファイルとデバイスの属性

NNMi はデバイスを検出する際に，SNMP を使用して幾つかの属性を直接収集します。重要な属性の 1 つは MIB II システムオブジェクト ID（sysObjectID）です。システムオブジェクト ID から，NNMi はベンダー，デバイスカテゴリ，デバイスファミリなどの追加属性を導き出します。

検出中，NNMi は MIB II system グループを収集して，データベースのトポロジ部分に格納します。System のケーパビリティは，**［ノード］** フォームに表示されます。ただし，これらのケーパビリティは NNMi の監視設定では使用されません。NNMi では，デバイスカテゴリ（システムオブジェクト ID のデバイスプロファイルによる）を使用して，デバイスをノードグループに分類します。ノードビューのテーブルでは，**［デバイスのカテゴリ］** 列に各ノードのデバイスカテゴリが明示されます。

メモ

（SNMP エージェントに加えて）Web エージェントが設定されている場合，NNMi は追加のプロトコル（例えば，VMware 環境用の SOAP プロトコル）を使用できます。

NNMi には，リリース時に多くのシステムオブジェクト ID のデバイスプロファイルが付属しています。ご使用の環境内のデバイスがデバイスプロファイルにない場合は，デバイスプロファイルをカスタム設定して，これらのデバイスをカテゴリ，ベンダーなどに対応づけることができます。

6.2　検出の計画

次の内容を検討します。

- 基本的な検出方法を選択する
- 自動検出ルール
- ノード名の解決
- サブネット接続ルール
- 検出シード
- 再検出の間隔
- オブジェクトを検出しない
- インタフェースの検出範囲

6.2.1　基本的な検出方法を選択する

リストベース検出だけを行うのか，ルールベース検出だけを行うのか，それともこの 2 つの方法を組み合わせて使用するのかを決定します。

(1)　リストベース検出

リストベース検出では，NNMi で検出する各ノードを検出シードとして明確に指定します。

NNMi では，テナントを使用して重複アドレスドメインを含むネットワークに対応します。重複アドレスドメインは，ネットワーク管理ドメインの静的ネットワークアドレス変換（NAT），動的ネットワークアドレス変換（NAT），またはポートアドレス変換（PAT）領域内に存在することがあります。そのようなネットワークの場合，NNMi はシード検出を使用して重複アドレスドメインを異なるテナントに配置します。詳細については，NNMi ヘルプを参照してください。

重要

- NNMi を使用して VMware ハイパーバイザーベースの仮想ネットワークを管理する場合は，NNMi ヘルプ「管理」の「仮想環境内のテナント」のヘルプトピックを参照してください。
- マルチテナントを設定する場合は，リストベース検出を使用することをお勧めします。

リストベース検出だけを使用することのメリットを次に示します。

- NNMi の管理対象を厳密に管理できます。
- 検出時にデフォルト以外のテナントの機能が利用できます。

- 設定が最も簡単です。
- 固定的なネットワークに適しています。
- NNMi を初めて使用する場合に適した方法です。自動検出ルールを，徐々に追加していくことができます。

リストベース検出だけを使用することのデメリットを次に示します。

- ネットワークに新規ノードが追加されても検出されません。
- 検出対象とするノードのリストを指定しなければなりません。

(2) ルールベースの検出

ルールベース検出では，NNMi が検出して NNMi トポロジに入れるネットワークの領域を定義するために 1 つ以上の自動検出ルールを作成します。それぞれのルールに対して，1 つ以上の検出シードを（シードを明確に指定するか Ping スイープを有効にすることによって）指定する必要があります。それによって NNMi がネットワークを自動的に検出します。

ルールベース検出を使用することのメリットを次に示します。

- 大規模なネットワークに適しています。NNMi は大量のデバイスを，最低限の設定項目に基づいて検出できます。
- 頻繁に変わるネットワークに適しています。ネットワークに追加した新しいデバイスは，管理者が介在しなくても検出されます（各デバイスは自動検出ルールの適用範囲内であることが前提）。

ルールベース検出を使用することのデメリットを次に示します。

- すぐにライセンス限度に達してしまいます。
- ネットワークの構造によっては，自動検出ルールの調整が複雑になることがあります。
- 自動検出ルールが非常に広範囲で，管理しようとしている数以上のデバイスを NNMi が検出する場合，不要なデバイスを NNMi トポロジから削除できますが，ノードの削除には時間が掛かることがあります。
- シードでないノードは，検出時にデフォルトのテナントに割り当てられます。NNMi のマルチテナント機能を使用したい場合は，検出後にテナントの割り当てを変更する必要があります。

6.2.2 自動検出ルールを計画する

(1) 自動検出ルールの設定

自動検出ルールを設定するときは，次の内容を指定します。

- 自動検出ルールの順序
- 検出から除外するデバイス

- Ping スィープを使用するかどうか
- 該当するものがある場合，使用する検出シード

(2) 自動検出ルールの順序

自動検出ルールの**順序**属性の値は，次のように検出範囲に影響します。

- IP アドレス範囲

  デバイスが 2 つの自動検出ルールに該当すると，順序番号が小さい方の自動検出ルールの設定が適用されます。例えばある自動検出ルールによって IP アドレスの一式が除外されると，それより大きな順序番号の自動検出ルールはこれらのノードを処理せず，そのアドレス範囲内のノードは，検出シードとしてリストされないかぎり検出されません。

- システムオブジェクト ID の範囲

  －自動検出ルールに IP アドレス範囲が含まれていない場合は，システムオブジェクト ID の設定が，それより大きな順序番号の自動検出ルールに適用されます。

  －自動検出ルールに IP アドレス範囲が含まれている場合，システムオブジェクト ID 範囲は自動検出ルール内でだけ適用されます。

(3) デバイスを検出から除外

- 特定のオブジェクトタイプが検出されないようにするには，検出したくないシステムオブジェクト ID を無視する自動検出ルールを，小さな順序番号で作成します。このルールに IP アドレス範囲を含めないでください。この自動検出ルールに小さい順序番号を付けることで，検出プロセスはこのルールに一致するオブジェクトを早い段階で読み飛ばします。
- IP アドレス範囲リストまたはシステムオブジェクト ID 範囲リストの中の［**ルールにより無視された**］とマークされたエントリは，その自動検出ルールだけに影響します。無視される範囲内に含まれるデバイスは，別の自動検出ルールに含めることができます。
- ［**検出の設定**］フォームの［**除外対象 IP アドレス**］タブでリストされるアドレスは，すべての自動検出ルールで除外されます。これらのアドレスは検出シードとして設定されないかぎり，NNMi トポロジには追加されません（検出シードは常に検出されます）。

メモ

一部のネットワークでは HSRP や VRRP などのルーティングプロトコルを使用してルーターに冗長性を持たせています。ルーターがルーター冗長グループで設定されている場合，ルーター冗長グループで設定されているルーターは保護された IP アドレス（1 つがアクティブで，1 つがスタンバイ）を共有します。NNMi は，同じ保護された IP アドレスを使用して設定された複数のルーター冗長グループの検出および管理をサポートしません。それぞれのルーター冗長グループには固有の保護された IP アドレスが必要です。

(4) Ping スイープ

NNMi では，Ping スイープを使用して，設定した自動検出ルールの IP アドレス範囲内のデバイスを検索できます。初期検出では，すべてのルールで Ping スイープを有効にするとよいでしょう。これによって十分な情報が NNMi 検出に提供されるので，検出シードを設定する必要がなくなります。

メモ

- Ping スイープは，16 ビットまたはそれより小さいサブネット（例えば 10.10.*.*）で機能します。

  Ping スイープは，特に ISP ネットワークのように制御が不要な WAN 全体でのデバイスの検出に便利です。
- ファイアウォールは Ping スイープをネットワークに対する攻撃として見なすことがよくあります。その場合，ファイアウォールは Ping スイープを発信したデバイスからのすべてのトラフィックをブロックすることがあります。

ヒント

Ping スイープは，小さな検出範囲にだけ有効にしてください。

(5) SNMP トラップからの検出ヒント

NNMi は，受信した SNMP トラップのソース IP アドレスを自動検出ルールに対するヒントとして処理します。SNMP トラップインシデントの詳細については，NNMi ヘルプ「管理」を参照してください。

(6) 自動検出ルールの検出シード

自動検出ルールごとに少なくとも 1 つの検出シードを指定してください。検出シードを指定するには次の方法があります。1 つまたは複数を組み合わせて検出シードを指定してください。

- ［設定］ワークスペース＞［検出］＞［シード］＞［検出シード］フォームでシードを入力します。
- `nnmloadseeds.ovpl` コマンドを使用して，シードファイルから情報をロードします。
- 少なくとも初回の検出で，Ping スイープをルールに対して有効にします。
- SNMP トラップを NNMi 管理サーバーに送信するようにデバイスを設定します。

(7) 自動検出ルールのベストプラクティス

- NNMi はすべての検出対象デバイスを自動的に管理するため，管理したいネットワークの範囲と厳密に一致する IP アドレス範囲を使用してください。

  －複数の IP アドレス範囲を 1 つの自動検出ルール内で使用して，検出を限定できます。

－自動検出ルールに大きな IP アドレス範囲を追加したあとに，そのルール内の検出から幾つかの IP アドレスを除外できます。

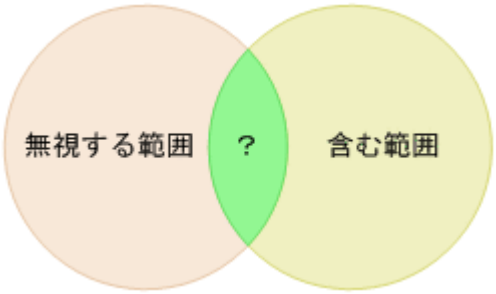
- システムオブジェクト ID 範囲の指定は接頭部分であり，絶対値ではありません。例えば，範囲 1.3.6.1.4.1.11 は 1.3.6.1.4.1.11.*と同じです。

(8) 例

検出ルールの重複

図 6-1 は，重複する 2 つの検出範囲を示しています。左側の円は，NNMi 検出で無視される IP アドレス範囲またはシステムオブジェクト ID 範囲を表しています。右側の円は，NNMi 検出で検出に含まれる IP アドレス範囲またはシステムオブジェクト ID 範囲を表しています。重複している領域は，これらの自動検出ルールの順序に応じて検出に含まれるか無視されます。

図 6-1　重複している検出範囲



デバイスタイプ検出を制限する

ネットワーク内のプリンタ以外のすべての HP デバイスを検出するには，HP エンタープライズシステムオブジェクト ID（1.3.6.1.4.1.11）を含む範囲を持つ 1 つの自動検出ルールを作成します。この自動検出ルールで，HP プリンタ（1.3.6.1.4.1.11.2.3 9）のシステムオブジェクト ID を無視する 2 番目の範囲を作成します。IP アドレス範囲を未設定のままにしてください。

6.2.3 ノード名の解決順序を計画する

デフォルトでは，NNMi はノードを次の順序で識別します。

1. 短い DNS 名
2. 短い sysName
3. IP アドレス

> **重要**
>
> パフォーマンス向上のために，NNMi は名前解決情報をキャッシュします。そのため，ノードのホスト名を変更しても NNMi のデータにはすぐには反映されません。

次の場合は，ノード名解決のデフォルト順序を変更してください。

- 組織が DNS 設定の更新を第三者にまかせている場合，ネットワークに新しいデバイスが追加されるごとにその sysName を定義するポリシーを設定するでしょう。この場合，ノード名解決の最初の選択肢として sysName を設定して，新しいデバイスがネットワークに導入されるとすぐに NNMi が検出できるようにします（sysName を，そのデバイスを使用している間は維持します）。
- 組織が管理対象デバイスの sysName の設定や維持をしない場合，sysName をノード名解決の 3 番目の選択肢として選択します。

ヒント

- DNS 完全名または DNS 短縮名を基本的な命名規則としている場合，NNMi 管理サーバーからすべての管理対象デバイスへの順方向と逆方向の DNS 解決があることを確認してください。

  DNS 完全名を命名規則としている場合，トポロジマップ上のラベルを長くできます。
- NNMi では，最小のループバックアドレスを Cisco デバイスの管理アドレスとして選択されるため，各 Cisco デバイスの最小のループバックアドレス上に DNS 解決を配置してください。

6.2.4 サブネット接続ルールを計画する

リストベース検出だけ

リストベース検出では，サブネット接続ルールを使用して WAN 上の接続を検出します。NNMi は予測される接続の各末端で検出したデバイスのサブネットメンバーシップを評価し（IP アドレスとサブネット接頭部を調べて），サブネット接続ルールで一致があるか調べます。

ルールベース検出だけ

自動検出ルールが有効で NNMi が「/28」と「/31」の間のサブネット接頭部で設定されたデバイスを見つけると，次を実施します。

1. NNMi は適用可能なサブネット接続ルールについて調べます。
2. 一致が見つかると，NNMi はサブネット内の有効な各アドレスをヒントとして使用して，そのアドレスでの検出を試みます。

ヒント

デフォルトの接続ルールを使用してください。問題がある場合だけそれらを変更してください。

6.2.5 検出シードを計画する

検出シードとして使用するデバイスについて説明します。

ヒント

- 優先する管理 IP アドレスを選択するルールの 1 つによって，最初に検出した IP アドレスを管理アドレスとして使用することが指定されます。優先 IP アドレスをシードアドレスとして設定することによって，NNMi に影響を与えることができます。
- Cisco デバイスの場合，ループバックアドレスを検出シードとして使用してください。ループバックアドレスが，デバイス上のほかのアドレスより確実に到達可能であるためです。DNS が，デバイスホスト名からループバックアドレスを解決するように正しく設定されていることを確認してください。

リストベース検出だけ

リストベース検出の場合，NNMi の管理対象にするすべてのデバイスをリスト化します。このリストは，資産管理ソフトウェアまたはほかのツールからエクスポートできるでしょう。

NNMi は，このリストに自動的にデバイスを追加しないので，担当しているすべてのデバイスや，監視やステータス計算に影響を及ぼすすべてのデバイスが，リストに含まれるようにしてください。

ルールベース検出だけ

ルールベース検出の場合，検出シードは任意で指定します。

Ping スイープが自動検出ルールに対して有効な場合，そのルールのシードを指定する必要はありません。

Ping スイープが無効な各自動検出ルールでは，ルールごとに少なくとも 1 つのシードを確認してください。ルールに IP アドレス範囲が複数含まれる場合，ルーターは WAN リンクを横断した ARP エントリを保持しないため，それぞれのルーティング可能範囲でシードが必要になります。

ヒント

ルールベース検出を完全なものにするためには，スイッチではなくルーターを検出シードとして使用してください。一般にルーターはスイッチより大きな ARP キャッシュを持っているためです。検出したいネットワークにコアルーターが接続されていれば，検出シードとしては最適な選択肢になります。

6.2.6 再検出の間隔を計画する

NNMi は，データベース内の各デバイスの設定情報を，設定された再検出間隔に従って再チェックします。さらに，NNMi は自動検出ルールの対象となる各ルーターから ARP キャッシュを収集して，ネットワーク上に新しいノードがあるか調べます。

デバイスの通信関連の設定に，インタフェースの番号変更のような変更があると，NNMi は自動的に，そのデバイスとその隣接デバイスに関するデータを更新します。

次のような変更では自動再検出のきっかけになりません。デバイスは設定された再検出間隔に基づいて更新されます。

- ノード内の変更（例えば，ファームウェアアップグレードまたはシステムの連絡先）。
- ネットワークに新しいノードが追加された。

ネットワーク内の変更のレベルに合った再検出間隔を選択します。構成が頻繁に変化するネットワークでは，最低 24 時間の間隔を使用することをお勧めします。構成が安定したネットワークでは，再検出間隔を広げることができます。

6.2.7　オブジェクトを検出しない方法を計画する

NNMi では，NNMi が特定のオブジェクトを無視するように設定する 5 つの方法があります。

- **［通信の設定］** フォームで，ICMP 通信や SNMP 通信（またはその両方）を，グローバルレベル，通信領域レベル，または特定のホスト名または IP アドレスのレベルの異なるレベルでオフにできます。これらのプロトコルのどちらかまたは両方を無効にした場合の影響の詳細については，「5.1.7　ポーリングプロトコル」を参照してください。
- **［検出の設定］** フォームで，特定の IP アドレスや SNMP システムオブジェクト ID からヒントを収集しない自動検出ルールを設定できます。この基準に一致するノードはマップとデータベース上で存在し続けますが，スパイラル検出ではこれらの IP アドレスまたはオブジェクトタイプを超える隣接デバイスの検出はしません。
- **［検出の設定］** フォームで，特定の IP アドレス範囲や特定の IP アドレス（またはその両方）をデータベースから除外する自動検出ルールを設定できます。スパイラル検出では，あらゆるノードのアドレスリストでこれらのアドレスを表示したり，デバイス間に接続を確立するとき，これらのアドレスを使用したりすることがないので，NNMi がこれらのアドレスの使用状況を監視することはありません。
- **［検出の設定］** フォームの **［除外対象 IP アドレス］** タブで，除外対象 IP アドレスフィルタを設定して，IP アドレス範囲を検出から除外できます。

  あるノードがすでに検出されたあとに，そのノードのすべての IP アドレスを **［除外対象 IP アドレス］** リストに入力しても，NNMi はそのノードを削除しません。さらに，NNMi の管理者が意図的に NNMi データベースからそのノードを削除しないかぎり，NNMi はそのノードの履歴全体を削除しません。

  IP アドレス範囲を除外する場合，ネットワーク管理ドメインの静的ネットワークアドレス変換（NAT），動的ネットワークアドレス変換（NAT），またはポートアドレス変換（PAT）領域内の重複アドレスも除外されます。

  NNMi では，テナントを使用して重複アドレスドメインを含むネットワークに対応します。そのようなネットワークの場合，NNMi はシード検出を使用して重複アドレスドメインを異なるテナントに配置します。詳細については，NNMi ヘルプを参照してください。
- **［検出の設定］** フォームの **［除外対象インタフェース］** タブで，インタフェースグループを選択して，特定のタイプのインタフェースを検出プロセスから除外できます。詳細については，NNMi ヘルプを参照してください。

6.2.8　インタフェースの検出範囲

NNMi では，フィルタを定義して検出されるインタフェース範囲を指定できます。これは，ノードが大きく，インタフェースのサブセットだけを検出する場合に特に便利です。**[除外対象インタフェース]** オプションを使用する場合は，デバイスから情報を取得したあとでインタフェースがフィルタリングされますが，検出するインタフェース範囲を指定する場合は，NNMi から範囲外のインタフェースに関する情報は要求されません。そのため，範囲ベースの検出では，一部のインタフェースを管理する場合，大きいデバイスの検出パフォーマンスを向上できます。

[検出の設定] フォームの **[含まれるインタフェース範囲]** タブで，システムオブジェクト ID プレフィック値および ifIndex 値を使用してインタフェース範囲を定義します。詳細については，NNMi ヘルプを参照してください。

6.3　検出の設定

ここでは，設定のヒントを一覧にし，幾つかの設定例について説明します。この項を読んだあとで，特定の手順の NNMi ヘルプの「検出を設定する」を参照してください。

重要

NNMi は，［検出シード］フォームを［保存して閉じる］とすぐにシードから検出を開始するので，シードを設定する前に次のことを必ず行ってください。

- すべての通信設定を完了する。
- すべての自動検出ルールを完了する（設定が必要な場合）。
- サブネット接続ルールを設定する。
- 名前解決を設定する。
- コンソールまでさかのぼって，すべての設定フォームの［保存して閉じる］を実行する。

メモ

ルールベース検出の場合，大幅な設定変更を行う前には，既存の設定のコピーを保存しておくことをお勧めします。詳細については，「4.2　ベストプラクティス：既存の設定を保存する」を参照してください。

6.3.1　自動検出ルールを設定する場合のヒント

- 新しい自動検出ルールを定義するときは，それぞれの設定を慎重に確認してください。新しいルールの定義では，自動検出はデフォルトで有効になっており，IP アドレス範囲はデフォルトで含まれており，システムオブジェクト ID 範囲はデフォルトで無視されます。

6.3.2　シードを設定する場合のヒント

シードを設定するときは，次のベストプラクティスを検討してください。

- 検出対象ノードがリスト化されたファイルがすでにある場合は，この情報をシードファイルとして書式設定して，`nnmloadseeds.ovpl` コマンドで NNMi にインポートします。
- NNMi が選択する IP アドレスに影響を与えるために，シードファイルに管理アドレスとして IP アドレスを指定します（ホスト名を使用すると，DNS が各ノードの IP アドレスを提供します）。
- シードファイルのエントリの書式を，次に示します。

```
IP_address # node name
IP_address2, ”<テナント名またはテナントのUUID>” # node name
```

- 保守目的のため，使用するシードファイルは 1 つだけにすることをお勧めします。必要に応じてノードを追加して，nnmloadseeds.ovpl コマンドを再度実行します。NNMi は新しいノードを検出しますが，既存のノードは再判定しません。
- ノードをシードファイルから削除しても，NNMi トポロジからは削除されません。NNMi トポロジからのノード削除は直接 NNMi コンソールで実施してください。
- ノードをマップやインベントリビューから削除しても，シードは削除されません。
- NNMi でノードを再検出したい場合は，そのノードをマップまたはインベントリビューと，NNMi コンソールの［設定］＞［検出］＞［シード］ビューから削除してから，そのノードを NNMi コンソールの［検出シード］フォームで再度入力するか，nnmloadseeds.ovpl コマンドを実行します。

ルールベース検出だけ

検出ルールは，そのルールに対するシードを指定する前に設定します。つまり，［検出の設定］フォームで［保存して閉じる］をクリックします（［検出シード］フォームで情報を保存すると，シード設定はすぐに更新されます）。

6.3.3 リンクアグリゲーションの検出

リンクアグリゲーションには，NNMi Advanced ライセンスが必要です。

リンクアグリゲーション（LAG）プロトコルによって，ネットワーク管理者はアグリゲータインタフェースとしてスイッチでインタフェースのセットを設定できます。この設定によって，帯域幅，データ通信速度，冗長性の向上と並行して，複数のインタフェースを使用して別のデバイスにアグリゲータレイヤー 2 接続を作成します。

詳細については，NNMi ヘルプでリンクアグリゲーションを検索してください。

6.3.4 サーバーからスイッチへのリンクアグリゲーション（S2SLA）の検出について

リンクアグリゲーションには，NNMi Advanced ライセンスが必要です。

ネットワーク管理者は，信頼性の向上およびサーバーとスイッチ間のリソースのさらなる活用を頻繁に求められます。多くのネットワーク管理者が，ネットワーク機器プロバイダーでは広範な使用法があるために，Link Aggregation Configuration Protocol（LACP）の使用を選択します。LACP は，IT エンジニアがサーバーからスイッチへの設定の両側でポートを結合したあとに，自動的にネゴシエーションされます。

ネットワーク管理者は多くの場合，信頼性および必要なサーバーとスイッチ間のリソースの使用を実現するために，2 種類のうちどちらかのスイッチからサーバーへの接続を使用することを選択します。

- オプション 1：サーバーの 2 つ以上のポートを結合し，スイッチにある同じ番号のポートに接続します。サーバーまたはスイッチのポートに障害が発生すると，バックアップポートがアクティブ化されます。
- オプション 2：サーバーとスイッチの両方を結合し，集約してすべてのポートの集約合計帯域幅を提供します。

  NNMi は，サーバーからスイッチへのリンクアグリゲーション（S2SLA）の検出機能を提供し，スイッチからサーバーへの接続の管理を容易にします。NNMi がノードの S2SLA 情報を適切に検出できるか確認するには，次のタスクを実行してください。

- デフォルトで，Linux は SNMP エージェントパッケージ，Net-SNMP をインストールしません。Net-SNMP が NNMi 管理サーバーにない場合，インストールする必要があります。
- Linux 上で結合しているインタフェースは，集約されたインタフェースの 1 つの MAC アドレスを前提とすることがありますが，必須ではありません。結合されたインタフェースは，どのサーバーのインタフェースにも属さない MAC アドレスを持つことができます。

メモ

集約でのすべてのインタフェースで同じ MAC アドレスが使用されます。SNMP インタフェーステーブルを確認して，アグリゲータインタフェースおよび集約されたインタフェースに同じ MAC アドレスを返します。共有 MAC アドレスは送信パケットで使用されます。アクセススイッチの FDB テーブルは，スイッチの集約されたインタフェースを介して伝えられると，この MAC アドレスを示します。

元の MAC アドレスを表示するには，次のコマンドを実行します。

```
cat /proc/net/bonding/bond0
```

6.4 検出の評価

ここでは，検出の進行状況と成功したかどうかを判定する方法を説明します。

6.4.1 初期検出の進行状況をたどる

NNMi 検出は，動的かつ継続的です。完了することはないため，「検出完了」のメッセージが表示されることはありません。初回の検出と接続には，多少の時間が掛かります。初期検出の進行状況を測定する方法を次に示します。

- ［システム情報］ウィンドウの［データベース］タブで，ノードカウントが予想レベルに達して一定になるのを監視します。このウィンドウは自動的に更新されません。初期検出時に，［システム情報］ウィンドウを複数回開きます。
- ［設定］＞［検出］＞［シード］ビューを見てください。このビューを，すべてのシードに「ノードが作成されました」の結果が表示されるまで更新してください。「ノードが作成されました」の結果は，デバイスがトポロジデータベースに追加されたことを示します。この結果は，NNMi がデバイスからすべての情報を収集してデバイスの接続を処理したことを示すものではありません。
- 代表ノードの［ノード］フォームを開きます。［検出状態］フィールドが「検出が完了」に移行するときには，NNMi はノードの基本特性，ノードの ARP キャッシュ，隣接検出プロトコル（該当する場合）の収集を済ませています。この状態は，NNMi がデバイスの接続解析を完了したことを示すものではありません。
- ［ノード］インベントリビューで，ネットワークのさまざまな領域のキーデバイスが存在していることを確認します。
- 代表ノードの［レイヤー 2 の近隣接続ビュー］を開き，その領域の接続解析が完了したかどうかを確認します。
- ［レイヤー 2 の接続］および［VLAN］インベントリビューを調べて，レイヤー 2 処理の進行状況を測定します。

6.4.2 シードの検出を確認する

1. ［シード］ビューを開く。
2. ［シード］ビューで，ノードのリストを［検出シードの結果］列でソートする。

   ノードがエラー状態の場合は，次について検討してください。

   - ノードに到達できなかった，または DNS 名が解決されなかったために検出が失敗した－これらのタイプの失敗に対しては，ノードへのネットワーク接続を確認して，DNS 名解決が正しいかどうかを調べてください。DNS 問題に対処するには，IP アドレスを使用してノードをシードするか，ホスト名を`hostnolookup.conf` ファイルに加えます。

IP アドレスが原因で名前解決されない場合には，該当する IP アドレスをipnolookup.conf ファイルに含めます。詳細については，hostnolookup.conf およびipnolookup.conf のリファレンスページを参照してください。

- ライセンスノード数超過－この状況は，検出されたデバイス数がライセンス限度に達したときに発生します。検出したノードを幾つか削除するか，ライセンスの追加を検討してください。

  ライセンス情報を追跡する際には，次の点に注意してください。

  - **消費量**：NNMi は，NNMi のライセンス容量限界までノードを検出および管理します（切り上げ）。
  - **VMware 環境**：デバイスプロファイルが vmwareVM の各デバイスは，1/10 のノードと同等です。

    ほかのすべてのデバイスは 1 つの検出されたノードと同等です。

- ノードが検出されたが SNMP 応答がない－SNMP 通信の問題は，シードされたデバイスだけでなく，自動検出によって検出されたデバイスにも発生します。詳細については，「5.4　通信の評価」を参照してください。

6.4.3　有効なデバイスプロファイルを確認する

1. **[ノード]** インベントリビューを開く。
2. **[デバイスのプロファイル]** 列を，「No Device Profile」文字列が含まれるようにフィルタリングする。
3. ノードが検出されてもデバイスプロファイルがない場合は，**[設定]** ＞ **[デバイスのプロファイル]** で新規デバイスプロファイルを追加してから，ノードに対して設定のポーリングを実行してそのデータを更新する。

6.4.4　ノードの検出を確認する

すべてのノードが正しく検出されるために，管理ドメイン内のほかのドメインには表示されない固有の IP アドレスを使用するノードだけを NNMi で管理するようにします。例えば，ノードが突然消えたり，データベース内の別のノードとマージされたりして，そのノードがルーター冗長グループの一部になっている場合，ルーター冗長グループに参加しているルーターを管理するには，ルーターの管理アドレスとして保護されたアドレス以外の固有の IP アドレスを使用し，そのアドレスで SNMP を有効にする必要があります。保護された IP アドレスを管理アドレスとして使用しようとすると，NNMi はルーターを適切に管理できません。

[ノード] インベントリビューでデータを調べます。管理アドレスがないノードがある場合は，これらのノードの通信設定を「5.4.1　ノードの SNMP の設定を確認する」の説明に従って確認します。

予想したノードが **[ノード]** インベントリビューにない場合は，次について確認します。

- 見つからなかったノードごとに，検出プロトコル（例えば CDP）が正しく設定されていることを確認します。

- 見つからないノードが WAN 上にある場合，そのノードを含む自動検出ルールの Ping スイープを有効にします。

6.4.5　自動検出ルールを評価する（ルールベース検出だけ）

予期しない検出結果に遭遇した場合は，自動検出ルールを再検討します。

NNMi 検出でアドレスヒントが見つかる場合は，最初の一致ルールを使用してノードを作成するかどうかを判定しています。一致するルールがない場合，NNMi 検出はヒントを廃棄します。自動検出ルールの順序番号によって，自動検出ルール設定が適用される順序が決まります。

それぞれの自動検出ルールで，次の設定を確認してください。

- ［マッチングノードの検出］を有効にし，自動検出がルールで実行されるようにする必要があります。
- 次の設定が，検出したいノードのタイプに対して正しいかどうかを確認します。
  －SNMP デバイスの検出
  －非 SNMP デバイスの検出
  デフォルトではルーターとスイッチだけが検出されて，SNMP 以外のノードは検出されません。［SNMP デバイスの検出］を有効にすると，すべての SNMP デバイスを検出します。［非 SNMP デバイスの検出］を有効にすると，非 SNMP デバイスも検出します。ご使用の環境を考慮せずにこれらの設定を有効にすると，予期した以上のノードを検出してしまうおそれがあります。

(1)　IP アドレス範囲

検出ヒントの IP アドレスは，IP アドレス範囲リスト内の［ルールに含める］エントリと一致する必要があります。含まれる IP アドレス範囲が自動検出ルールの中にない場合，すべてのアドレスヒントが一致と見なされます（この場合は，「6.3.1　自動検出ルールを設定する場合のヒント」を参照してください）。さらに，アドレスは「ルールにより無視された」とマークされたエントリと一致してはなりません。すべてのチェックが正常に一致すると，そのルールの設定が検出ヒントの処理に使用されます。

- 予想したデバイスの幾つかが検出されない場合，そのデバイスの IP アドレスが範囲の中に含まれていているか，また小さい順序番号のルールで無視されていないかを確認してください。
- 必要以上のデバイスが検出されている場合は，検出範囲を変更するか，検出したくないデバイスの IP アドレスが無視される範囲を追加してください。また，［SNMP デバイスの検出］も有効かどうかを確認します。

(2)　システムオブジェクト ID の範囲

検出ヒントのシステムオブジェクト ID（OID）は，システムオブジェクト ID 範囲リストの中の［ルールに含める］エントリと一致する必要があります。含まれるシステムオブジェクト ID 範囲が自動検出ルールの中にない場合，すべてのオブジェクト ID が一致と見なされます。さらに，OID は「ルールにより無

視された」とマークされたエントリと一致してはなりません。すべてのチェックが正常に一致すると，そのルールの設定は検出ヒントの処理に使用されます。

- システムオブジェクト ID 範囲を使用して，自動検出を拡大し，デフォルトのルーターおよびスイッチ以外も含めるか，特定のルーターおよびスイッチを除外します。
- 検出された各ノードは，トポロジデータベースに追加される前に指定された IP アドレス範囲とシステムオブジェクト ID 範囲の両方と一致する必要があります。

6.4.6　接続と VLAN を評価する

NNMi はレイヤー 2 接続と VLAN を，デバイスがトポロジに追加されたあとの別のステップとして作成します。接続と VLAN を評価する前の初期検出として十分な時間を考慮してください。

レイヤー 2 の接続を評価するには，対象とする各ネットワーク領域のノードグループを作成し，続いてそのノードグループのトポロジマップを表示します（[ノードグループ］インベントリで，ノードグループを選択して，［アクション］＞［マップ］＞［ノードグループマップ］をクリックします）。このマップでほかのノードに接続していないノードを探します。

VLAN を評価するには，［VLAN］インベントリビューからそれぞれの［VLAN］フォームを開いて，その VLAN のポートのリストを調べます。

6.4.7　デバイスを再検出する

デバイスの削除を確認するには，次の手順を実行します。

1. デバイスの設定ポーリングを実行する。
2. デバイスを削除する。

そのデバイスがシードの場合，シードを削除してから再度シードを追加します。

6.5　検出の調整

標準的な検出が行われるようにするためには，検出設定を調整して重大なデバイスと重要なデバイスだけが検出されるようにしてください。

- IP アドレス範囲やシステムオブジェクト ID（またはその両方）でフィルタリングします。
- 非 SNMP デバイスと SNMP デバイス（スイッチでもルーターでもないデバイス）の検出を制限します。
- コマンドラインで NNMi データベースからノードを削除するには，`nnmnodedelete.ovpl` コマンドを使用します。このコマンドで，NNMi データベースからノードが削除されますが，シード定義は削除されません。コマンドラインで NNMi データベースからシード定義を削除するには，`nnmseeddelete.ovpl` コマンドを使用します。
- 検出プロトコルコレクションを無効にすることで修復できる特別な検出状況もあります。詳細については，「21.24　特定ノードに対して検出プロトコルを使用しないように設定する」を参照してください。

6.5.1　応答のないオブジェクトを削除する

応答がなくなってから削除するまでの日数を指定して，次のオブジェクトを削除できます。

- 応答のないノード
- 停止中の接続

応答のないノードを削除するには，次の手順を実行します。

1. **［設定］** ワークスペースで，**［検出］** ＞ **［検出の設定］** を選択する。
2. **［非応答オブジェクト制御の削除］** 領域で，対象のオブジェクトを削除するまでの日数を入力する。
   オブジェクトを削除しない場合は，「0」を入力します。指定した日数が経過したあとに，応答のないオブジェクトはデータベースから削除されます。

メモ

［非応答ノードの削除］が有効な場合，NNMi は次の状況下にある仮想マシンノードを削除しません。

- VM が SNMP エージェントをサポートしていない
- VMware Tools がインストールされていないために VM に IP アドレスがない
- VM の IP アドレス障害モニタリングが設定されていない

詳細については，NNMi ヘルプ「管理」の「スケジュールを設定する」のヘルプトピック記載されている「応答のないノードを削除するかどうか設定する」を参照してください。

7 NNMi ステータスポーリング

この章では，NNMi State Poller サービスを設定し，ネットワーク監視を拡張および微調整するための情報を示します。

この章は，NNMi ヘルプの情報を補充するものです。監視動作方法の紹介，および監視設定方法の詳細は，NNMi ヘルプの「ネットワークの稼働状態をモニタリングする」を参照してください。

7.1　ステータスポーリングの概念

この節では，State Poller がポーリンググループの評価に使う順序など，ネットワーク監視の簡単な概要を示します。この項を読んだあと，さらに詳細な情報については「7.2　ステータスポーリングの計画」に進んでください。

ネットワーク検出と同じように，ネットワークでクリティカル，または最も重要なデバイスのネットワーク監視に関心を集中する必要があります。NNMi は，トポロジデータベースにあるデバイスにだけポーリングを実施できます。どのネットワークデバイスを監視するか，使用するポーリングの種類，およびポーリングする間隔を制御できます。

[モニタリングの設定] フォームのインタフェースとノードの設定を使って，デバイスのステータスポーリングを高度化し，さまざまなクラス，インタフェースの種類，およびノードの種類についてポーリングの種類と間隔を設定できます。

State Poller のデータ収集が ICMP（ping）応答を基礎にするか，または SNMP データを基礎にするかを設定できます。NNMi は，ユーザーが有効にするデータ収集の種類から，実際の MIB オブジェクトへの内部的なマップを自動処理し，設定を大幅に簡略化します。

メモ

（SNMP エージェントに加えて）Web エージェントが設定されている場合，NNMi は追加のプロトコル（例えば，VMware 環境用の SOAP プロトコル）を使用できます。

ポーリング設定の計画を作成するときは，State Poller サービス用にインタフェースグループとノードグループをセットアップする方法について考える必要があります。グループという概念については「4.6 ノードグループおよびインタフェースグループ」と「4.7　ノード／インタフェース／アドレス階層」を参照してください。

7.1.1　評価の順序

インタフェースまたはノードは複数のグループに属することがあるので，State Poller は，定義された評価順序で，設定されたポーリング間隔およびポーリング種類を適用します。検出されたトポロジ内の各オブジェクトについて次のように評価されます。

1. オブジェクトがインタフェースの場合，State Poller は基準を満たすインタフェースグループを探す。グループは小さい順序番号から大きい順序番号へ順に評価されます。最初に一致するグループが見つかると，その時点で評価は停止します。
2. オブジェクトに一致するインタフェースグループがない場合，ノードグループが小さい順序番号から大きい順序番号へ順に評価される。

最初に一致するグループが見つかると，その時点で評価は停止します。インタフェースのうち，独自の特性に関してインタフェースグループと一致しないものは，所属するノードからポーリング設定を継承します。

3. 検出されたものの，ノードまたはインタフェースの設定定義に含まれないデバイスは，グローバルな監視設定（［モニタリングの設定］フォームの［デフォルト設定］タブ）によって監視動作が確定される。

7.2　ステータスポーリングの計画

この節では，ポーリング設定チェックリストなど，State Poller 設定の計画作成について説明します。監視の計画作成に便利な詳細情報によって，ポーリンググループの作成法が決まり，ポーリングプロセスの間にどの種類のデータを取得する必要があるかが決まります。

7.2.1　ポーリングチェックリスト

次のチェックリストを使って，State Poller 設定の計画を作成できます。

□NNMi で何を監視できますか？

□オブジェクトの種類，場所，相対的重要性，そのほかの基準に基づいて，監視対象は論理的にどのように分類できますか？

□NNMi は，各グループをどのくらいの頻度で監視する必要がありますか？

□監視されるアイテムの情報を取得するために，何のデータを収集する必要がありますか？次のものが含まれることがあります。

－ICMP（ping）応答

－SNMP 障害データ

－追加の SNMP コンポーネント稼働状態データ

メモ

（SNMP エージェントに加えて）Web エージェントが設定されている場合，NNMi は追加のプロトコル（例えば，VMware 環境用の SOAP プロトコル）を使用できます。

(1)　ポーリング設定の例

ポーリング設定プロセスの理解を深めるために，次の例について考えます。ネットワークに ProximiT の最新のプロキシサーバーが含まれていると仮定します。これらのデバイスに到達できることを確認する必要がありますが，プロキシサーバーの SNMP 監視は要求しません。

1. NNMi で何を監視できますか？
   監視できるのは検出されたものだけであるため，自動検出ルールを設定して，NNMi のデータベースに自分のプロキシサーバーがあることを確認します。検出の設定の詳細は，「6.　NNMi 検出」を参照してください。
2. 監視対象は論理的にどのように分類できますか？

複数のプロキシサーバーを 1 つのグループに分類し，同じ監視設定を適用するのが合理的です。デバイスのインタフェース（SNMP）監視を行っていないので，インタフェースグループは必要ありません。

このノードグループを使ってビューをフィルタし，プロキシサーバーのステータスをグループとしてチェックし，グループをサービス停止中にしてファームウェアを更新することもできます。

3. NNMi は，各グループをどのくらいの頻度で監視する必要がありますか？

   サービスレベル契約条項で，プロキシサーバーについて 5 分間のポーリング間隔で十分です。

4. どのデータを収集する必要がありますか？

   監視設定がほかのグループと異なるのは次の点です。プロキシサーバーの例として，ICMP 障害の監視を有効にし，SNMP 障害ポーリングの監視を無効にします。グループについての SNMP 障害監視がない場合，コンポーネント稼働状態監視は適用されません。

5. ネットワークデバイスから NNMi にどの SNMP トラップを送信するのですか？

   次のポーリング間隔を待機しないでトラップが受信される場合，NNMi は一部の SNMP トラップを使用してデバイスをポーリングします。

   これらの設定選択肢に関する計画作成情報の詳細は，次の項を参照してください。

   - 「7.2.2　NNMi で監視できる項目」
   - 「7.2.3　監視の停止」
   - 「7.2.4　監視されないノードへのインタフェース」
   - 「7.2.5　モニタリングの拡張」
   - 「7.2.6　ノードグループとインタフェースグループを作成する」
   - 「7.2.7　ポーリング間隔を計画する」
   - 「7.2.8　収集するデータを計画する」
   - 「7.2.9　SNMP トラップが NNMi に送信する内容を決定する」

7.2.2　NNMi で監視できる項目

State Poller サービスは，検出された各インタフェース，アドレス，および管理ドメインでアクティブに監視されるように指定されている SNMP エージェントを監視します。 State Poller サービスは，カード，シャーシ，ノードセンサー，物理センサー，ルーター冗長性グループなどを監視するようにも設定できます。

メモ

ほとんどの場合，インタフェースに接続されたポーリングによってだけ，十分に正確な根本原因分析ができます。監視対象インタフェースのセットを拡張すると，ポーリングのパフォーマンスに影響が及ぶおそれがあります。

NNMi がハイパーバイザーネットワーク環境を監視している場合は，さらに次のものを含むオブジェクトも監視されます。

- ハイパーバイザー
- ハイパーバイザーでホストされている仮想マシン（VM）
- 仮想スイッチ
- アップリンク（インタフェースオブジェクトとして表される）

ヒント

- 仮想マシンに VMware Tools がインストールされていることを確認し，NNMi によって提供されている仮想マシンノードグループを使用して，仮想マシンに関連付けられている IP アドレスの障害ポーリングを有効にしてください。基盤となる仮想マシンが削除された場合や NNMi が管理できないハイパーバイザーに移動された場合にも，NNMi がすべての VM ノードを特定できるようにするには，この方法を実践することをお勧めします。障害ポーリングを有効にする方法の詳細については，NNMi ヘルプ「管理」の「デフォルト設定」を参照してください。
- 仮想マシン（VM） に関連付けられている IP アドレスに対して障害ポーリングを有効にするには，NNMi が提供している仮想マシンノードグループを使用してください。基盤となる仮想マシンが削除された場合や NNMi が管理できないハイパーバイザーに移動された場合にも，NNMi がすべての VM ノードを特定できるようにするには，この方法を実践することをお勧めします。詳細については，NNMi ヘルプ「管理」の「スケジュールを設定する」のヘルプトピック記載されている「応答のないノードを削除するかどうか設定する」を参照してください。

モニタリングの詳細については，NNMi ヘルプを参照してください。「7.2.5　モニタリングの拡張」も参照してください。

7.2.3　監視の停止

NNMi 管理モードを使用して，デバイスまたはインタフェースを［管理対象外］または［サービス停止中］に設定できます。［管理対象外］は恒久的な状況と見なされます。オブジェクトのステータスを知る心配をする必要はありません。［サービス停止中］は一時的な状況と見なされます。 1 つ以上のオブジェクトがオフラインになり，停止中のインシデントが過剰になります。

すべてのグループ設定全体のオーバーレイとして，管理モードを考えてください。グループ，ポーリング間隔，種類に関係なく，オブジェクトのステータスが［管理対象外］または［サービス停止中］に設定されている場合，State Poller はそのオブジェクトと通信しません。

ヒント

検出を行い，データベースに配置することを選択したデバイスやインタフェース（またはその両方）の中には，ポーリングの必要がないものもあります。［管理対象外］に恒久的に設定する

オブジェクトに注意してください。1 つ以上のノードグループを作成し，管理モードを簡単に設定することもできます。

7.2.4　監視されないノードへのインタフェース

直接管理していないデバイスに接続されているインタフェースのステータスを知る必要があることがあります。例えば，アプリケーションまたはインターネットサーバーへの接続が確立されているかどうか知る必要があるものの，そのサーバーのメンテナンスは担当していないことがあります。検出ルールにそのサーバーを組み入れていないと，NNMi はそのサーバーに接するインタフェースを未接続と見なします。

監視されていないノードに接続する重要なインタフェースのステータスを監視する方法には次の 2 つがあります。

- 監視されていないノードの検出。

  監視されていないノードを NNMi トポロジに追加するとき，NNMi は，トポロジの残りの部分にノードを接続しているインタフェースを接続済みと見なします。この場合，監視設定に従ってこれらのインタフェースをポーリングできます。NNMi はノードを管理対象として検出します。NNMi に監視させたくないノードを非管理対象にしてください。

  メモ

  検出された各ノードは，そのノードを管理しているかどうかにかかわらず，ライセンスの最大数まで数えられます。

- 未接続インタフェースのポーリング。

  未検出ノードの接続を含むネットワークデバイスのノードグループを作成できます。次に，ノードグループの未接続インタフェースのポーリングを有効にします。

  NNMi は，ノードグループのデバイス上のインタフェースをすべてポーリングするので，多数のインタフェースのあるデバイスに対するトラフィックが大量に追加されます。

7.2.5　モニタリングの拡張

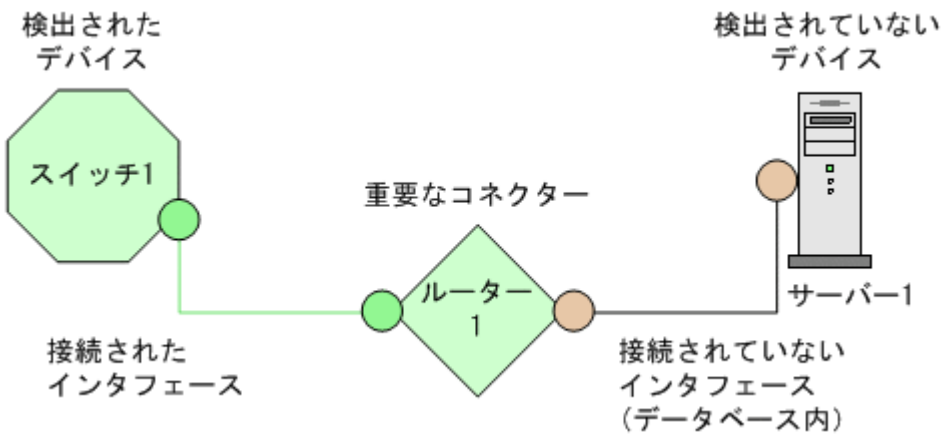
監視を拡張して，次が含まれるようにできます。

- 未接続インタフェース。

  デフォルトでは，NNMi が監視する未接続インタフェースは IP アドレスのあるものだけであり，**ルーター**ノードグループに含まれます。

メモ

NNMi は，次のように，NNMi が検出した別のデバイスに接続されていないインタフェースとして未接続インタフェースを定義します。



- ルーターインタフェースのように，IP アドレスのあるインタフェース。
- SNMP をサポートしないデバイス用の ICMP ポーリング。
  デフォルトで ICMP ポーリングは，**非 SNMP デバイス**ノードグループについて有効です。

7.2.6 ノードグループとインタフェースグループを作成する

ノードグループとインタフェースグループをセットアップしてから，監視を設定する必要があります。したがって，ノードグループとインタフェースグループを設定するときはポーリング要求について考慮します。重要なデバイスを頻繁に監視できるようにノードグループとインタフェースグループを設定するのが理想的です。クリティカルでないデバイスをチェックする場合，チェック回数を減らすこともできます。

ヒント

ネットワークを監視するノードおよびインタフェースグループのセットを 1 つ設定します。マップでネットワークの可視化用に異なるノードグループのセットを設定します。

これらグループは，**[設定]** ＞ **[オブジェクトグループ]** ＞ **[ノードグループ]** または **[設定]** ＞ **[オブジェクトグループ]** ＞ **[インタフェースグループ]** ワークスペースを使用して定義します。これらグループは，デフォルトで，インシデント，ノード，インタフェース，およびアドレスビューをフィルタするのに使うのと同じグループです。監視設定用に，別のノードフィルタまたはインタフェースフィルタ定義を作成するには，ノードグループまたはインタフェースグループを開き，**[ノードグループ]** フォームまたは **[インタフェースグループ]** フォームで **[ビューフィルターリストに追加]** チェックボックスをオンにします。**[保存して閉じる]** をクリックして定義を保存します。

［モニタリングの設定］フォームの［ノードの設定］タブと［インタフェースの設定］タブにあるノードグループまたはインタフェースグループのレベルで，ポーリングの種類とポーリングの間隔を設定します。

類似のポーリングのニーズごとに，インタフェースやデバイス（またはその両方）をグループにまとめる基準を決定します。計画作成に際して考慮する必要がある要因は次のとおりです。

- ネットワークのどのエリアにこれらのデバイスがありますか？タイミング制限がありますか？
- デバイスの種類ごとにポーリング間隔または収集するデータを変更しますか？インタフェースの種類ごとに変更しますか？
- NNMi が提供する事前設定されたグループを使用できますか？

> **ヒント**
>
> 同時にサービス停止中になりそうなオブジェクトのグループ定義を，場所ごとまたはそのほかの基準ごとに作成できます。例えば，IOS アップグレードを適用する間は，すべての Cisco ルーターをサービス停止中モードにできます。

(1) インタフェースグループ

基準に基づいて，どのインタフェースグループを作成するか決定します。インタフェースグループが最初に評価されることを覚えておいてください（「7.1 ステータスポーリングの概念」参照）。インタフェースグループはノードグループのメンバーを参照できるので，インタフェースグループの設定を実施する前に，ノードグループの設定を完了した方がよいケースもあります。

事前設定されたインタフェースグループ

NNMi には，設定済みの便利なインタフェースグループが幾つかあります。例えば，次のとおりです。

- ISDN 接続に関連づけられた IFType のある全インタフェース
- 音声接続用のインタフェース
- ポイントツーポイント通信用のインタフェース
- ソフトウェアループバックインタフェース
- VLAN インタフェース
- リンクアグリゲーションプロトコルに関与するインタフェース

既存のグループを使用するか，それらを変更するか，または自分専用のグループを作成できます。

インタフェースグループには次の 2 種類の設定項目があります。つまり，所属するノードが含まれるノードグループとインタフェースの IFType またはほかの属性です。これらは次のように組み合わせられます。

- IFType と無関係に，ノードグループ内のノードのすべてのインタフェースをグループにまとめる。IFType または属性（名前，エイリアス，説明，速度，インデックス，アドレス，またはそのほかの IFType 属性など）は選択しない。

- インタフェースが存在するノードに関係なく，特定の IFType または属性のすべてのインタフェースをグループにまとめる。
- 特定のノードグループに存在する特定の IFType または属性のインタフェースだけをグループにまとめる。

(2) ノードグループ

インタフェースグループの計画を作成してから，ノードグループの計画を作成します。監視用に作成されたノードグループがビューのフィルタに意味があるとは限らないので，それらは個別に設定できます。

事前設定されたノードグループ

設定作業を簡単にするために，ノードグループのデフォルト集合を用意しています。これらの基礎になっているのは，検出プロセスの間にシステムオブジェクト ID から導出されたデバイスカテゴリです。デフォルトのノードグループには次が含まれます。

- ルーター
- ネットワーキングインフラストラクチャデバイス（スイッチ，ルーターなど）
- Microsoft Windows システム
- SNMP コミュニティ文字列がわからないデバイス
- 重要ノード。Causal Engine によって内部的に使用されており，コネクタ障害の危険にさらされているデバイスの特殊処理を提供します。詳細については，NNMi ヘルプの「定義済ビューフィルターとして使用されるノードグループ」を参照してください。
- 仮想マシン

既存のグループを使用するか，それらを変更するか，または自分専用のグループを作成できます。

次のノード属性を使用して，関連するノードの定義に条件を付けることができます。

- ノード上の IP アドレス
- ホスト名のワイルドカード抽出
- デバイスプロファイルから得られる情報（例えば，カテゴリ，ベンダー，ファミリ）
- MIB II sysName，sysContact，sysLocation

使われない余分なエントリがリストに追加されないように，設定および表示用に豊富なグループのセットを作成し，バランスを取ってください。

ヒント

シンプルで再使用可能な小さいグループを作成し，監視または視覚化のためにこれらを組み合わせて，階層的なまとまりにできます。例えば，「すべてのルーター」と「IP アドレスの末尾が 100 のすべてのシステム」のように，グループ定義は重なることがあります。ノードは複数のグループに属することがあります。

デバイスプロファイルとの相互作用

各デバイスが検出されると，NNMi はシステムオブジェクト ID を使用して，使用可能なデバイスプロファイルのリストを検索します。デバイスプロファイルは，ベンダー，製品，ファミリ，デバイスカテゴリなど，デバイスの追加属性を導出するために使用されます。

ノードグループを設定するとき，これら導出された属性を使用して，監視設定に適用するデバイスをカテゴリにまとめられます。例えば，ベンダーを問わずに，ネットワーク全体のすべてのスイッチを特定のポーリング間隔でポーリングすることもできます。デバイスカテゴリの「スイッチ」を自分のノードグループの定義特性として使えます。システムオブジェクト ID がカテゴリ「スイッチ」にマップされるデバイスはすべて，そのノードグループの設定が反映されます。

ヒント

NNMi がハイパーバイザーネットワーク環境を管理している場合は，仮想マシン(VM) だけが含まれるノードグループを作成できます。これらのノードは，vmwareVM デバイスプロファイルを使用して識別できます。このノードグループを使用すると，ハイパーバイザーでホストされなくなった VM がないかをチェックすることもできます。このノードグループを選択したあと，Hosted On = null でフィルターし，これらの VM を特定します。このノードグループを使用して，VM に関連付けられている IP アドレスの障害ポーリングを有効にすることもできます。これは，関連付けられたハイパーバイザーが削除されている場合でも VM を継続的に監視できるようにするベストプラクティスでもあります。

7.2.7 ポーリング間隔を計画する

NNMi がデータを収集するのに使うポーリング間隔をオブジェクトグループごとに，選択します。サービスレベル契約条項に一致するように，間隔は 1 分間と短くすることもできますし，数日間と長くすることもできます。

ヒント

間隔が短いと，迅速にネットワーク問題を認識するのに役立ちます。しかし，あまりに短い間隔であまりに多くのオブジェクトをポーリングすると，State Poller にバックログを発生させる可能性があります。リソース利用と間隔の間でお使いの環境にとって，最良のバランスを見つけてください。

メモ

根本原因分析エンジンは，24 時間に一回ステータスポーリングを実施し，ステータス，結果およびインシデントの情報を更新します。このステータスポーリングは，デバイスに設定されたポーリング周期には影響しません。

7.2.8　収集するデータを計画する

State Poller サービスは，ポーリングを使って，ネットワークで監視されているデバイスに関する状態情報を収集します。ポーリングは ICMP や SNMP（またはその両方）を使って実行できます。

ICMP（ping）

ICMP アドレス監視は，ping 要求を使って，管理対象の各 IP アドレスが使用可能かどうかを確認します。

SNMP ポーリング

SNMP 監視は，監視されている各 SNMP エージェントが SNMP クエリーに応答していることを確認します。

- State Poller は，間隔ごとに 1 つのクエリーで監視されている各オブジェクトから，設定済みの SNMP 情報を収集するよう最適化されています。設定の変更をすると，State Poller は各オブジェクトのグループメンバーシップを再計算し，収集する間隔とデータセットに再適用します。
- SNMP 監視は，監視されているすべてのインタフェースとコンポーネントに SNMP クエリーを発行し，MIB II インタフェーステーブル，HostResources MIB，およびベンダー固有 MIB から現在の値を要求します。障害監視に使われる値もあります。

Web ポーリング

（SNMP エージェントに加えて）Web エージェントが設定されている場合，NNMi は追加のプロトコルを使用できます。例えば，VMware 環境用の SOAP プロトコルなどです。

SNMP コンポーネント稼働状態データ

コンポーネントヘルス監視をグローバルなレベルで有効または無効にできます。障害に関するコンポーネント稼働監視は，デバイスの障害ポーリング間隔設定に従います。

ポーリングごとに追加データを収集しても，ポーリングの実行時間への影響はありません。しかし，各オブジェクトに関して格納される追加データによって，State Poller 用に必要なメモリ容量が増加する可能性があります。

ヒント

監視設定変更をまとめて実施すると，State Poller の進行中の操作への影響を少なくできます。

7.2.9　SNMP トラップが NNMi に送信する内容を決定する

NNMi は，SNMP トラップを受信したとき，次のポーリング間隔を待つのではなく，デバイスのポーリングに次の SNMP トラップを使用します。

- CempMemBufferNotify
- CiscoColdStart

- CiscoEnvMonFanNotification
- CiscoEnvMonFanStatusChangeNotif
- CiscoEnvMonRedundantSupplyNotification
- CiscoEnvMonSuppStatusChangeNotif
- CiscoEnvMonTemperatureNotification
- CiscoEnvMonTempStatusChangeNotif
- CiscoEnvMonVoltageNotification
- CiscoEnvMonVoltStatusChangeNotif
- CiscoFRUInserted
- CiscoFRURemoved
- CiscoLinkDown
- CiscoLinkUp
- CiscoModuleDown
- CiscoModuleUp
- CiscoModuleStatusChange
- CiscoRFProgressionNotif
- CiscoRFSwactNotif
- CiscoWarmStart
- HSRPStateChange
- IetfVrrpStateChange
- Rc2kTemperature
- RcAggLinkDown
- RcAggLinkUp
- RcChasFanDown
- RcChasFanUp
- RcChasPowerSupplyDown
- RcChasPowerSupplyUp
- Rcn2kTemperature
- RcnAggLinkDown
- RcnAggLinkUp
- RcnChasFanDown
- RcnChasFanUp

- RcnChasPowerSupplyDown
- RcnChasPowerSupplyUp
- RcnSmltIstLinkDown
- RcnSmltIstLinkUp
- RcSmltIstLinkUp
- RcVrrpStateChange
- SNMPColdStart
- SNMPLinkDown
- SNMPLinkUp
- SNMPWarmStart

トラップを受信したときに NNMi にデバイスをポーリングさせるには，これらのトラップを NNMi に送信するようにネットワークデバイスを設定します。

メモ

SNMP トラップインシデント設定の詳細については，NNMi コンソールから，［設定］ワークスペースに移動し，［インシデント］＞［SNMP トラップの設定］の順に選択します。

「(5)　SNMP トラップからの検出ヒント」も参照してください。

7.3　ステータスポーリングの設定

この節では，設定のヒントを示し，設定例を幾つか挙げます。この節を読んだあと，特定の手順については，NNMi ヘルプの「NNMi モニタリング動作を設定する」を参照してください。

> **メモ**
>
> 大幅な設定変更を行う前には，既存の設定のコピーを保存しておくことをお勧めします。詳細については，「4.2　ベストプラクティス：既存の設定を保存する」を参照してください。

7.3.1　監視するインタフェースグループとノードグループを設定する

ポーリングにノードグループとインタフェースグループを使用すべき理由については，前のセクション「7.2.6　ノードグループとインタフェースグループを作成する」を参照してください。

NNMi コンソールまたは CSV ファイルを使用して，ノードグループまたはインタフェースグループを作成できます。例えば，ノードグループ情報が Microsoft Excel ワークシートにある場合，この情報をCSV ファイルとして保存してから，nnmloadnodegroups.ovpl コマンドを使用して，NNMi に追加できます。同様に，nnmloadinterfacegroups.ovpl コマンドを使用して，インタフェースグループ情報を NNMi に追加できます。詳細については，nnmloadnodegroups.ovpl およびnnmloadinterfacegroups.ovpl のリファレンスページを参照してください。

NNMi コンソールでノードグループおよびインタフェースグループを作成するには，**［設定］** ワークスペースを使用します。詳細については，NNMi ヘルプの「ノードまたはインタフェースのグループ作成」を参照してください。

（例）

ProximiT プロキシサーバー用にノードグループを設定する方法は次のとおりです。

1. **［設定］** ＞ **［オブジェクトグループ］** ＞ **［ノードグループ］** を開き，**［新規作成］** をクリックする。
2. グループProxy Servers という名前を付け，**［ビューフィルターリストに追加］** をオンにする。
3. **［追加のフィルター］** タブで，hostname 属性を選択し，演算子の設定をlike にする。
4. ノードのホスト名にprox*.yourdomain.com として入力し，**［保存して閉じる］** をクリックする。
   値は，prox*.example.com のようにワイルドカードを入力します。
   ProximiT デバイスについて Device Profile（デバイスプロファイル）と Category（カテゴリ）を設定してある場合は，**［デバイスフィルター］** タブを使って **［デバイスのカテゴリ］** の選択個所にアクセスし，作成した Proxy Server カテゴリをグループのベースにできます。
5. グループ定義で **［保存して閉じる］** をクリックする。

メモ

インタフェースグループ設定で参照する前に，ノードグループを設定する必要があります。

7.3.2　インタフェースの監視を設定する

State Poller は，ノードグループの前にインタフェースグループのメンバーを分析します。作成した各インタフェースグループ，および使用する既存のインタフェースグループについて，［モニタリングの設定］フォームの［インタフェースグループの設定］タブを開き，State Poller がそのグループを処理する方法に関する個別の設定を作成します。設定には次のものが含まれます。

- 障害モニタリングの有効化または無効化
- 障害ポーリング間隔の設定
- NNMi がグループ内の未接続インタフェース（または IP アドレスをホストしている未接続インタフェース）を監視するかどうかの選択

インタフェースグループごとに異なる設定ができます。State Poller は，小さい順序番号から順にリストを評価します。

ヒント

複数のグループにあてはまるオブジェクトは順序番号の小さいグループから設定を適用されることを頭に入れておきつつ，順序番号をダブルチェックします。

7.3.3　ノードの監視を設定する

あるオブジェクトが設定済みのインタフェースグループにあてはまらない場合，State Poller はノードグループ内のメンバーシップについて，そのオブジェクトを評価します。設定は小さい順序番号から順に評価し，最初に合致するノードグループに適用されます。

ノードグループごとに，［モニタリングの設定］フォームを開いてから［ノードの設定］タブを開きます。State Poller がそのグループを処理する方法に関する個別の設定を作成します。設定には次のものが含まれます。

- 障害モニタリングの有効化または無効化
- 障害ポーリング間隔の設定
- NNMi がグループ内の未接続インタフェース（または IP アドレスをホストしている未接続インタフェース）を監視するかどうかの選択

ノードグループごとに異なる設定ができます。

ヒント

複数のグループにあてはまるオブジェクトは順序番号の小さいグループから設定を適用されることを頭に入れておきつつ，順序番号をダブルチェックします。

7.3.4　監視のデフォルトを設定する

State Poller は，定義済みのインタフェースの設定またはノードの設定に合致しないオブジェクトについて［デフォルト設定］タブの設定を適用します。このタブの設定を検討し，デフォルトレベルで自分の環境に合致することを確認します。例えば，デフォルト設定としてすべての未接続インタフェースをポーリングすることはほとんどないでしょう。

メモ

変更を有効にするためには，コンソールに戻るまでに，すべての［モニタリングの設定］フォームを必ず［保存して閉じる］ようにしてください。

7.4　ステータスポーリングの評価

この節では，監視設定の進行と成功を評価する方法を説明します。

7.4.1　ネットワーク監視の設定を確認する

NNMi が指定のノードまたはインタフェースの監視に使う設定をすると，ステータスポーリングをいつでも開始できます。

ネットワーク監視の設定を確認するには，次の点をチェックします。

- (1)　インタフェースまたはノードは正しいグループのメンバーでしょうか？
- (2)　どの設定が適用されていますか？
- (3)　どのデータが収集されていますか？

(1)　インタフェースまたはノードは正しいグループのメンバーでしょうか？

あるグループにどのインタフェースまたはノードが属するか確認するには，［設定］ワークスペースで次の 1 つを選択します。

- ノードグループ
- インタフェースグループ

ヘルプの指示に従って，グループのメンバーを表示します。オブジェクトは複数のグループのメンバーになれること，ほかのグループの順序番号の方が小さい可能性があることを頭に入れておいてください。

その代わりに，オブジェクト（インタフェースまたはノード）を開き［ノードグループ］タブまたは［インタフェースグループ］タブをクリックして，オブジェクトが属するグループの完全なリストを表示することもできます。このリストは，グループ名でソートされているため，どの設定が適用されるかを決定する順序番号とは関係ありません。

オブジェクトがグループのメンバーでない場合は次のとおりです。

1. ［インベントリ］ビューで，ノードのデバイスプロファイルを調べる。
2. ［設定］＞［デバイスのプロファイル］で，そのデバイスプロファイルに関する属性の情報を確認する。
3. ノードグループ定義の属性要件を確認する。
   不一致がある場合は，［デバイスのプロファイル］のカテゴリを修正して，その種類のデバイスがノードグループに当てはまるようにできます。ノードの属性を更新してグループに一致させるためには，［アクション］＞［ポーリング］＞［設定のポーリング］を実行する必要があります。

(2) どの設定が適用されていますか？

特定のノード，インタフェース，またはアドレスに有効な監視設定をチェックするには，該当する［インベントリ］ビュー内のそのオブジェクトを選択し，［アクション］＞［設定の詳細］＞［モニタリングの設定］を選択します。NNMi に現在の監視設定が表示されます。

［障害 SNMP ポーリングの有効化］と［障害のポーリング間隔］の値を調査します。これらの値が予想どおりでない場合は，［ノードグループ］または［インタフェースグループ］の値を見て，どのグループが適用されるか調べます。

オブジェクトに対する通信が無効にされていないことを確認するために，オブジェクトの［アクション］＞［設定の詳細］＞［通信の設定］をチェックします。

(3) どのデータが収集されていますか？

特定のデバイスのステータスポーリングを開始し，予想された種類のポーリング（SNMP，ICMP）がそのデバイスについて実行されていることを確認できます。

ヒント

（SNMP エージェントに加えて）Web エージェントが設定されている場合，NNMi は追加のプロトコルを使用できます。例えば，VMware 環境用の SOAP プロトコルなどです。

ノードを選択し，［アクション］＞［ポーリング］＞［ステータスのポーリング］をクリックします。NNMi はデバイスのリアルタイムのステータスチェックを実行します。実行中のポーリングの種類と結果が出力されます。ポーリングの種類が予想したものでない場合は，ノードの監視設定，および監視設定のそれぞれのグローバル，インタフェース，またはノードに関する設定をチェックします。

7.4.2 ステータスポーリングのパフォーマンスの評価

自分の環境のステータスポーリングのパフォーマンスを評価するには，State Poller 稼働状態チェックの情報を使って，State Poller サービスの動作を数値で表し，評価します。

State Poller 稼働状態情報は，Status Poller がポーリング要求に応じることができるかどうかを示します。

(1) State Poller は最新の状態が反映されていますか？

次の表に説明されているように，［システム情報］ウィンドウの［ステートポーラー］タブで State Poller サービスの現在の稼働状態情報をいつでもチェックできます。

表 7-1 State Poller 稼働状態情報

情報	説明
ステータス	State Poller サービスの全般的なステータス

情報	説明
ポーリングカウンタ	• 過去 5 分以内に要求された収集 • 過去 5 分以内に完了した収集 • 処理中の収集 • 収集要求の遅延
過去 5 分以内にスキップを実行した時刻	設定済みのポーリング間隔内で完了しなかった定期的に実行するポーリングの数。値が 0 でない場合は，ポーリングエンジンの処理が追いついていないか，または応答が戻ってくるまえにポーリングが実施されています。 • 監視する必要があるもの：この値が増加し続ける場合は，ターゲットとの通信に問題があるか，または NNMi の負荷が過剰です。 • 実行する必要があるアクション：nnm.log ファイルで文字列com.hp.ov.nms.statepoller で始まるクラスのメッセージを探して，スキップされたポーリングのターゲットを特定します。 スキップされたポーリングのターゲットが同じ場合，設定を変更して，これらのターゲットのポーリング頻度を低くするか，またはタイムアウトを増やします。 スキップされたポーリングのターゲットが異なる場合，NNMi のシステムパフォーマンス（特に ovjboss の使用可能メモリ）を確認します。
過去 5 分以内の古い収集	古い収集というのは，少なくとも 10 分間，ポーリングエンジンから応答を受信していない収集のことです。稼働状態が良好なシステムでは古い収集はありません。 • 監視する必要があるもの：この値が一定して増加する場合は，ポーリングエンジンに問題があります。 • 実行する必要があるアクション：nnm.log ファイルで文字列com.hp.ov.nms.statepoller で始まるクラスのメッセージを探して，古い収集のターゲットを特定します。 古い収集のターゲットが 1 つの場合，この問題を解決できるまでターゲットを管理から除きます。 古い収集のターゲットが異なる場合，NNMi システムと NNM データベースのパフォーマンスを確認します。NNMi を停止して再起動します。
ポーラーの結果キューの長さ	• 監視する必要があるもの：値が 0 または 0 に近いことを確認してください。0 よりも大きい場合，次のアクションを実行してください。 • 実行する必要があるアクション：キューのサイズがきわめて大きい場合，ovjboss はメモリ領域不足の可能性があります。
状態マッパーキュー期間	• 監視する必要があるもの：値が 0 または 0 に近いことを確認してください。0 よりも大きい場合，次のアクションを実行してください。 • 実行する必要があるアクション：このキューのサイズがきわめて大きい場合は，NNMi システムと NNMi データベースのパフォーマンスをチェックします。
状態アップデータキュー期間	• 監視する必要があるもの：値が 0 または 0 に近いことを確認してください。0 よりも大きい場合，次のアクションを実行してください。 • 実行する必要があるアクション：このキューのサイズがきわめて大きい場合は，NNMi システムと NNMi データベースのパフォーマンスをチェックします。
状態アップデータ例外	監視の要点：この値は 0 になるはずです。

7.5 ステータスポーリングの調整

ステータスポーリングのパフォーマンスは次の重要な変数の影響を受けます。

- ポーリングされるデバイス／インタフェースの数
- 設定されるポーリングの種類
- 各デバイスのポーリングの頻度

これらの変数は，ネットワーク管理のニーズによって決まります。ステータスポーリングについてパフォーマンス上の問題がある場合は，次の設定を確認してください。

- 個別のノードのポーリング設定はノードグループとインタフェースグループ内のメンバーシップによって制御されるので，類似のポーリング要求のあるノードまたはインタフェースがグループに含まれていることを確認します。
- 未接続インタフェースまたは IP アドレスをホストするインタフェースをポーリングしている場合は，設定をチェックして，必要なインタフェースだけをポーリングしていることを確認します。特別な制御を用意し，最小のインタフェースのサブセットを選んでポーリングするために，（**[モニタリングの設定]** フォームの **[デフォルト設定]** にではなく）**[ノードの設定]** フォームまたは **[インタフェースの設定]** フォームでこれらのポーリングを有効にしてください。
- 未接続インタフェースのポーリングでは，未接続のすべてのインタフェースが監視されることを覚えておいてください。IP アドレスのある未接続のインタフェースだけを監視するには，IP アドレスをホストするインタフェースのポーリングを有効にします。

監視設定とは無関係に，ステータスポーリングは，ネットワーク応答性に左右され，全般的なシステムパフォーマンスの影響を受ける可能性があります。デフォルトのポーリング間隔でのステータスポーリングは多くのネットワーク負荷を掛けませんが，NNMi サーバーとポーリングされているデバイスの間のネットワークリンクのパフォーマンスが低い場合，ステータスポーリングのパフォーマンスも低くなる可能性があります。タイムアウトを大きくし，再試行の数を小さく設定すると，ネットワーク負荷を低減できますが，これらの設定変更はあまり効果がないかもしれません。タイミングの良いポーリングを行うには，適切なネットワークパフォーマンスと十分なシステムリソース（CPU，メモリ）が必要です。

コンポーネント稼働状態監視を有効または無効にしても，ポーリングのタイミングには影響がありません。スケジュールされた時刻に，追加の MIB オブジェクトが収集されるだけです。ただし，コンポーネントヘルス監視を無効にすると，State Poller が使用するメモリの量が減少する可能性があります。

8 NNMi インシデント

NNMi には，多数のデフォルトインシデントと相関処理が用意されています。デフォルトインシデントを利用すると，NNMi コンソールにすぐにインシデントを表示できます。また，相関処理を利用すると，インシデントを管理する数を減らすことができます。この章では，NNMi インシデントを設定することでネットワーク管理を微調整するのに役立つ情報を説明します。この章は，NNMi ヘルプの情報を補充するものです。

NNMi インシデントの概要およびインシデント設定方法の詳細については，NNMi ヘルプの「インシデントを設定する」を参照してください。

8.1　インシデントの概念

NNMi では，次のソースからネットワークステータス情報が収集されます。

- NNMi の Causal Engine ではネットワークの稼働状態が分析され，継続的に各デバイスの稼働状態ステータス値が提供されます。Causal Engine では，可能な場合は常にネットワーク障害の根本原因も広範囲に評価され，決定されます。
- ネットワークデバイスからの SNMP トラップ。NNMi の Causal Engine は，分析中にトラップを症状に関する情報として使用します。

NNMi は，これらの情報をネットワーク管理に有用な情報を提供するネットワークステータス情報に変換します。NNMi には，ネットワークオペレータが考慮する必要があるインシデント数を減らす多くのデフォルトインシデント相関処理が用意されています。

デフォルトのインシデント相関処理をカスタマイズして，環境のネットワーク管理要件に一致する新規インシデント相関処理を作成できます。

NNMi コンソールのインシデント設定によって，NNMi が作成できるインシデントタイプが定義されます。インシデント設定が受信した SNMP トラップと一致しない場合，その情報は廃棄されます。ソースオブジェクトの管理モードが，NNMi データベースで［非管理対象］もしくは［サービス停止中］に設定されている場合，またはデバイスが障害ポーリングで監視されていない場合，NNMi では常に受信トラップは廃棄されます。

`nnmtrapconfig.ovpl -dumpBlockList` は，インシデント設定がないか，または無効なため，インシデントパイプラインに渡されなかった SNMP トラップなど，現在のインシデント設定に関する情報を出力します。

さらに，NNMi では NNMi トポロジにないネットワークデバイスからの SNMP トラップは廃棄されます。このデフォルト動作の変更の詳細については，NNMi ヘルプの「未解決の受信トラップを処理する」を参照してください。

詳細については，NNMi ヘルプの「NNMi によるインシデントの収集方法」を参照してください。

8.1.1　インシデントライフサイクル

次の表は，インシデントのライフサイクルの段階を説明したものです。

表 8-1　NNMi インシデントライフサイクル

ライフサイクル状態	説明	状態設定者	インシデント使用者
なし	NNMi イベントパイプラインはすべてのソースから入力を受領し，必要に応じてインシデントを作成します。	該当なし	• NNMi

ライフサイクル状態	説明	状態設定者	インシデント使用者
ダンプニング済み	インシデントは保管場所にあり，別のインシデントとの相関処理待ちです。インシデントビューアのインシデントを減らすために，この待機期間があります。 ダンプニング周期はインシデントタイプによって異なります。詳細については，「8.1.7　インシデントの抑制，強化，およびダンプニング」を参照してください。	NNMi	• NNMi
登録済み	インシデントは，インシデントビューで見ることができます。 インシデントは任意の設定済み宛先へ転送されます（近隣またはグローバルマネージャー）。	NNMi ユーザーはインシデントビューでこの状態を設定することもできます。	• ユーザー • ライフサイクル移行アクション
進行中	インシデントは問題を調査するユーザーに割り当てられています。 ネットワーク管理者によってこの状態の特定の意味が定義されます。	ユーザー	• ユーザー • ライフサイクル移行アクション
完了	インシデントによって指定された問題は，対処が完了し，解決しています。 ネットワーク管理者によってこの状態の特定の意味が定義されます。	ユーザー	• ユーザー • ライフサイクル移行アクション
解決済み	このインシデントによってレポートされた問題が解決したことを NNMi が確認したことを示します。例えば，デバイスからインタフェースを取り外すと，そのインタフェースに関するインシデントはすべて，自動的に「解決済み」になります。	ユーザーまたは NNMi	• ユーザー • ライフサイクル移行アクション

8.1.2　トラップおよびインシデント転送

次の表は，トラップおよびインシデントを NNMi 管理サーバーから別の宛先へ転送する方法を要約したものです。

表 8-2　トラップおよび NNMi インシデント転送でサポートされている方法

項目	NNMi トラップ転送	NNMi Northbound インタフェーストラップ転送	グローバルネットワーク管理のトラップ転送
転送対象	ネットワークデバイスからの SNMP トラップ	ネットワークデバイスからの SNMP トラップ	ネットワークデバイスからの SNMP トラップ
転送フォーマット	受信したままの SNMPv1，SNMPv2c，または SNMPv3 トラップ（SNMPv3 トラップは SNMPv2c トラップへ変換可能）	NNMi インシデントから作成された SNMPv2c トラップ	NNMi インシデント
追加情報	ほとんどの場合，NNMi は varbind を追加して元のソースオブジェクトを識別します。 NNMi が SNMPv1 トラップを変更することはありません。	NNMi は varbind を追加して元のソースオブジェクトを識別します。	リージョナルマネージャープロセスによってインシデントに追加された情報はすべて，転送済みインシデントに保持されます。
設定先	［設定］ワークスペースの［インシデント］＞［トラップサーバー］＞［トラップ転送設定］	［統合モジュールの設定］ワークスペースの［Northbound インタフェース］	［SNMP トラップの設定］フォームまたは syslog 設定の［グローバルマネージャーへの転送］タブ
注	－	－	グローバルマネージャーのインシデントビューに表示されるリモートインシデントを転送します。転送済みインシデントはグローバルマネージャー上での相関処理に参加します。
詳細情報	NNMi ヘルプの「トラップ転送を設定する」	－	NNMi ヘルプの「SNMP トラップインシデントのグローバルマネージャーへの転送を設定する (NNMi Advanced)」

（凡例）　－：該当なし。

8.1.3　受信済み SNMP トラップ

NNMi が管理デバイスから受信する SNMP トラップを別のアプリケーションに転送する場合は，次のどちらかの方法を使用します。

- NNMi SNMP トラップ転送を使用します。
- NNMi Northbound インタフェースの SNMP トラップ転送メカニズムを使用します。

受信側アプリケーションがトラップを識別する方法は次のように異なります。

- Windows（すべて）および Linux（元のトラップではない場合）

デフォルトおよび SNMPv3 から SNMPv2c への変換転送オプションに該当します。

Windows NNMi 管理サーバー上の NNMi SNMP トラップ転送メカニズムによって，送信先へ転送する前に各 SNMP トラップが改編されます。トラップは NNMi 管理サーバーからのものと考えられます（この情報は，［**トラップ転送先**］フォームで元のトラップ転送オプションが選択されていない Linux NNMi 管理サーバーにも適用されます）。

トラップ送信元デバイスと受信するアプリケーションでのイベントとの関連づけを正しくするため，これらのトラップに関するルールを，追加される varbind によってカスタマイズする必要があります。originIPAddress(.1.3.6.1.4.1.11.2.17.2.19.1.1.3)varbind からの値を解釈します。originIPAddress の値は汎用タイプ InetAddress のバイト文字列で，originIPAddressType(.1.3.6.1.4.1.11.2.17.2.19.1.1.2)varbind の値によって決まる InetAddressIPv4 または InetAddressIPv6 です。ルールによって originIPAddressType varbind を読み取って，originIPAddress varbind のインターネットアドレスタイプ（ipv4(1)，ipv6(2)）の値を決定する必要があります。ルールによって originIPAddress の値を表示文字列に変換する必要もあります。

NNMi が転送されたトラップに追加する varbind の詳細については，NNMi ヘルプの「NNMi が提供するトラップ varbinds」，RFC2851 および次のファイルを参照してください。

- Windows：`%NNM_SNMP_MIBS%¥Vendor¥Hewlett-Packard¥hp-nnmi.mib`
- Linux：`$NNM_SNMP_MIBS/Vendor/Hewlett-Packard/hp-nnmi.mib`

- 元のトラップ転送が設定された Linux

  Linux NNMi 管理サーバーによって，NNMi が受信するものと同じフォーマットでトラップを転送できます。各トラップは管理対象デバイスがトラップ転送先に直接送信したように表示されるため，受信するアプリケーションに設定された既存のトラップ処理は変更なしで動作します。

- NNMi Northbound インタフェース（全オペレーティングシステム）

  NNMi Northbound インタフェースは各 SNMP トラップを強化してから，トラップ転送先に転送します。トラップは NNMi 管理サーバーからのものと考えられます。受信側アプリケーションのトラップ送信デバイスとイベント間の関連づけを正しくするため，これらのトラップのルールを収集した varbind に対してカスタマイズする必要があります。nnmiIncidentSourceNodeHostname(1.3.6.1.4.1.11.2.17.19.2.2.21)および nnmiIncidentSourceNodeMgmtAddr(1.3.6.1.4.1.11.2.17.19.2.2.24)varbind によって元のソースオブジェクトが識別されます。

8.1.4 MIB

NNMi では，次の管理情報ベース（MIB）ファイルを NNMi データベースにロードする必要があります。

- カスタムポーラー機能，折れ線グラフ，またはその両方の MIB 式で使用するすべての MIB 変数
- NNMi が稼働状態を監視するセンサー（ファン，または電源など）

NNMi では，管理情報ベース（MIB）ファイル，または MIB ファイルで定義されているトラップを NNMi データベースにロードする必要があります。

8.1.5 カスタムインシデント属性

NNMi では，カスタムインシデント属性（CIA）を使用して，インシデントに追加情報が追加されます。

- SNMP トラップインシデントの場合，NNMi では元のトラップ varbind はインシデントの CIA として格納されます。
- 管理イベントインシデントの場合，NNMi では関連情報（com.hp.ov.nms.apa.symptom など）はインシデントの CIA として追加されます。

インシデント CIA を使用すると，インシデントライフサイクル移行アクション，抑制，重複削除，強化などの範囲を絞り込むことができます。CIA を使用して，インシデントビューまたはフォームのアプリケーションメニュー項目の信頼性を絞り込むこともできます。

指定のインシデントに NNMi がどの CIA を追加するかを決定するには，インシデントビューのサンプルインシデントを開き，［カスタム属性］タブの情報を確認します。

(1) 解決済み管理イベントインシデントに追加される CIA

管理イベントインシデントの原因となった状態が該当しなくなったと NNMi Causal Engine が判断すると，NNMi はそのインシデントのライフサイクル状態を［解決済み］に設定し，次の表にリストされている CIA をインシデントに追加します。NNMi コンソールユーザーは，［インシデント］フォームの［相関処理の注］フィールドでこの情報を確認できます。ライフサイクル移行アクションでは，CIA の値が直接使用されることがあります。

表 8-3 解決済みインシデントのカスタムインシデント属性

名前	説明
cia.reasonClosed	NNMi がインシデントをキャンセルしたか解決済みにした理由。 この理由は，NodeUp やInterfaceUp など結果の名前にもなります。このフィールドが設定されていない場合は，NNMi コンソールユーザーがインシデントを解決済みにしたということになります。cia.reasonClosed CIA の NNMi の期待値を判断するには，NNMi ヘルプの「NNMi によるインシデントの解決方法」を参照してください。
cia.incidentDurationMs	機能停止のタイムスタンプ（単位：ミリ秒）。 ステータスが停止中になってから動作中に戻るまで NNMi が測定します。この値は，cia.timeIncidentDetectedMs とcia.timeIncidentResolvedMs の CIA の差です。停止中インシデントと動作中インシデントのタイムスタンプを比較するより正確な測定値です。
cia.timeIncidentDetectedMs	NNMi Causal Engine が最初に問題を検出したときのタイムスタンプ（単位：ミリ秒）。
cia.timeIncidentResolvedMs	問題が解決したことを NNMi Causal Engine が検出したときのタイムスタンプ（単位：ミリ秒）。

NNMi は，多くの一次的根本原因インシデントと二次的根本原因インシデントに，表 8-3 に示した CIA を追加します。例えばNodeDown インシデントには，InterfaceDown インシデントとAddressNotResponding

インシデントが二次的根本原因として含まれることがあります。NNMi がNodeDown インシデントを解決済みにすると，NNMi は二次的インシデントも解決済みにして，それぞれのインシデントのコンテキストの値を含む CIA を二次的インシデントに追加します。

NNMi は，次のデフォルト管理イベントインシデントタイプには表 8-3 に示した CIA を追加しません。

- NNMi コンソールユーザーが手動で解決済みにしたインシデント
- NNMi データベースから削除されたオブジェクトに応答して NNMi が解決済みにしたインシデント
- IslandGroupDown インシデント
- NnmClusterFailover，NnmClusterLostStandby，NnmClusterStartup，NnmClusterTransfer の各インシデント
- 次のファミリのインシデント

  相関処理

  ライセンス

  NNMi 稼働状態

  トラップ分析

8.1.6 インシデント数の削減

NNMi には，ネットワークオペレータが NNMi コンソールで見るインシデント数を削減する次のカスタマイズ可能相関処理が用意されています。

- Pairwise 相関処理

  CiscoLinkDown に続く CiscoLinkUp のように，論理的な関係があり，［**インシデント**］ビューに両方を表示させる必要がない場合に，関連するインシデントとしてまとめて管理します。具体的には，インタフェースが LinkDown から LinkUp したときに LinkDown/LinkUp のメッセージを抑止します。
- 重複削除相関処理

  指定した時間ウィンドウ内に複数のインシデントのコピーを受信すると，重複削除インシデントの重複が相関処理されます。新たに受信した各重複インシデントの時間ウィンドウが再開始されます。このように，NNMi では相関処理時間ウィンドウの全期間中，重複を受信しなくなるまで重複インシデントが相関処理されます。
- レート相関処理

  指定時間帯内にインシデントに関する指定コピー数を受信すると，レートインシデントの重複が相関処理されます。時間ウィンドウの残り時間にかかわらず，指定数のインシデントを受信すると NNMi によってレートインシデントが生成されます。

8.1.7 インシデントの抑制，強化，およびダンプニング

NNMi には，インシデントからほとんどの値を取得する便利な機能セットが用意されています。各インシデントタイプに対して，次のインシデント設定オプションでインシデントが関連する場合を具体的に指定できます。

- 抑制

  インシデントが抑制設定に一致すると，そのインシデントは NNMi コンソールインシデントビューに表示されません。インシデントの抑制は，あるノード（ルーター，スイッチなど）にとっては重要であるが，ほかにとっては重要ではないインシデント（SNMPLinkDown トラップなど）の場合に便利です。

- 強化

  インシデントが強化設定に一致すると，インシデントのコンテンツに応じて，NNMi によって 1 つ以上のインシデント値（重大度，メッセージなど）が変更されます。インシデントの強化は，トラップ varbind（ペイロード）に識別情報を継承するトラップ処理（RMONFallingAlarm など）の場合に便利です。

- ダンプニング

  インシデントがダンプニング設定に一致すると，ダンプニング周期中，NNMi によってインシデントビューの表示更新，アクション実行などが遅延されます。インシデントのダンプニングは，NNMi Causal Engine がインシデントの根本原因分析を実行する時間が必要なときに，NNMi コンソールのインシデント数を減らせるため，分析の精度を上げることができます。

NNMi には，各インシデントタイプに抑制，強化，ダンプニングに対する次の設定レベルが用意されています。

- インタフェースグループ設定

  ソースオブジェクトが NNMi インタフェースグループのメンバーである場合のインシデント動作が指定されます。各インタフェースグループに異なる動作を指定できます。

- ノードグループ設定

  ソースオブジェクトが NNMi ノードグループのメンバーである場合のインシデントの動作が指定されます。各ノードグループに異なる動作を指定できます。

- デフォルト設定

  デフォルトのインシデント動作が指定されます。

NNMi では，各インシデントの設定領域（抑制，強化，ダンプニング）に対して，次の手順を使用して特定のインシデントの動作が決定されます。

1. インタフェースグループ設定をチェックする。
   - ソースオブジェクトが任意のインタフェースグループ設定に一致する場合は，一致内で最も小さい順序番号で定義された動作を実行し，一致検索を停止します。
   - ソースオブジェクトがどのインタフェースグループ設定とも一致しない場合は，手順 2.を続行します。
2. ノード グループ設定をチェックする。

- ソースオブジェクトが任意のノードグループ設定に一致する場合は，一致内で最も小さい順序番号で定義された動作を実行し，一致検索を停止します。
- ソースオブジェクトがどのノードグループ設定とも一致しない場合は，手順 3.を続行します。

3. デフォルト設定で定義された動作を実行する（ある場合）。

8.1.8 ライフサイクルの移行アクション

ライフサイクル移行アクションは管理者が提供するコマンドであり，インシデントのライフサイクル状態が変化してアクション設定と一致したときに実行されます。インシデントのアクション設定は，各インシデントタイプのそれぞれのライフサイクル状態ごとに設定されます。このインシデントタイプが特定のライフサイクル状態に移行すると，アクション設定によって，実行するコマンドが特定されます。コマンドには引数を指定でき，引数でインシデント情報がアクションコードに渡されます。

アクションコードは，NNMi 管理サーバーで正しく実行されるJython ファイル，スクリプト，実行可能ファイルのどれかにできます。アクションコードは各インシデントタイプに固有のものにしたり，多くのインシデントタイプを処理するようにしたりできます。例えば，ConnectionDown，NodeDown，NodeOrConnectionDown のどれかのインシデントを NNMi が作成したときにネットワークオペレータを呼び出すアクションコードを作成できます。それぞれのインシデントタイプの［登録済み］ライフサイクル状態に 1 つのインシデントアクションというように，3 つのインシデントアクションを設定できます。

同じように，アクションコードを 1 つのライフサイクル状態の変化に固有にしたり，複数のライフサイクル状態の変化に対応させたりできます。例えば，NNMi がInterfaceDown インシデントを作成したときにトラブルチケットを生成し，InterfaceDown インシデントがキャンセルされたときにトラブルチケットを解決済みにするアクションコードを作成できます。［登録済み］状態に 1 つ，［解決済み］状態に 1 つというように，InterfaceDown インシデントに 2 つのインシデントアクションを設定できます。

それぞれのアクション設定には，CIA に基づいてペイロードフィルタを組み込んで，アクションが実行されるときを制限できます。さらにフィルタリングするには，インシデントの強化を使用して CIA をインシデントに追加できます。NNMi はインシデントソースからその属性の値を判別します。例えば，一部のノードにカスタム属性を追加した場合は，この情報をインシデントに CIA として追加し，インシデントアクションのペイロードフィルタをこの属性値に基づくようにできます。

8.2 インシデントの計画

次の領域で決定します。

- 処理する SNMP トラップ
- 表示するインシデント
- インシデントに対する NNMi の対応方法

8.2.1 処理する SNMP トラップを計画する

ネットワークに関連するデバイストラップを識別し，各トラップのインシデント設定を計画します。NNMi では，MIB を NNMi にロードしないでトラップを処理できます。

NNMi の`nnmincidentcfg.ovpl -loadTraps` スクリプトを使用すると，SNMP トラップのインシデント設定の作成や更新を，MIB ファイルを使用して自動化できます。MIB ファイルに TRAP-TYPE または NOTIFICATION-TYPE マクロが含まれる場合は，インシデント設定に必要な情報を取得できます。

NNMi トポロジにないデバイスからのトラップを表示するかどうかを決定します。

8.2.2 表示するインシデントを計画する

インシデントのデフォルトセットで開始することをお勧めします。インシデント設定は徐々に拡大および削減できます。

重複削除，レート設定，Pairwise 相関処理によって削減できるインシデントを計画します。

詳細については，NNMi ヘルプ「管理」を参照してください。

8.2.3 インシデントに対する NNMi の対応方法を計画する

インシデントが発生した場合に，どのような NNMi のアクション（例えば，ネットワークオペレータへの電子メール送信など）を実行するか，各アクションを実行するライフサイクルの状態を計画します。

詳細については，NNMi ヘルプ「管理」を参照してください。

8.3　インシデントの設定

インシデントの設定手順については，NNMi ヘルプの「インシデントを設定する」を参照してください。

> **メモ**
>
> 大きな設定変更を行う前には，既存の設定のコピーを保存しておくことをお勧めします。詳細については，「4.2　ベストプラクティス：既存の設定を保存する」を参照してください。

8.3.1　インシデントの抑制・強化・ダンプニングを設定する

インシデントの抑制，強化，ダンプニングを設定するときは，次のことに注意してください。

- 各インタフェースグループ，ノードグループ，またはデフォルト設定に対して設定を適用できる場合に，さらに絞り込むためのペイロードフィルタを指定できます。
- インシデント設定フォームの［**インタフェースの設定**］タブにインタフェースグループを設定します。
- インシデント設定フォームの［**ノードの設定**］タブにノードグループを設定します。
- インシデント設定フォームの［**抑制**］，［**強化**］，および［**ダンプニング**］タブにデフォルトを設定します。

8.3.2　ライフサイクル移行アクションを設定する

ライフサイクル移行アクションを設定するときは，次のことに注意してください。

- デフォルトでは，NNMi は次の場所でアクションを実行します。
  - Windows：`%NnmDataDir%shared¥nnm¥actions`
  - Linux：`$NnmDataDir/shared/nnm/actions`

  アクションがこの場所にない場合は，［**ライフサイクルの移行アクション**］フォームの［**コマンド**］フィールドでアクションの絶対パスを指定します。

  > **重要**
  >
  > Jython ファイルは`actions` ディレクトリに配置する必要があります。

- アクション設定を変更するたびに，NNMi によって`actions` ディレクトリで`Jython` ファイルが再読み取りされて NNMi にロードされます。
- アクションは，グループとしてインシデントタイプに対して有効になります。
- アクションに渡すことができる NNMi 情報については，NNMi ヘルプの「インシデントアクションを設定するための有効なパラメーター」を参照してください。

8.3.3 トラップログを設定する

NNMi では，すべての着信 SNMP トラップをログファイル（テキストファイルまたは CSV ファイル）に記録できます。トラップは次の場所に記録されます。

- Windows：%NnmDataDir%log¥nnm
- Linux：$NnmDataDir/log/nnm

トラップログファイルは，nnmtrapconfig.ovpl スクリプトを使用して設定します。次の形式を選択できます。

- CSV（デフォルト）：トラップは CSV 形式で記録されます（trap.csv）。
- LOG：トラップはテキスト形式で記録されます（trap.log）。
- BOTH：トラップは CSV とテキストの両方の形式で記録されます（2 つのログファイル）。
- OFF：トラップは記録されません。

例えば，BOTH モードでトラップを記録する場合は，次のコマンドを使用します。

```
nnmtrapconfig.ovpl -setProp trapLoggingMode BOTH -persist
```

-persist 引数を使用することで，トラップサービスの再起動後もすべてのトラップサーバープロパティがそのまま有効になります。-persist 引数を使用しない場合，すべてのトラップサーバープロパティはサービスが停止されるまでの間だけが有効です。

トラップはロールファイルに書き込まれます。ログファイルのサイズが定義された上限（nnmtrapconfig.ovpl スクリプトを使用して定義）に達すると，ファイル名がtrap.<format>.old に変更され，既存のファイルは置き換えられます。

詳細については，nnmtrapconfig.ovpl リファレンスページを参照してください。NNMi ヘルプの「トラップログ記録を設定する」もあわせて参照してください。

8.3.4 インシデントログを設定する

受信インシデント情報がincident.csv ファイルに書き込まれるように，インシデントログを設定できます。この機能は，インシデント履歴を追跡およびアーカイブする場合に役立ちます。

インシデントログを設定して有効にするには，［設定］ワークスペースの［インシデントの設定］エリアにある［インシデントログの設定］タブに移動して設定します。詳細については，NNMi ヘルプを参照してください。

8.3.5 トラップサーバープロパティを設定する

トラップサーバープロパティ（nnmtrapserver.properties）を設定するには，nnmtrapconfig.ovpl スクリプトを使用します。

nnmtrapserver.properties ファイルを直接編集しないでください。nnmtrapconfig.ovpl スクリプトを使用してこのファイルを変更してください。

トラップサーバープロパティには次のデフォルト値が設定されています。

表 8-4 トラップサーバープロパティとそのデフォルト値

トラップサーバープロパティ	デフォルト値
com.hp.ov.nms.trapd.udpPort	162
com.hp.ov.nms.trapd.trapInterface	すべてのインタフェース
com.hp.ov.nms.trapd.recvSocketBufSize	53,248 キロバイト
com.hp.ov.nms.trapd.pipeline.qSize	50,000 トラップ
com.hp.ov.nms.trapd.blocking	true
com.hp.ov.nms.trapd.blockTrapRate	50 トラップ/秒
com.hp.nms.trapd.unblockTrapRate	50 トラップ/秒
com.hp.ov.nms.trapd.overallBlockTrapRate	150 トラップ/秒
com.hp.nms.trapd.overallUnblockTrapRate	150 トラップ/秒
com.hp.ov.nms.trapd.analysis.minTrapCount	100 トラップ
com.hp.ov.nms.trapd.database.fileSize	100 メガバイト
com.hp.ov.nms.trapd.database.fileCount	5 ファイル
com.hp.ov.nms.trapd.database.qSize	300,000 トラップ
com.hp.ov.nms.trapd.discohint.cacheSize	5,000 エントリ
com.hp.ov.nms.trapd.discohint.cacheEntryTimeout	3,600 ミリ秒

詳細については，nnmtrapconfig.ovpl リファレンスページを参照してください。

8.4 インシデント設定のバッチロード

nnmincidentcfgdump.ovpl とnnmincidentcfgload.ovpl の 2 つのスクリプトをインシデント設定のバッチロードと併用できます。

8.4.1 nnmincidentcfgdump.ovpl でインシデント設定ファイルを生成する

NNMi では，nnmincidentcfgdump.ovpl スクリプトを使用して，インシデント設定を作成または更新し，その後nnmincidentcfgload.ovpl スクリプトを使用して NNMi データベースにロードできます。ファイルは非 XML 形式で生成されます。

次のディレクトリにある形式の説明を使用して，ファイルを編集できます。

- Windows：%NnmInstallDir%examples¥nnm¥incidentcfg
- Linux：/opt/OV/examples/nnm/incidentcfg

インシデント設定のファイルを生成するには，次の構文の例を使用します。

```
nnmincidentcfgdump.ovpl -dump <file_name> -uuid -u <NNMiadminUsername> -p <NNMiadminPassword
>
```

詳細については，nnmincidentcfgdump.ovpl リファレンスページを参照してください。

8.4.2 nnmincidentcfgload.ovpl でインシデント設定をロードする

NNMi では，nnmincidentcfgload.ovpl スクリプトを使用して，フォーマットされた設定ファイルから NNMi データベースにインシデント設定をロードできます。

必要な形式については，次のディレクトリを参照してください。

- Windows：%NnmInstallDir%examples¥nnm¥incidentcfg
- Linux：/opt/OV/examples/nnm/incidentcfg

インシデント設定ファイルを NNMi データベースにロードする前に検証するには，次の構文の例を使用します。

```
nnmincidentcfgload.ovpl -validate <file_name> -u <NNMiadminUsername> -p <NNMiadminPassword>
```

インシデント設定をロードするには，次の構文の例を使用します。

```
nnmincidentcfgload.ovpl -load <file_name> -u <NNMiadminUsername> -p <NNMiadminPassword>
```

次の点に注意してください。

- NNMi は，名前またはそのほかのキー識別子が一致するすべての設定を更新します。

  nnmincidentcfgdump.ovpl スクリプトを使用して，既存のインシデント設定の設定ファイルを非 XML 形式で作成します。その後必要に応じて，NNMi データベースにロードする前にこのファイルを編集できます。

  NNMi は，これらの設定に関連づけられたコード値（インシデントファミリなど）の上書きも行います。
- NNMi は，NNMi データベースにないキー識別子のすべてのインシデント設定を追加します。
- NNMi は，エクスポートされたファイル内で一致しないキー識別子の既存のインシデント設定は変更しません。
- NNMi は，設定ファイルで提供されていない場合は一意のオブジェクト ID（UUID）を解決します。
- NNMi が UUID を解決できない場合は，UUID が作成されます。

重要

高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，ovstop コマンドおよびovstart コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

詳細については，nnmincidentcfgload.ovpl リファレンスページを参照してください。

8.5 インシデントの評価

このセクションでは，インシデント設定を評価する方法を説明します。

- NNMi がネットワークのすべての管理対象デバイスからトラップを受信したことを確認します。

  NNMi がトラップを受信していない場合は，NNMi 管理サーバーでファイアウォールの設定を確認します。

  > **メモ**
  >
  > 一部のウイルス対策ソフトウェアにはファイアウォールが組み込まれ，システムのファイアウォールとは別に設定されています。

- 最も重要なトラップがインシデントに変換されることを確認します。
- 正しいライフサイクルの状態移行でインシデントアクションが実行されていることを確認します。
- NNMi がインシデントを期待どおり処理していることを確認します。

  ［アクション］ ＞ **［インシデントの設定レポート］** メニューには，既存のインシデントをそのインシデントタイプの現在の設定に対してテストする複数のオプションがあります。これらのメニュー項目のどれかを使用しても，現在 NNMi コンソールにあるインシデントは変更されません。

8.6 インシデントの調整

NNMi コンソールインシデントビューのインシデント数を削減します。次のメソッドのどれかを使用します。

- NNMi コンソールでは必要のないインシデントタイプのインシデント設定を無効にします。
- 監視する必要がないネットワークオブジェクトの管理モードを［非管理対象］または［サービス停止中］に設定します。NNMi では，これらのノードとそのインタフェースからのほとんどの受信トラップを廃棄します。
- NNMi でネットワークオブジェクトが監視されないように設定します。NNMi では，監視されないソースオブジェクトからのほとんどの受信トラップを廃棄します。
- 受信インシデントの追加条件または関係を識別します。これらの条件または関係が発生すると，NNMi では受信管理イベントや SNMP トラップの条件またはパターンを識別して，関連するインシデントどうしを相関関係の子として入れ子にすることで，インシデントのフローが変更されます。

8.6.1 未定義のトラップのインシデントを有効化にする

NNMi はデフォルトでインシデント定義のない SNMP トラップを破棄します。

インシデント定義のない SNMP トラップを「UndefinedSNMPTrap」インシデントとして生成するには，次の手順を実行します。

1. 次のファイルをテキストエディタで開く。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
2. 次の行を検索する。

   ```
   #!com.hp.nnm.events.allowUndefinedTraps=false
   ```

   次のように編集します。

   ```
   com.hp.nnm.events.allowUndefinedTraps=true
   ```

3. （任意）インシデントの重大度を指定する。

   次の行を検索します。

   ```
   #!com.hp.nnm.events.undefinedTrapsSeverity=NORMAL
   ```

   「YourSpecifiedSeverity」にインシデントの重大度を指定します。

   ```
   com.hp.nnm.events.undefinedTrapsSeverity=YourSpecifiedSeverity
   ```

   有効な値は，NORMAL, WARNING, MINOR, MAJOR, CRITICAL です。
4. （任意）インシデントの根本原因を指定する。

次の行を検索します。

```
#!com.hp.nnm.events.undefinedTrapsNature=INFO
```

「YourSpecifiedNature」にインシデントの根本原因を指定します。

```
com.hp.nnm.events.undefinedTrapsNature=YourSpecifiedNature
```

有効な値は，ROOTCAUSE，SECONDARYROOTCAUSE，SYMPTOM，SERVICEIMPACT，NONE，INFO です。

5. （任意）UndefinedSNMPTrap インシデントを複数回出すかどうかを指定する。

   各トラップ OID につき 1 度だけ UndefinedSNMPTrap インシデントを生成するか，トラップを受信するたびに毎回 UndefinedSNMPTrap インシデントを生成するかを選択できます。

   デフォルトでは，1 度だけ生成します。毎回生成するように変更する場合は，次の行を検索します。

```
#!com.hp.nnm.events.allowMultipleUndefinedTrapIncidents=false
```

   次のように編集します。

```
com.hp.nnm.events.allowMultipleUndefinedTrapIncidents=true
```

6. 変更を保存する。
7. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

8. 「UndefinedSNMPTrap」インシデントの一覧を見直す。

   インシデントとして表示したい SNMP トラップは，インシデント定義を設定する必要があります。詳細については，NNMi ヘルプを参照してください。

8.6.2 SNMP トラップの MIB データの文字列を正しく解釈し表示する

SNMP トラップの MIB データは，どのような文字セットで解釈すればよいか判断できません。

そのため，NNMi は SNMP トラップの MIB データ(sysDescription や sysContact など)を，文字化けして表示する場合があります。

正しく表示するためには，次の手順を実行し，NNMi が MIB データの文字列を解釈するときに使用する文字セットを設定します。

メモ

新規インストール，かつ，日本語環境の場合，デフォルトで下記が設定されています。

- UTF-8，EUC_JP，windows-31j，Shift_JIS

1. 次のファイルをテキストエディタで開く。

- Windows：%NNM_PROPS%¥nms-jboss.properties
- Linux：$NNM_PROPS/nms-jboss.properties

2. 次の行を検索し，コメントアウト(#!com.hp.nnm.sourceEncoding=)されている場合は，コメント記号(#!)を削除する。

```
#!com.hp.nnm.sourceEncoding=
```

3. com.hp.nnm.sourceEncoding プロパティを編集する。

   nms-jboss.properties ファイルの例を参考にして，com.hp.nnm.sourceEncoding プロパティに，使用される環境でサポートしている文字セットをコンマ（,）区切りで追加します。

4. 変更を保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

文字セットによる解釈をしないで，MIB データを 16 進形式で表示する場合は，次の手順を実行します。

1. 次のファイルをテキストエディタで開く。
   - Windows：%NnmDataDir%shared¥nnm¥conf¥nnmvbnosrcenc.conf
   - Linux：$NnmDataDir/shared/nnm/conf/nnmvbnosrcenc.conf
2. トラップ OID と VarBind OID の組み合わせを追加する。

   nnmvbnosrcenc.conf ファイルの例を参考にして，対象となる MIB データのトラップ OID と VarBind OID の組み合わせを追加します。

   NNMi はインシデントフォームのカスタム属性値で，指定した MIB データを 16 進形式で表示します。
3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

9 NNMi コンソール

この章では，NNMi コンソールを使用して NNMi の機能を設定する具体的な方法について説明します。

9.1 ノードグループの使用例

ここでは，実際的な例を示して，ノードグループの設定について説明します。

設定するノードグループ

My Network：ほかのノードグループを含んでいる最上位レベルのコンテナノードグループ

USA：ほかのノードグループを含んでいる中間レベルのコンテナノードグループ

Colorado：Colorado に存在するノードを含んでいるノードグループ

この例で，Colorado はノードが含まれている唯一のノードグループです。

ノードグループの設定で，次のことに注意してください。

- 事前にノードグループマップのレイアウトを設計するのが効果的です。
- ネットワーク監視のために，ノードグループとインタフェースグループのセットを 1 つ設定するのが効果的です。マップによって，ネットワーク可視化用に異なるノードグループのセットを設定します。
- NNMi では，幾つかの方法でノードグループとノードグループマップを設定できます。ここで説明する手順を理解することで，ノードグループやノードグループマップをより効率良く作成する方法を見つけることもできます。

ここでは，ノードグループとノードグループマップを設定する場合の手順について説明します。

ノードグループの作成

- 手順 1：My Network ノードグループを作成する。
- 手順 2：USA ノードグループを作成する。
- 手順 3：フィルタを使用してColorado ノードグループを作成する。
- 手順 4：ノードグループメンバーを表示してノードグループのフィルタ結果を確認する。
- 手順 5：My Network ノードグループのノードグループ階層を設定する。
- 手順 6：USA ノードグループのノードグループ階層を作成する。

親ノードグループには，ノードが含まれていない場合があります。その代わり，定義に子ノードグループだけが含まれています。この例では，My Network およびUSA ノードグループが，子ノードグループだけを含む親ノードグループです。

ノードグループマップの設定

- 手順 1：ノードグループマップを作成する。
- 手順 2：ノードグループマップを表示する。
- 手順 3：ノードグループのステータスを設定する。
- 手順 4：ノードグループマップの順序を設定する。
- 手順 5：ノードグループマップに背景イメージを追加する。

9.1.1　ノードグループを作成する

ノードグループを作成してノードグループマップに追加します。

(1)　手順 1：My Network ノードグループを作成する

次の手順で，My Network ノードグループを作成します。

1. [設定] ワークスペースに移動する。
2. [オブジェクトグループ] から [ノードグループ] を選択する。
3. [新規作成] アイコンをクリックする。
4. [名前] 属性に，「My Network」と入力する。
5. [注] 属性に，「最上位のノードグループです」と入力する。
6. [保存して閉じる] をクリックしてこの設定を保存する。

(2)　手順 2：USA ノードグループを作成する

1. [設定] ワークスペースに移動する。
2. [オブジェクトグループ] から [ノードグループ] を選択する。
3. [新規作成] アイコンをクリックする。
4. [名前] 属性に，「USA」と入力する。
5. [保存して閉じる] をクリックしてこの設定を保存する。

(3)　手順 3：フィルタを使用してColorado ノードグループを作成する

Colorado ノードグループを作成するには，フィルタエディタを使用してノードを選択するフィルタを設定します。

メモ

できれば，[追加のノード] タブを使用して一連のノードを指定するのではなく，[追加のフィルター] タブを使用してください。ノードグループフィルタを使用すると，NNMi では，新規ノードがネットワークに追加されるときに，ノードを正しいノードグループに自動的に配置できます。

1. [設定] ワークスペースに移動する。
2. [オブジェクトグループ] から [ノードグループ] を選択する。
3. [新規作成] アイコンをクリックする。
4. [名前] 属性に，「Colorado」と入力する。
5. [追加のフィルター] タブを選択する。

6. ノードが入力したホスト名値のどれかと一致する場合に NNMi がノードを照合するよう指定するには，［OR］をクリックする。
7. フィルタエディタの［属性］フィールドで，［hostname］を選択する。
   ［hostname］を選択すると，ノードがこのノードグループに属するかどうかを判断するときに，NNMi はホスト名値と照合します。
8. ［演算子］フィールドで，［like］を選択する。
   ［like］を選択すると，検索でワイルドカード文字を使用できます。
9. ［値］フィールドに，ノードグループに含めるデバイスを表す値を入力する。
   例えば，cisco*.ntc.example.com は，cisco<*値*>.<*network_domain*>という名前のデバイスを表します。
10. ［追加］をクリックする。
11. ［属性］フィールドで，［hostname］を選択する。
12. ［演算子］フィールドで，［like］を選択する。
13. ［値］フィールドに，Colorado ノードグループに追加する残りのデバイス名を表すワイルドカードを入力する。
    この例では，「cisco?*」を使用します。
14. ［追加］をクリックする。
15. ［保存］をクリックして，ウィンドウを閉じずにノードグループを保存する。

(4) 手順 4：ノードグループのフィルタ結果を確認する

ノードグループフィルタを確認するため，作成したノードグループのメンバーを表示できます。

［アクション］＞［ノードグループの詳細］＞［メンバーの表示］を選択して，ノードグループ内のすべてのノードを含んだビューを開きます。

メモ

ノードグループフィルタが正しく動作すると確信できるまで，ノードグループフィルタ定義の結果を調べてください。

(5) 手順 5：My Network ノードグループのノードグループ階層を設定する

My Network ノードグループを最上位レベルにして，ノードグループの階層を作成します。

1. ［設定］ワークスペースの［オブジェクトグループ］＞［ノードグループ］ビューに戻り，作成したノードグループの一覧を表示する。
2. My Network ノードグループに移動して，［開く］をクリックする。
3. ［子ノードグループ］タブをクリックする。
4. ［新規作成］アイコンをクリックする。

5. [子ノードグループ] 属性で，[検索] アイコンをクリックして [クイック検索] を選択する。

> **重要**
>
> [クイック検索] を使用して，ノードグループなどのオブジェクトがすでに存在する場合にはそれを選択します。

6. [USA] を子ノードグループとして選択する。
7. [OK] をクリックする。
8. [保存して閉じる] をクリックして変更を保存し，[ノードグループの階層] フォームを閉じる。
9. [保存して閉じる] をクリックして変更を保存し，[ノードグループ] フォームを閉じる。

(6) 手順 6：USA ノードグループのノードグループ階層を作成する

Colorado をUSA ノードグループの子ノードグループとして設定します。「(5) 手順 5：My Network ノードグループのノードグループ階層を設定する」の手順を繰り返して行い，Colorado ノードグループをUSA ノードグループの子に指定します。

これで，作成したノードグループごとにノードグループマップを作成する準備ができました。

9.1.2 ノードグループマップを設定する

作成したノードグループを使用してノードグループマップを設定するには，次の手順を実行します。

(1) 手順 1：ノードグループマップを作成する

各ノードグループのノードグループマップを作成するには，[アクション] メニューを使用します。

1. マップを作成するノードグループを開く。
   a [設定] ワークスペースの [オブジェクトグループ] ＞ [ノードグループ] オプションに戻り，作成したノードグループの一覧を表示します。
   b 対象のノードグループに移動し，[開く] アイコンをクリックします。
2. [アクション] ＞ [マップ] ＞ [ノードグループマップ] を選択して，ノードグループマップを表示する。
3. ノードおよびノードグループマップのアイコンの位置を決める。
4. [マップを保存] アイコンをクリックして，ノードマップアイコンを作成する。

> **メモ**
>
> ノードの位置を変更しない場合でも，ノードグループマップを作成するときには，いつでも [マップを保存] を使用してください。[マップを保存] によってノードグループマップが作成されます。

ノードグループマップが正常に作成されたことを知らせるダイアログボックスが表示されます。

5. [OK] をクリックする。
6. 作成した各ノードグループで，手順 1.～手順 5.までを繰り返す。

(2) 手順 2：ノードグループマップを表示する

ノードグループマップを表示するには，次の手順を実行します。

1. [トポロジマップ] ワークスペースに移動する。
2. [ノードグループの概要] を選択する。
3. 最上位レベルマップ [My Network] を選択する。
4. アイコンをダブルクリックして，子ノードグループのマップに移動する。
5. マップ上部の階層リンクを使用して前のマップに戻る。

(3) 手順 3：ノードグループのステータスを設定する

NNMi によって，ノードグループのステータスの計算方法を設定できます。ノードグループのステータスを設定するときには，次の中から NNMi で使用する方法を決めます。

- ノードグループ内で最も深刻なノードのステータスを使用する。
- NNMi で使用するパーセンテージの計算結果を指定する。

> **メモ**
>
> [ステータスの設定] はグローバル設定です。NNMi は，デフォルトでノードグループ内の最も深刻なノードのステータスを使用します。

1. [設定] ワークスペースに移動する。
2. [ステータスの設定] を選択する。
3. [ステータスの設定] フォームを調べ，デフォルトのパーセンテージを把握する。
   パーセンテージを使用するには，[最も重大なステータスを伝達] チェックボックスをオフにしてから，変更を保存する必要があります。

(4) 手順 4：ノードグループマップの順序を設定する

ノードグループマップの順序は，[トポロジマップ] ワークスペースに表示されるマップの順序を決めるのに役立ちます。この例では，ノードグループマップの順序を使用して，[トポロジマップ] ワークスペースのリストの最初にMy Network ノードグループマップが表示されるよう指定します。

1. [設定] ワークスペースに移動する。
2. [ユーザーインタフェース] から [ノードグループマップの設定] を選択する。

メモ

次の例では，デフォルトの［**トポロジマップ順序**］の値は，すべてのユーザー定義マップで50です。

My Networkを［**トポロジマップ**］ワークスペースの最初のマップとして一覧に表示するようNNMiに指示するには，［**トポロジマップ順序**］の値をほかのどのマップの［**トポロジマップ順序**］の値よりも小さい数字（例えば5）にします。

3. My Networkノードグループマップを開く。
4. ［**トポロジマップ順序**］属性で，値を5に変更する。
5. ［**保存して閉じる**］をクリックして変更を保存し，フォームを閉じる。

マップを最初にNNMiコンソールに表示するかどうかも指定できます。それには，［**設定**］ワークスペースで［**ユーザーインタフェースの設定**］オプションを使用します。

1. ［**設定**］ワークスペースに移動する。
2. ［**ユーザーインタフェース**］から［**ユーザーインタフェースの設定**］をクリックする。
3. ［**初期ビュー**］属性で，ドロップダウンメニューを使用して［**クイックアクセスマップフォルダの最初のノードグループ**］ワークスペースを選択する。

   これによって，My Networkマップが初期ビューに表示されます。

   初期ビューを確認するには，NNMiからサインアウトしてからもう一度サインインします。My Networkマップが NNMiコンソールに表示されるビューになります。

(5) 手順5：ノードグループマップに背景イメージを追加する

マップに背景グラフィックを含めるには，選択したノードグループマップで［**ノードグループマップの設定**］を使用します。

まず，事前準備として，背景イメージファイル（例：user.png）を以下に配置します。

- Windowsの場合：%NnmDataDir%shared¥nnm¥www¥htdocs¥images
- Linuxの場合：$NnmDataDir/shared/nnm/www/htdocs/images

1. ［**設定**］ワークスペースに移動する。
2. ［**ユーザーインタフェース**］をクリックする。
3. ［**ノードグループマップの設定**］をクリックする。
4. My Networkノードグループマップを開く。
5. ［**背景イメージ**］タブに移動する。
6. ［http://MACHINE:PORT/nnmdocs/images/］をクリックする。

   NNMiに，グラフィックの一覧が表示されます。

7. [user.png] を右クリックする。
8. リンクの場所をコピーする。
9. ディレクトリのリストウィンドウを閉じる。

> **メモ**
>
> コピーしたリンクの「/nnmdocs/images/ファイル名」の部分（/nnmdocs/images/user.png）を［背景イメージ］属性に貼り付けます。
>
> あとで変更する場合のために，［背景イメージのスケール］の値をメモします。

10. ［保存して閉じる］をクリックして変更を保存する。
11. ［トポロジマップ］ワークスペースの［クイックアクセスマップ］に移動し，［My Network］を選択して，新しいマップを背景グラフィックと一緒に表示する。

9.1.3 ノードグループを削除する

作成したColorado ノードグループを削除します。

1. ［設定］ワークスペースに移動する。
2. ［オブジェクトグループ］から［ノードグループ］をクリックする。
3. リストでColorado ノードグループを選択し，［開く］ボタンをクリックする。
   Colorado ノードグループに移動してColorado ノードグループの内容が表示されます。
4. ［ノードグループを削除］ボタンをクリックする。
   ダイアログボックスが表示されます。ノードグループを削除するとノードグループに含まれるすべてのオブジェクトと参照も削除されることが警告されます。
5. ［OK］をクリックしてノードグループを削除する。

9.2　ネットワークの概要マップに表示されるノードの最大数を削減する

[ネットワークの概要] マップには，レイヤー 3 ネットワークで最も高度に接続された 250 までのノードを含むマップが表示されます。このマップに含まれるノード数が多過ぎると，ノードを移動するときのマップの反応が遅くなったり，複雑過ぎて実際の表示に適さなくなったりするおそれがあります。**[ネットワークの概要]** マップに表示されるノードの最大数は次の例のように増減できます。

(例)：**[ネットワークの概要]** マップに表示されるノードの最大数を 250 から 100 に変更する。

次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-ui.properties`
   - Linux：`$NNM_PROPS/nms-ui.properties`
2. 次の行を探す。

   ```
   #!com.hp.nnm.ui.networkOverviewMaxNodes=250
   ```

   表示されるノードの最大値を次のように指定します。

   ```
   com.hp.nnm.ui.networkOverviewMaxNodes=100
   ```

   重要

   行の先頭の「#!」を忘れずに削除してください。

3. 変更を保存する。
4. 次のコマンドを実行して NNMi を再起動する。

   ```
   ovstop
   ovstart
   ```

9.3 ノードグループマップに表示されるノードの最大数を削減する

数百単位のノードを含むようにノードグループマップを設定すると，ノードグループを表示するマップには，予期される詳細なノードアイコンではなく，多くの小さいノードアイコンが表示されます。より詳細なマップを表示するには，ズーム機能を使用する必要があります。ズーム機能を使用すると，マップを表示するときの NNMi コンソールのパフォーマンスが低下するおそれがあります。

次の手順を実行して，表示されるノードまたは表示されるエンドポイント，またはその両方の数を制限してください。

1. NNMi コンソールで，［設定］をクリックする。
2. ［ユーザーインタフェース］の下にある［ユーザーインタフェースの設定］をクリックする。
3. ［デフォルトのマップ設定］タブを選択する。
4. ［表示するノードの最大数］フィールドに表示された値を変更する。
5. ［表示するエンドポイントの最大数］フィールドに表示された値を変更する。
6. ［保存して閉じる］をクリックする。

詳細は，NNMi ヘルプの「デフォルトマップ設定を定義する」を参照してください。

9.4 分析ペインのゲージの設定

分析ペインの［ゲージ］タブには， State Poller とカスタムポーラーの SNMP データを示すために，リアルタイムの SNMP ゲージが表示されます。これらのゲージには，ノード，インタフェース，カスタムノード収集のデータや，CPU ，メモリ，バッファー，バックプレーンタイプのノードコンポーネントのデータが表示されます。

次のプロパティファイルを編集してゲージを設定できます。

- Windows の場合：%NNM_PROPS%¥nms-ui.properties
- Linux の場合：$NNM_PROPS/nms-ui.properties

設定する各プロパティで，行の始めにコメント文字（#!）が存在する場合は削除します。

メモ

後続の項で説明するプロパティはすべてのノードに適用されます（個別のノードグループにプロパティを適用することはできません）。

ヒント

変更を行う前にnms-ui.properties ファイルのバックアップコピーを作成します。バックアップコピーは，編集するプロパティファイルが格納されているディレクトリに配置しないでください。

詳細については，nms-ui.properties ファイル内のコメントも参照してください。

9.4.1 分析ペインを無効にする

NNMi コンソールからアナリシス（分析）ペインを無効にするには，次の手順で実行します。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-ui.properties
   - Linux：$NNM_PROPS/nms-ui.properties
2. 次のプロパティが含まれる行を探す。

   ```
   #!com.hp.nnm.ui.analysisPaneDisabled = true
   ```

   次のように行の先頭の「#! 」を削除して，アナリシス（分析）ペインを無効にします。

   ```
   com.hp.nnm.ui.analysisPaneDisabled = true
   ```

3. 変更を保存する。

4. 次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

9.4.2　表示されるゲージ数の制限

次の行を編集して目的の値を入力し，表示するゲージの最大数を設定します。

```
com.hp.nnm.ui.maxGaugePerAnalysisPanel =
```

ヒント

ゲージ数が多いほど，分析ペインの表示時のパフォーマンスに影響します。ゲージ数が少ないほどゲージのサイズが大きくなります。

9.4.3　分析ペインにあるゲージの更新間隔の設定

次のプロパティ値を編集して，分析ペインに表示されるゲージの更新間隔（秒）を設定します。

```
com.hp.nnm.ui.analysisGaugeRefreshSecs =
```

ヒント

値を「0」に設定すると，ゲージが更新されなくなります。更新間隔を 10 秒より速くすると，一部の SNMP エージェントでは短時間で値がキャッシュされ，結果が同じになります。

9.4.4　ゲージの非表示

次の行を編集し，非表示にするゲージのリストを入力して，（すべてのゲージビューの）表示しないゲージを定義します。

```
com.hp.nnm.ui.analysisGaugeNoDisplayKeyPatterns =
```

次の点に注意してください。

- 関連するすべての行からコメント文字を削除してください。
- ゲージのリスト内にコメントを含めることはできません。
- ゲージのリスト内に空白行を含めないようにします。

空白行がある場所でエントリが終了します。

- コメント内の設定がこのプロパティのデフォルト設定です。

  この設定を拡張または修正する場合，これらの設定を含める必要があります。含めないと，予期しない数のゲージが表示されます。

9.4.5　表示されるノードゲージの順序の制御

ノードゲージが表示される順序を制御するには，次の行を編集します。

```
com.hp.nnm.ui.analysisGaugeNodeComponentKeys =
```

次の点に注意してください。

- このプロパティ設定では，ワイルドカードはサポートされていません。
- リストにコメントまたは空白行が含まれていないことを確認してください。
- このプロパティのデフォルト設定がコメントとして表示されます。この設定を拡張または修正する場合，これらの設定を含める必要があります。含めないと，設定した順序で表示されません。

9.4.6　表示されるインタフェースゲージの順序の制御

インタフェースゲージが表示される順序を制御するには，次の行を編集します。

```
com.hp.nnm.ui.analysisGaugeInterfaceKeys =
```

次の点に注意してください。

- このプロパティ設定では，ワイルドカードはサポートされていません。
- リストにコメントまたは空白行が含まれていないことを確認してください。
- コメント内の設定がこのプロパティのデフォルト設定です。この設定を拡張または修正する場合，これらの設定を含める必要があります。含めないと，意図した順序で表示されません。

9.4.7　表示されるカスタムポーラーゲージの順序の制御

カスタムポーラーゲージが表示される順序を制御するには，次の行を編集します。

```
com.hp.ov.nnm.ui.analysisGaugeCustomPolledInstanceKeys =
```

> **メモ**
>
> この属性にデフォルト設定はありません。

9.4.8　ゲージプロパティの適用方法の理解

ゲージプロパティは次の順序で適用されます。

1. すべてのゲージのリストが State Poller から取得されます。
2. analysisGaugeNoDisplayKeyPatterns が最初に適用されて，指定のゲージがリストから削除されます。
3. analysisGaugeNodeComponentKeys，analysisGaugeInterfaceKeys，または analysisGaugeCustomPolledInstanceKeys が必要に応じて適用され，表示されるゲージのリストの順序が決まります。
4. 最後に，maxGaugePerAnalysisPanel が適用されて，表示されるリストが切り捨てられます。

9.4.9　ゲージに関する問題のトラブルシューティング

このセクションでは，ゲージに関する次の問題のトラブルシューティングについて説明します。

- 「(1)　表示されるゲージが多すぎる」

(1)　表示されるゲージが多すぎる

ゲージが多すぎる場合は，次のどちらかを実行します。

- maxGaugePerAnalysisPanel プロパティを使用して，表示されるゲージ数を制限します。
  詳細については，「9.4.2　表示されるゲージ数の制限」を参照してください。
- analysisGaugeNoDisplayKeyPatterns プロパティを使用して，不要なゲージを削除します。
  詳細については，「9.4.4　ゲージの非表示」を参照してください。

9.5 マップラベルのスケールサイズと境界の設定

NNMi 管理者は，nms-ui.properties ファイルを使用してマップビューに次の調整を加えることができます。

- マップとしてのノードラベルおよびポートラベルのスケール値は，ズーム機能によってサイズ変更される。
- マップ上でのノードまたはポートとそれらのラベル間のサイズ差を決定するために使用できる最大相対スケール係数。
- ノードとポートのラベルが黒い枠で囲まれるかどうか。

> **メモ**
>
> デフォルトでは，ラベルが重なるときに読みやすいように，ノードとポートのラベルは黒い枠で囲まれます。

次の表に変更するプロパティを示します。

> **ヒント**
>
> 各スケール調整プロパティ値は，NNMi で使用される実際のスケール係数を掛けたものです。例えば，labelScaleAdjust 値を 0.50 に変更すると，マップ上に表示されるラベルはその通常のサイズの半分になります。

表 9-1　nms-ui.properties ファイルで変更するプロパティ

プロパティ	デフォルト値	説明
com.hp.nnm.ui.labelScaleAdjust	1.0	ノードとポートのマップラベルのスケールサイズを調整します。
com.hp.nnm.ui.omitLabelRectangle	false	ノードラベルとポートラベルを囲むために黒い枠を使用するかどうかを決定します。 **注：**枠を表示しない場合，この値をtrue に設定します。

> **メモ**
>
> 変更を適用するには，マップビューを開き直すか，または変更します。

9.6 Loom 図および Wheel 図の自動折りたたみしきい値の設定

NNMi 管理者は，Loom 図と Wheel 図が相当複雑になったときに読みやすくするために，これらの図が初期動作として自動的にノードの折りたたみ（インタフェースの非表示）とスイッチの折りたたみ（ポートの非表示）を行うポイントを設定できます。この設定は，nms-ui.properties ファイルの次のプロパティを調整して行います。

表 9-2 Loom および Wheel の自動折りたたみしきい値

プロパティ	説明
com.hp.nnm.ui.wheelAutoCollapseThreshold	このプロパティは，Wheel 図の自動的な折りたたみが開始されるまでに境界線の周囲に必要なラベル数を指定するために使用します。
com.hp.nnm.ui.loomAutoCollapseThreshold	このプロパティは，Loom 図の自動的な折りたたみが開始されるまでに図全体で必要なラベル数を指定するために使用します。

自動折りたたみしきい値を設定するには，次の手順を実行します。

1. 次のファイルを編集します。
   - Windows の場合：%NNM_PROPS¥nms-ui.properties
   - Linux の場合：$NNM_PROPS/nms-ui.properties
2. 必要に応じて，必要なプロパティをコメント解除します。詳細については， nms-ui.properties ファイル内のコメントを参照してください。
3. 必要に応じてしきい値を更新し，変更を保存します。
4. 変更を適用するには，NNMi コンソールで図を開き直します。

9.7　デバイスのプロファイルアイコンをカスタマイズする

NNMi では，デバイスのプロファイルまたは特定のノードに関連づけられているアイコンをカスタマイズできます。これらのアイコンはテーブルビューやメニュー項目に表示されます。また，NNMi トポロジマップの前景イメージとしても表示されます。

［設定］ワークスペースの**［ユーザーインタフェース］**フォルダにある**［アイコン］**オプションからアイコンを変更できます。

また，コマンドラインを使ってアイコンを変更または削除するには，`nnmicons.ovpl` コマンドを使用してください。詳細については，`nnmicons.ovpl` リファレンスページ，または NNMi ヘルプを参照してください。

9.8 テーブルビューのリフレッシュレートをオーバーライドする

NNMi では，NNMi 管理者が NNMi コンソールにあるテーブルビューのデフォルトのリフレッシュレートをオーバーライドできます。

推奨される最小リフレッシュレートは，30 秒です。リフレッシュレートを 30 秒未満に設定すると，パフォーマンスが低下することがあります。

NNMi テーブルビューのデフォルトのリフレッシュレートをオーバーライドするには，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-ui.properties`
   - Linux：`$NNM_PROPS/nms-ui.properties`
2. リフレッシュレートを変更するビューの URL 内の`viewInfoId` パラメーターを特定する。
   a リフレッシュレートを変更するビューを開きます。
   b ［新しいウィンドウでビューを表示］をクリックします。
   c URL 内の`viewInfoId` パラメーターをメモします。
   （例）
   ```
   viewInfoId=allIncidentsTableView
   ```
3. 次の形式を使用して，ビューとそのリフレッシュレートを秒数で指定する行を`nms-ui.properties` に追加する。
   ```
   com.hp.ov.nms.ui.refreshViewSecs.VIEWKEYWORD = SECS
   ```

   重要
   - VIEWKEYWORD は，ビューの URL 内の`viewInfoId` パラメーターです。
   - SECS は，リフレッシュレート（秒数）です。
   - コマンドラインの末尾に余分なスペースがないことを確認してください。

   例えば，［すべてのインシデント］ビューのリフレッシュレートを 120 秒に変更するには，`nms-ui.properties` に下記の行を追加します。

   ```
   com.hp.ov.nms.ui.refreshViewSecs.allIncidentsTableView = 120
   ```
4. 変更を保存する。
5. 新しいリフレッシュレートを確認するには，別のビューを開いてから，リフレッシュレートを変更したビューに戻る。

10 NNMi での証明書の使用

証明書は，Web サーバーの識別情報をブラウザに示すものです。この証明書には，自己署名するか CA（認証機関）による署名を付けることができます。nnm-key.p12 ファイルでは，プライベートキーと証明書は対応するパブリックキーとともに格納されます。nnm-trust.p12 ファイルには，通信する他者の証明書，または他者を識別するときに信頼する認証機関の証明書が保存されています。NNMi は，nnm-key.p12 ファイルとnnm-trust.p12 ファイルの両方に自己署名証明書を含めます。特定の NNMi 機能を使用するため，NNMi 管理サーバーはそれぞれの証明書を相互に共有する必要があります。この章では，NNMi 管理サーバー間でこれらの証明書をコピーする方法と，nnmcertmerge.ovpl スクリプトを使用してnnm-key.p12 ファイルおよびnnm-trust.p12 ファイルに証明書をマージする方法について説明します。

10.1　NNMi 証明書について

注意

NNMi 11-50 以降のバージョンでは，証明書を保存するために Public Key Cryptography Standards (PKCS) #12 リポジトリが導入されています。NNMi 11-50 以降の新しいインスタンスをシステムにインストールすると，新しい PKCS #12 ファイルベースの証明書管理方法を利用できます。以前のバージョンの NNMi からアップグレードした環境では，引き続き JKS リポジトリが証明書の格納に使用されます。

アップグレードした環境で，PKCS #12 リポジトリに移行するには，「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従います。

証明書を使用する上で参考となる用語について説明します。

表 10-1　証明書関連の用語

コンセプト	説明
キーストアーとトラストストアー	**トラストストアー：**NNMi トラストストアーは，NNMi が信頼するソースから取得した公開キーを格納するファイルです。 新たに NNMi 11-50 以降のバージョンをインストールした環境では，トラストストアーファイルの名前は`nnm-trust.p12` です。 **注意** NNMi が旧バージョンからバージョン 11-50 以降にアップグレードされた管理サーバーでは，トラストストアーファイルの名前は`nnm.truststore` です。ただし，追加の手順（「10.2 アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」に記載）を実行して`nnm.truststore` ファイルを`nnm-trust.p12` ファイルに移行できます。 **キーストアー：**NNMi キーストアーは，NNMi サーバーのプライベートキーをインポートするファイルです。 新たに NNMi 11-50 以降のバージョンをインストールした環境では，キーストアーファイルの名前は`nnm-key.p12` です。 **注意** NNMi が旧バージョンからバージョン 11-50 以降にアップグレードされた管理サーバーでは，キーストアーファイルの名前は`nnm.keystore` です。ただし，追加の手順（「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」に記載）を実行して`nnm.keystore` ファイルを`nnm-key.p12` ファイルに移行できます。 これらのファイルは，次の場所に格納されています。 • Windows の場合：`%NNM_DATA%¥shared¥nnm¥certificates¥` • Linux の場合：`$NNM_DATA/shared/nnm/certificates/`

コンセプト	説明
デフォルトの NNMi 証明書	NNMi は，デフォルトのプロパティを使用して生成される自己署名証明書とともにインストールされます。このデフォルトの証明書は，別の自己署名証明書または CA 署名の証明書に置き換えることができます。
ツール	（Java の Keytool ユーティリティを使用する）nnmkeytool.ovpl ユーティリティを使用して証明書を生成および管理します。 NNMi には，証明書をマージして NNMi システムでの信頼性を確立する nnmmergecert.ovpl ユーティリティも付属しています。このプログラムは，高可用性，フェイルオーバー，およびグローバルネットワーク環境のセットアップで使用します。
サポートされる暗号化アルゴリズム	NNMi は， RSA アルゴリズムを使用して生成された証明書を受け入れます。 DSA アルゴリズムはサポートされません。
自己署名証明書	自己署名証明書は，一般にサーバーと既知のクライアントグループ間にセキュア通信を確立するために使用します。 NNMi は，デフォルトのプロパティを使用して生成される自己署名証明書とともにインストールされます。 注：自己署名証明書を使用するように設定されている NNMi インスタンスは，ユーザーが Web ブラウザーで NNMi Web コンソールへのアクセスを試みると警告メッセージを表示します。
CA 署名証明書	証明書署名要求に対する応答として受け取る署名付きサーバー証明書には， CA 署名付きの NNMi 証明書と，1 つ以上の CA 証明書が含まれます（1 つ以上の CA 証明書が存在する場合は証明書チェーンとも呼びます）。 注：これらの証明書は 1 つのファイルに入っていることもあれば，2 つの別々のファイルに入っていることもあります。
ルート署名証明書	サーバーおよびユーザーの証明書の署名について信頼できる認証機関を示します。
中間 CA 証明書	サーバーまたはユーザーではなく，ルート CA または中間 CA（それ自体が署名機関）のどちらかで署名される証明書です。 注：中間 CA 証明書を含め，NNMi サーバー証明書からルート CA 証明書までの証明書のリストは，証明書チェーンと呼ばれます。

10.2 アップグレードされたNNMi環境で新しいキーストアーを使用するための設定

NNMi 11-50より前のバージョンでは，NNMiは証明書を保存するためにJava KeyStore (JKS) リポジトリを提供していました。NNMi 11-50以降のバージョンでは，証明書を保存するためにPublic Key Cryptography Standards (PKCS) #12リポジトリが導入されています。NNMi 11-50以降の新しいインスタンスをシステムにインストールすると，新しいPKCS #12ファイルベースの証明書管理方法を利用できます。

ただし，古いバージョンのNNMiをバージョン11-50以降にアップグレードした場合，PKCS #12ファイルベースの証明書管理はすぐには利用できず，NNMiでは証明書管理にJKSリポジトリが引き続き使用されます。

追加的な設定作業により，アップグレードされたNNMi管理サーバーを設定して，PKCS #12ファイルベースの証明書管理の新しい方法が使用されるようにすることができます。

アップグレードされたNNMi管理サーバーを設定して，PKCS #12ファイルベースの証明書管理が使用されるようにするには，次の手順を実行します。

1. ルートまたは管理者としてNNMi管理サーバーにログオンします。
2. 次のコマンドを実行して，新しいキーストアーファイルに移行します。

   Windowsの場合：

   ```
   %NnmInstallDir%bin¥nnmkeytool.ovpl -importkeystore -srckeystore
   %NnmDataDir%shared¥nnm¥certificates¥nnm.keystore -destkeystore
   %NnmDataDir%shared¥nnm¥certificates¥nnm-key.p12 -srcstoretype JKS -deststoretype PKCS12 -
   srcprovidername SUN -destprovidername SunJSSE -alias <src_alias>
   ```

   Linuxの場合：

   ```
   /opt/OV/bin/nnmkeytool.ovpl -importkeystore -srckeystore
   /var/opt/OV/shared/nnm/certificates/nnm.keystore -destkeystore
   /var/opt/OV/shared/nnm/certificates/nnm-key.p12 -srcstoretype JKS -deststoretype PKCS12 -
   srcprovidername SUN -destprovidername SunJSSE -alias <src_alias>
   ```

   注意

   コマンド実行後，「出力先キーストアのパスワードを入力してください」，「新規パスワードを再入力してください」，および「ソース・キーストアのパスワードを入力してください」と3回パスワードの入力を求められますので，すべてにnnmkeypassと入力してください。

   この新しい証明書管理の方法では，キーストアーに同時に複数の証明書を保持することはできません。このインスタンスで，<src_alias>は，移行したい以前のキーストアーファイルに含まれている証明書のエイリアスです。

   以前のキーストアーファイルに含まれている証明書エイリアスについては，次のファイルに設定されているcom.hp.ov.nms.ssl.KEY_ALIASの設定値を指定してください。

- Windows の場合：%NNM_CONF%¥nnm¥props¥nms-local.properties
- Linux の場合：$NNM_CONF/nnm/props/nms-local.properties

> **メモ**
>
> アプリケーションフェイルオーバー構成の NNMi 管理サーバーの場合は，<src_alias>にコマンドを実行するサーバーの証明書のエイリアスを指定してください。

3. 次のコマンドを実行して，新しいトラストストアーファイルに移行します。

Windows の場合：

```
%NnmInstallDir%bin¥nnmkeytool.ovpl -importkeystore -srckeystore
%NnmDataDir%shared¥nnm¥certificates¥nnm.truststore -destkeystore
%NnmDataDir%shared¥nnm¥certificates¥nnm-trust.p12 -srcstoretype JKS -deststoretype PKCS12
 -srcprovidername SUN -destprovidername SunJSSE
```

Linux の場合：

```
/opt/OV/bin/nnmkeytool.ovpl -importkeystore -srckeystore
/var/opt/OV/shared/nnm/certificates/nnm.truststore -destkeystore
/var/opt/OV/shared/nnm/certificates/nnm-trust.p12 -srcstoretype JKS -deststoretype PKCS12
 -srcprovidername SUN -destprovidername SunJSSE
```

> **注意**
>
> コマンド実行後，「出力先キーストアのパスワードを入力してください」，「新規パスワードを再入力してください」，および「ソース・キーストアのパスワードを入力してください」と 3 回パスワードの入力を求められますので，すべてにovpass と入力してください。

4. 次の場所にあるserver.properties ファイルをテキストエディタで開きます。
   - Windows の場合：%NnmDataDir%nmsas¥nms
   - Linux の場合：/var/opt/OV/nmsas/nms
5. ファイルの現在の内容を削除します。
6. ファイルに次の内容を追加します。

```
nmsas.server.security.keystore.type=PKCS12
nmsas.server.security.keystore.file=${com.hp.ov.DataDir}/shared/nnm/certificates/nnm-key.
p12
nmsas.server.security.keystore.cred=nnmkeypass
nmsas.server.security.truststore.file=${com.hp.ov.DataDir}/shared/nnm/certificates/nnm-tr
ust.p12
nmsas.server.security.truststore.cred=ovpass
nmsas.server.security.keystore.alias=
nms.comm.soap.https.PROTOCOLS=TLSv1.2
```

7. ファイルを保存します。
8. 次の場所にあるnms-local.properties ファイルをテキストエディタで開きます。
   - Windows の場合：%NnmDataDir%conf¥nnm¥props

- Linux の場合：/var/opt/OV/conf/nnm/props

9. すべての javax パラメーターの値を変更します。

パラメーター	値
javax.net.ssl.trustStore	${NnmDataDir}/shared/nnm/certificates/nnm-trust.p12
javax.net.ssl.trustStoreType	PKCS12
javax.net.ssl.keyStore	${NnmDataDir}/shared/nnm/certificates/nnm-key.p12
javax.net.ssl.keyStoreType	PKCS12

10. ファイルを保存します。
11. nnm.keystore ファイルおよびnnm.truststore ファイルを次のディレクトリから削除します。
    - Windows の場合：%NnmDataDir%shared¥nnm¥certificates
    - Linux の場合：/var/opt/OV/shared/nnm/certificates
12. NNMi を再起動します。

10.3　PKCS #12 リポジトリを使った証明書の使用

NNMi 11-50 より前のバージョンでは，NNMi は証明書を保存するために Java KeyStore (JKS) リポジトリを提供していました。NNMi 11-50 以降のバージョンでは，証明書を保存するために Public Key Cryptography Standards (PKCS) #12 リポジトリが導入されています。NNMi 11-50 以降の新しいインスタンスをシステムにインストールすると，新しい PKCS #12 ファイルベースの証明書管理方法を利用できます。

ただし，古いバージョンの NNMi をバージョン 11-50 以降にアップグレードした場合，PKCS #12 ファイルベースの証明書管理はすぐには利用できず，NNMi では証明書管理に JKS リポジトリが引き続き使用されます。

このセクションでは，新しくインストールした NNMi（または証明書リポジトリが PKCS #12 形式に移行された環境）で証明書を操作する手順を説明します。

メモ

NNMi 13-00 以降のバージョンを新規インストールした場合，NNMi の自己署名証明書に，Subject Alternative Name (SAN) が付与されます。

NNMi 12-60 以前のバージョンからアップグレードした環境では，NNMi の自己署名証明書に SAN が付与されません。NNMi の自己署名証明書を使用して NNMi に SSL 通信している環境で，SAN が付与された自己署名証明書への入れ替えが必要な場合は，この製品のリリースノートを参照してください。

また，Microsoft Edge や Google Chrome などの一部のブラウザで，証明書の認証に SAN を使用する場合があります。CA 署名証明書を使用する場合は，必要に応じて CA 署名機関に SAN が付加された証明書を発行するように依頼してください。

10.3.1　自己署名証明書の生成

注意

NNMi 11-50 以降のバージョンでは，証明書を保存するために Public Key Cryptography Standards (PKCS) #12 リポジトリが導入されています。NNMi 11-50 以降の新しいインスタンスをシステムにインストールすると，新しい PKCS #12 ファイル ベースの証明書管理方法を利用できます。以前のバージョンの NNMi からアップグレードした環境では，引き続き JKS リポジトリが証明書の格納に使用されます。

アップグレードした環境で，PKCS #12 リポジトリに移行するには，「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従います。

自己署名証明書を生成するには，次の手順を実行します。

1. nnm-key.p12 ファイルおよびnnm-trust.p12 ファイルが存在する NNMi 管理サーバーのディレクトリに変更します。
   - Windows の場合：%NnmDataDir%shared¥nnm¥certificates
   - Linux の場合：$NnmDataDir/shared/nnm/certificates
2. nnm-key.p12 ファイルのバックアップコピーを保存します。
3. 既存のnnm-key.p12 ファイルを削除します。
4. システムからプライベートキーを生成します。

   このプライベートキーを生成するには，nnmkeytool.ovpl コマンドを使用します。

   a. 次のコマンドをそのまま実行します。

      - Windows の場合：

```
%NnmInstallDir%bin¥nnmkeytool.ovpl -genkeypair -validity 36500 -keyalg rsa -keystor
e nnm-key.p12 -storetype PKCS12 -storepass nnmkeypass -alias <alias_name> -ext SAN=
DNS:<FQDN>
```

      - Linux の場合：

```
$NnmInstallDir/bin/nnmkeytool.ovpl -genkeypair -validity 36500 -keyalg rsa -keystor
e nnm-key.p12 -storetype PKCS12 -storepass nnmkeypass -alias <alias_name> -ext SAN=
DNS:<FQDN>
```

メモ

エイリアス（この例では<alias_name>）は，この新規作成キーを識別する名前です。エイリアスには任意の文字列を使用できますが，正しいバージョンを簡単に識別できるように完全修飾ドメイン名（FQDN）に続けてサフィックスを指定することをお勧めします。例えば，myserver.mydomain-<number>やmyserver.mydomain-<date>のようなエイリアス名を使用できます。

また，<FQDN>には，システムの FQDN（ブラウザで NNMi コンソールにアクセスする際に使用する FQDN）を入力してください。

   b. 必要な情報を入力します。

メモ

姓名の入力を求められたら，システムの FQDN（ブラウザで NNMi コンソールにアクセスする際に使用する FQDN）を入力してください。

   自己署名証明書が生成されます。

   CA 署名証明書を取得するためには，さらに CSR ファイルを生成し， CA に送信する必要があります。詳細については，「10.3.2　CA 署名証明書の生成」を参照してください。

10.3.2　CA 署名証明書の生成

注意

NNMi 11-50 以降のバージョンでは，証明書を保存するために Public Key Cryptography Standards (PKCS) #12 リポジトリが導入されています。NNMi 11-50 以降の新しいインスタンスをシステムにインストールすると，新しい PKCS #12 ファイルベースの証明書管理方法を利用できます。以前のバージョンの NNMi からアップグレードした環境では，引き続き JKS リポジトリが証明書の格納に使用されます。

アップグレードした環境で，PKCS #12 リポジトリに移行するには，「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従います。

CA 署名証明書を取得してインストールするには，次の手順を実行します。

1. 自己署名証明書を生成します。詳細については，「10.3.1　自己署名証明書の生成」を参照してください。
2. 次のコマンドを実行して，CSR（証明書署名要求）ファイルを作成します。
   - Windows の場合：

     ```
     %NnmInstallDir%bin¥nnmkeytool.ovpl -keystore nnm-key.p12 -certreq -storetype PKCS12 -s
     torepass nnmkeypass -alias <alias_name> -ext SAN=DNS:<FQDN> -file CERTREQFILE
     ```

   - Linux の場合：

     ```
     $NnmInstallDir/bin/nnmkeytool.ovpl -keystore nnm-key.p12 -certreq -storetype PKCS12 -s
     torepass nnmkeypass -alias <alias_name> -ext SAN=DNS:<FQDN> -file CERTREQFILE
     ```

   メモ

   - 上記のコマンドでは，<alias_name>，<FQDN>は証明書の生成時に指定したエイリアス，およびシステムの FQDN です。
   - CSR ファイルの内容を確認するには，次のコマンドを実行します。
     - Windows の場合：

       ```
       %NnmInstallDir%bin¥nnmkeytool.ovpl -printcertreq -file CERTREQFILE -store
       type PKCS12
       ```

     - Linux の場合：

       ```
       $NnmInstallDir/bin/nnmkeytool.ovpl -printcertreq -file CERTREQFILE -store
       type PKCS12
       ```

3. CA 署名機関に CSR を送信します（CA 署名機関が証明書ファイルに署名して返します）。各種の CA 証明書についての詳細は，「(1)　CA 署名証明書のタイプ」を参照してください。

注意

Microsoft Edge や Google Chrome などのブラウザで JP1/IM3 のインテリジェント統合管理基盤のカスタム UI 連携を使用する，かつ，コンソール画面への通信に SSL を使用する場合，証明書の Subject Alternative Name (SAN) にシステムの FQDN を設定し，その FQDN を使用してブラウザにアクセスする必要があります。

そのため，JP1/IM3 のインテリジェント統合管理基盤のカスタム UI 連携を使用する場合は，CA 署名機関へ CSR を送信する際に SAN が付加された証明書を発行するように依頼し，必ず証明書に SAN を付加するようにしてください。

JP1/IM3 のインテリジェント統合管理基盤については，「26.1　JP1/IM3 のインテリジェント統合管理基盤との連携」を参照してください。

CA 署名機関から，次のどちらかが返されます。

- 単一の署名付きサーバー証明書ファイル（このセクションでは`myserver.crt` ファイル）。サーバー証明書（CA 署名 NNMi 証明書），1 つ以上の中間 CA 証明書，およびルート CA 証明書を含む単一のファイル。この単一のファイル内のすべての証明書が証明書チェーンを形成します。
- 次の一対のファイル。一方が署名付きサーバー証明書ファイル（このセクションでは`myserver.crt` ファイル）で，もう一方（`myca.crt` ファイル）に CA 証明書が含まれています。`myserver.crt` ファイルには，1 つのサーバー証明書または証明書チェーンが含まれていますが，`myca.crt` ファイル内にあるルート CA 証明書は含まれていません。

メモ

CA から返される証明書のフォームがこれと異なる場合は，証明書チェーンおよびルート CA 証明書を取得する方法の詳細を CA 提供者に問い合わせてください。なお，サポートしている証明書の形式は PEM (Privacy Enhanced Mail) 形式のみです。PEM 形式の証明書を入手してください。

4. 証明書ファイルを用意します。

   証明書チェーンをキーストアーファイルにインポートし，ルート CA 証明書をトラストストアーファイルにインポートしてください。

   - 手順 3.で単一のファイルを受け取った場合

     そのファイル内のすべてのルート CA 証明書を別の`myca.crt` ファイルにコピーします。

   - 手順 3.で一対のファイルを受け取った場合

     `myca.crt`（ルート CA 証明書）ファイルの内容を`myserver.crt` ファイルの末尾に追加します。また，不要な中間証明書があればそれらを`myca.crt` ファイルからすべて削除します。これにより，完全な証明書チェーンを含んでいる 1 つのファイル`myserver.crt` ファイルと，ルート CA 証明書を含んでいる 1 つのファイル`myca.crt` ファイルが生成されます。

5. これらの証明書が記録されているファイルを NNMi 管理サーバーの任意の場所にコピーします。

この例では，次の場所にファイルをコピーします。

- Windows の場合：%NnmDataDir%shared¥nnm¥certificates
- Linux の場合：$NnmDataDir/shared/nnm/certificates

6. キーストアーおよびトラストストアーファイルの存在する NNMi 管理サーバー上のディレクトリに移動します。
   - Windows の場合：%NnmDataDir%shared¥nnm¥certificates
   - Linux の場合：$NnmDataDir/shared/nnm/certificates
7. 次のコマンドを実行して，証明書をキーストアーファイルにインポートします。
   - Windows の場合：

```
%NnmInstallDir%bin¥nnmkeytool.ovpl -importcert -trustcacerts -keystore nnm-key.p12 -st
oretype PKCS12 -storepass nnmkeypass -alias <alias_name> -file <path_to_myserver.crt>
```

   - Linux の場合：

```
$NnmInstallDir/bin/nnmkeytool.ovpl -importcert -trustcacerts -keystore nnm-key.p12 -st
oretype PKCS12 -storepass nnmkeypass -alias <alias_name> -file <path_to_myserver.crt>
```

メモ

上記のコマンドで，

- <path_to_myserver.crt>は，CA 署名サーバー証明書を保存した場所への絶対パスです。
- <alias_name>は，証明書の生成時に指定したエイリアスです。

8. 証明書の信頼を確認するメッセージが表示されたら，y を入力します。

証明書をキーストアにインポートするときの出力例

このコマンドによる出力形式は次のとおりです。

```
Owner:CN=NNMi_server.example.com
Issuer:CN=NNMi_server.example.com
Serial number:494440748e5
Valid from:Tue Oct 28 10:16:21 MST 2008 until:Thu Oct 04 11:16:21 MDT 2108
Certificate fingerprints:
MD5:29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
SHA1:C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03
Trust this certificate?[no]:y
Certificate was added to keystore
```

9. 次のコマンドを実行して，ルート証明書をトラストストアーファイルにインポートします。
   - Windows の場合：

```
%NnmInstallDir%bin¥nnmkeytool.ovpl -import -alias <alias_name> -storetype PKCS12 -keys
tore nnm-trust.p12 -file <path_to_myca.crt> -storepass ovpass
```

   - Linux の場合：

```
$NnmInstallDir/bin/nnmkeytool.ovpl -import -alias <alias_name> -storetype PKCS12 -keys
tore nnm-trust.p12 -file <path_to_myca.crt> -storepass ovpass
```

メモ

上記のコマンドで，

- <path_to_myca.crt>は，ルート証明書を保存した場所の絶対パスです。
- <alias_name>は，証明書の生成時に指定したエイリアスです。

10. トラストストアーの内容を確認します。

- Windows の場合：

```
%NnmInstallDir%bin¥nnmkeytool.ovpl -list -keystore nnm-trust.p12 -storetype PKCS12 -st
orepass ovpass
```

- Linux の場合：

```
$NnmInstallDir/bin/nnmkeytool.ovpl -list -keystore nnm-trust.p12 -storetype PKCS12 -st
orepass ovpass
```

トラストストアーの出力例

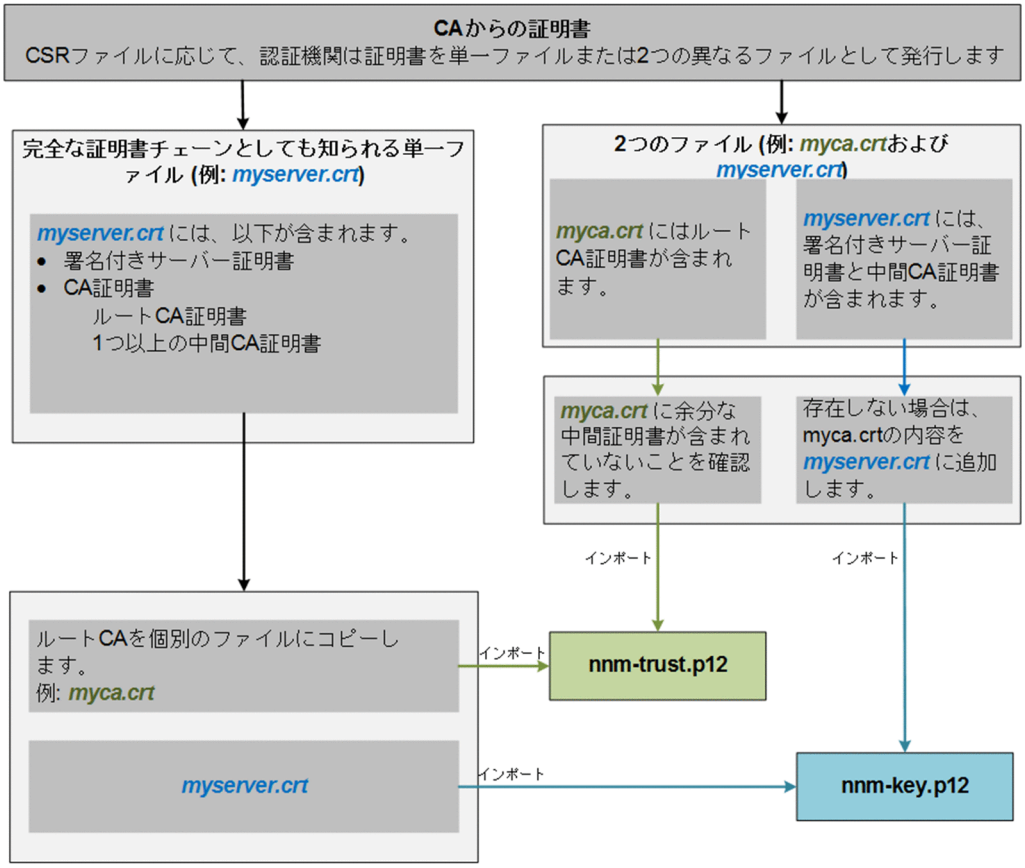
トラストストアーの出力形式は次のとおりです。

```
Keystore type: PKCS12
Keystore provider:BCFIPS
Your keystore contains 1 entry
nnmi_ldap, Nov 14, 2008, trustedCertEntry,
Certificate fingerprint (MD5):29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
```

ヒント

トラストストアーには複数の証明書を含めることができます。

(1) CA 署名証明書のタイプ



メモ

CA によって証明書が別のフォームで返される場合は，証明書チェーンとルート CA 証明書を取得する方法について，CA 提供者に問い合わせてください。

認証機関（CA）からは次のどちらかが提供されます。

- サーバー証明書（CA によって署名された NNMi 証明書）と 1 つ以上の CA 証明書が含まれる署名付きサーバー証明書ファイル。このセクションでは，署名付きサーバー証明書を`myserver.crt` ファイルとして示しています。

  CA 証明書には，次のどちらかを指定できます。

  - ルート CA 証明書：サーバーおよびクライアントの証明書の署名について信頼できる機関を示します。
  - 中間 CA 証明書：サーバーまたはユーザーではなく，ルート CA または中間 CA（それ自体が署名機関）のどちらかで署名される証明書。

メモ

中間 CA 証明書を含め， NNMi サーバー証明書からルート CA 証明書までの証明書のリストは，証明書チェーンと呼ばれます。

- 署名付きサーバー証明書と，1 つ以上の CA 証明書が含まれる別のファイル。このセクションでは，署名付きサーバー証明書をmyserver.crt ファイル，CA 証明書をmyca.crt ファイルとして示しています。myserver.crt ファイルは，1 つのサーバー証明書または証明書チェーンを含んでいる必要がありますが，myca.crt ファイル内にあるルート CA 証明書を含んでいる必要はありません。

NNMi に新しい証明書を設定するには，証明書チェーンをnnm-key.p12 ファイルにインポートし，ルート CA 証明書をnnm-trust.p12 ファイルにインポートする必要があります。サーバー証明書をnnm-key.p12 ファイルにインポートする場合はmyserver.crt ファイルを使用し，CA 証明書をnnm-trust.p12 ファイルにインポートする場合はmyca.crt ファイルを使用します。

メモ

CA によって証明書が別のフォームで返される場合は，別個の証明書チェーンとルート CA 証明書を取得する方法について， CA 提供者に問い合わせてください。

完全な証明書チェーンを含んでいる 1 つのファイルで提供された場合，そのファイルからルート CA 証明書フォームをmyca.crt ファイルにコピーします。myca.crt ファイルを使用してnnm-trust.p12 ファイルへインポートすると，NNMi が証明書を発行した CA を信頼するようになります。

2 つのファイルで提供された場合，myca.crt ファイルの内容をmyserver.crt ファイルの末尾に追加します（ファイルに含まれていない場合）。また，余分な中間証明書がある場合は，それらをmyca.crt ファイルからすべて削除します。これにより，次のファイルが生成されます。

- myserver.crt（完全な証明書チェーンを含んでいる）
- myca.crt（ルート CA 証明書を含んでいる）

メモ

CA だけを使用している場合，一般にルート CA 証明書がnnm-trust.p12 ファイルに追加されます。中間 CA またはサーバー証明書を nnm-trust.p12 ファイルに追加すると，それらの証明書は明示的に信頼済みとなり，取り消しなどの追加情報についてのチェックはされません。 CA が要求する場合には，追加の証明書だけをnnm-trust.p12 ファイルに追加してください。

CA 署名機関から受け取るファイルの例を次に示します。

独立サーバーで，複数の CA 証明書ファイルがある場合


```
-----BEGIN CERTIFICATE-----
Sample/AVQQKExNQU0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLEwdOZXR3b3JseGVSZXZvY2F0aW9uTGlzdD9iYXNl
P29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlw
.............................................................
.............................................................
TZImiZPyLGQBGRYDaW50MRIwEAYKCZImiZPyLGQBGRYCc2cxEzARBgNVBAMTCmNbpSo6o/76yShtT7Vrlfz+mXjWyEHa
Iy/QLCpPebYhejHEg4dZgzWWT/lQt==
-----END CERTIFICATE-----
```

結合サーバーで，1 つのファイルに複数の CA 証明書がある場合


```
-----BEGIN CERTIFICATE-----
Sample1/VQQKExNQU0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLEwdOZXR3b3JseGVSZXZvY2F0aW9uTGlzdD9iYXNl
P29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlw
.............................................................
.............................................................
TZImiZPyLGQBGRYDaW50MRIwEAYKCZImiZPyLGQBGRYCc2cxEzARBgNVBAMTCmNbpSo6o/76yShtT7Vrlfz+mXjWyEHa
Iy/QLCpPebYhejHEg4dZgzWWT/lQt==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Sample2/Gh0dHA6Ly9jb3JwMWRjc2cyLnNnLmludC5wc2FnbG9iYWwuY29tL0NlcRaOCApwwggKYMB0GA1UdDgQWBBSq
aWZzCRcpvJWOFPZ/Be9b+QSPyDAfBgNVHSMC
.............................................................
.............................................................
Wp5Lz1ZJAOu1VHbPVdQnXnlBkx7V65niLoaT90Eqd6laliVlJHj7GBriJ90uvVGuBQagggEChoG9bGRhcDovLy9DTj1j
b3JwMWRjc2cyL==
-----END CERTIFICATE-----
```


10.3.3　NNMi キーストアーからの証明書の削除

NNMi キーストアーに同時に複数の証明書を保持することはできません。NNMi 管理サーバー上の証明書を置換または更新する前に，既存の証明書を NNMi キーストアーから削除してください。

NNMi キーストアーから証明書を削除するには，次の手順を実行します。

1. nnm-key.p12 ファイルおよびnnm-trust.p12 ファイルが存在する NNMi 管理サーバーのディレクトリに変更します。
   - Windows の場合：%NnmDataDir%shared¥nnm¥certificates
   - Linux の場合：$NnmDataDir/shared/nnm/certificates
2. nnm-key.p12 ファイルのバックアップコピーを保存します。
3. キーストアーの内容を確認し，既存の証明書のエイリアスを書き留めます。
   - Windows の場合：
     ```
     %NnmInstallDir%bin¥nnmkeytool.ovpl -list -keystore nnm-key.p12 -storetype PKCS12 -stor
     epass nnmkeypass
     ```
   - Linux の場合：
     ```
     $NnmInstallDir/bin/nnmkeytool.ovpl -list -keystore nnm-key.p12 -storetype PKCS12 -stor
     epass nnmkeypass
     ```
4. 次のコマンドを実行して，キーストアーから既存の証明書を削除します。
   - Windows の場合：
     ```
     %NnmInstallDir%bin¥nnmkeytool.ovpl -delete -keystore nnm-key.p12 -storetype PKCS12 -st
     orepass nnmkeypass -alias <alias>
     ```

- Linux の場合：

```
$NnmInstallDir/bin/nnmkeytool.ovpl -delete -keystore nnm-key.p12 -storetype PKCS12 -st
orepass nnmkeypass -alias <alias>
```

> **メモ**
>
> エイリアス（この例では<alias>）は，既存の証明書を識別する名前です。

5. 次のコマンドを実行して，NNMi を再起動します。

> **メモ**
>
> NNMi を再起動するまで，変更は反映されません。

- Windows の場合：

```
%NnmInstallDir%bin¥ovstop -c
%NnmInstallDir%bin¥ovstart -c
```

- Linux の場合：

```
$NnmInstallDir/bin/ovstop -c
$NnmInstallDir/bin/ovstart -c
```

10.3.4 既存の証明書と新規の自己署名証明書または CA 署名証明書との置き換え

自己署名証明書は， NNMi のインストール時に作成され，インストールされます。証明書の置き換えは一般に次の目的で行います。

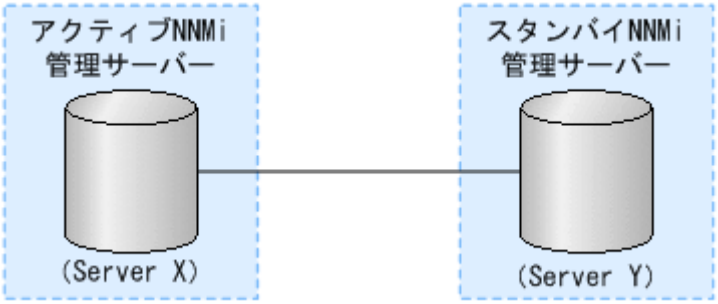
- デフォルトの証明書の代わりに新規の自己署名証明書または CA 署名証明書を使用する。
- 期限の切れた証明書を更新する。

証明書を置き換えるには，次の手順を実行します。

1. 自己署名証明書を生成します。詳細については，「10.3.1　自己署名証明書の生成」を参照してください。
   また，組織で CA の署名した証明書が必要な場合は，CSR（証明書署名要求）ファイルを生成して CA 署名証明書を取得します。詳細については，「10.3.2　CA 署名証明書の生成」を参照してください。
2. 次の構文を使用して，NNMi コンソールへの HTTPS アクセスをテストします。
   `https://<fully_qualified_domain_name>:<port_number>/nnm/.`
   CA 署名証明書を使用した場合，ブラウザーによって CA が信頼されると， NNMi コンソールへの HTTPS 接続が信頼されます。
   自己署名証明書を使用した場合， NNMi コンソールへの信頼性のない HTTPS 接続についての警告メッセージがブラウザーに表示されます。

10.3.5　アプリケーションフェイルオーバー環境での証明書の使用

図 10-1　アプリケーションフェイルオーバーでの証明書の使用法



注意

NNMi 11-50 以降のバージョンでは，証明書を保存するために Public Key Cryptography Standards (PKCS) #12 リポジトリが導入されています。NNMi 11-50 以降の新しいインスタンスをシステムにインストールすると，新しい PKCS #12 ファイルベースの証明書管理方法を利用できます。以前のバージョンの NNMi からアップグレードした環境では，引き続き JKS リポジトリが証明書の格納に使用されます。

アップグレードした環境で，PKCS #12 リポジトリに移行するには，「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従います。

アプリケーションフェイルオーバー機能を設定するときには，両方のノードのトラストストアーファイルの内容をマージして，1 つの`nnm-trust.p12` ファイルを作成する必要があります。

次の手順を実行し，自己署名証明書または CA 署名証明書を使用するようにアプリケーションフェイルオーバー機能を設定します。

注意

NNMi およびアプリケーションフェイルオーバー機能で自己署名証明書を使用する場合，次の手順を完了しないと，NNMi のプロセスがスタンバイ NNMi 管理サーバー（この例の Server Y）で正常に起動しません。

1. Server Y で次のディレクトリに変更します。
   - Windows の場合：`%NnmDataDir%shared¥nnm¥certificates`
   - Linux の場合：`$NnmDataDir/shared/nnm/certificates`
2. `nnm-trust.p12` ファイルを，Server Y から Server X の一時保存場所にコピーします。
   以降の手順では，これらのファイルの保存場所を`<truststore>`と呼びます。
3. Server X で次のコマンドを実行し，Server Y のトラストストアーを Server X の`nnm-trust.p12` ファイルにマージします。

```
nnmcertmerge.ovpl -truststore <truststore>
```

4. マージしたnnm-trust.p12 ファイルを server X から server Y にコピーし，どちらのノードにもマージ済みファイルがあるようにします。

   このファイルの保存場所は，次のとおりです。

   - Windows の場合：%NnmDataDir%shared¥nnm¥certificates
   - Linux の場合：$NnmDataDir/shared/nnm/certificates

5. Server X と Server Y の両方で次のコマンドを実行します。

   完全修飾ドメイン名を含め，両方のサーバーからの表示結果が一致することを確認します。一致しない場合は続行しないで，最初からやり直します。

   Windows の場合：

```
%NnmInstallDir%bin¥nnmkeytool.ovpl -list -keystore
%NnmDataDir%shared¥nnm¥certificates¥nnm-trust.p12 -storetype PKCS12 -storepass ovpass
```

   Linux の場合：

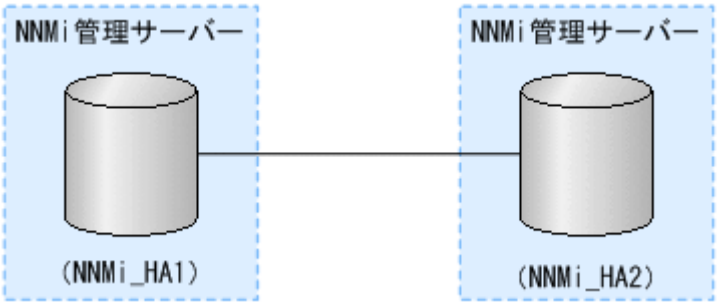
```
$NnmInstallDir/bin/nnmkeytool.ovpl -list -keystore
$NnmDataDir/shared/nnm/certificates/nnm-trust.p12 -storetype PKCS12 -storepass ovpass
```

6. 「18.　アプリケーションフェイルオーバー構成の NNMi を設定する」から，アプリケーションフェイルオーバー機能の設定を続行します。

10.3.6　高可用性環境での証明書の使用

このセクションでは，HA 環境で自己署名証明書または CA 証明書を使用するように NNMi を設定する方法について説明します。

図 10-2　HA での証明書の使用法



注意

NNMi 11-50 以降のバージョンでは，証明書を保存するために Public Key Cryptography Standards (PKCS) #12 リポジトリが導入されています。NNMi 11-50 以降の新しいインスタンスをシステムにインストールすると，新しい PKCS #12 ファイルベースの証明書管理方法を利用できます。以前のバージョンの NNMi からアップグレードした環境では，手動で PKCS #12 リポジトリに移行する必要があります。

アップグレードした環境で，PKCS #12 リポジトリに移行するには，「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従います。

(1)　デフォルト証明書を使用した高可用性の設定

NNMi で HA を正しく有効にするための設定プロセスでは，プライマリクラスタノードとセカンダリクラスタノードの間でデフォルトの自己署名証明書を共有します。HA 下で実行される NNMi でデフォルトの証明書を使用するために，追加の手順を実行する必要はありません。

(2)　新しい証明書を使用した高可用性の設定

このセクションでは，newcert という新規の自己署名証明書または CA 証明書を作成します。次の手順を実行して，この新規の CA 証明書または自己署名証明書を使用するように HA を設定します。

重要

高可用性（HA）でファイルの変更を行うとき，クラスターの両方のノードに変更を加える必要があります。変更によって NNMi 管理サーバーを停止して再起動する必要がある場合，ノードをメンテナンスモードにしてからovstop コマンドおよびovstart コマンドを実行する必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

ヒント

この手順は，「19.5　共有 NNMi データ」の説明に従って，NNMi に HA を設定する前または後に実行できます。

1. NNMi_HA1 で次のディレクトリに変更します。
   - Windows の場合：%NnmDataDir%shared¥nnm¥certificates
   - Linux の場合：$NnmDataDir/shared/nnm/certificates
2. NNMi_HA1 で，次のコマンドを実行して，newcert をnnm-key.p12 ファイルにインポートします。

   Windows の場合：

   ```
   %NnmInstallDir%bin¥nnmkeytool.ovpl -import -alias <newcert_Alias> -storetype PKCS12 -keys
   tore nnm-key.p12 -file newcert -storepass nnmkeypass
   ```

   Linux の場合：

   ```
   $NnmInstallDir/bin/nnmkeytool.ovpl -import -alias <newcert_Alias> -storetype PKCS12 -keys
   tore nnm-key.p12 -file newcert -storepass nnmkeypass
   ```

10.3.7　グローバルネットワーク管理環境での証明書の使用

注意

NNMi 11-50 以降のバージョンでは，証明書を保存するために Public Key Cryptography Standards (PKCS) #12 リポジトリが導入されています。NNMi 11-50 以降の新しいインスタンスをシステムにインストールすると，新しい PKCS #12 ファイルベースの証明書管理方法を利用できます。以前のバージョンの NNMi からアップグレードした環境では，引き続き JKS リポジトリが証明書の格納に使用されます。

アップグレードした環境で，PKCS #12 リポジトリに移行するには，「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従います。

NNMi のインストール時には，インストールスクリプトによって NNMi 管理サーバーの自己署名証明書が作成されます。この証明書には，ノードの完全修飾ドメイン名を含むエイリアスが記録されています。インストールスクリプトは，この自己署名証明書を NNMi 管理サーバーの`nnm-key.p12` ファイルおよび `nnm-trust.p12` ファイルに追加します。

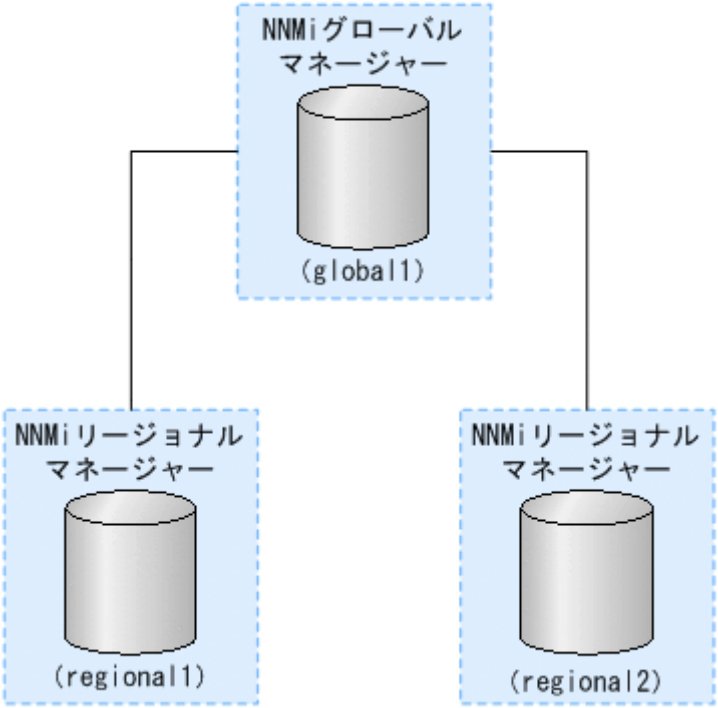
次の手順を実行し，次の図に基づいて自己署名証明書または CA 署名証明書を使用するようにグローバルネットワーク管理機能を設定します。

開始する前に，必要な証明書がリージョナルマネージャーシステムで作成されていることを確認してください。詳細については，「10.3.4　既存の証明書と新規の自己署名証明書または CA 署名証明書との置き換え」を参照してください。

メモ

新たにインストールした NNMi 11-50 以降のインスタンスと，旧バージョンからバージョン 11-50 以降にアップグレードした NNMi 管理サーバーを組み合わせて使用する場合は，「バージョン 11-50 にアップグレードされた NNMi 管理サーバー」のガイドラインに従ってください。

図 10-3　グローバルネットワーク管理



1. regional1 および regional2 で次のディレクトリに変更します。
   - Windows の場合：`%NnmDataDir%shared¥nnm¥certificates`
   - Linux の場合：`$NnmDataDir/shared/nnm/certificates`
2. `nnm-trust.p12` ファイルを，上記の regional1 および regional2 の場所から，global1 の任意の一時保管場所にコピーします。
3. global1 で次のコマンドを実行し，regional1 および regional2 の証明書を global1 の`nnm-trust.p12` ファイルにマージします。

```
nnmcertmerge.ovpl -truststore <regional1_nnm-trust.p12_location>
nnmcertmerge.ovpl -truststore <regional2_nnm-trust.p12_location>
```

4. global1 で，次のコマンドを実行して，NNMi を再起動します。

```
ovstop
ovstart
```

重要

高可用性（HA）でファイルの変更を行う場合は，クラスターの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，`ovstop` コマンドおよび`ovstart` コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

10.3.8　ディレクトリサービスへの SSL 接続を設定する

注意

NNMi 11-50 以降のバージョンでは，証明書を保存するために Public Key Cryptography Standards (PKCS) #12 リポジトリが導入されています。NNMi 11-50 以降の新しいインスタンスをシステムにインストールすると，新しい PKCS #12 ファイルベースの証明書管理方法を利用できます。以前のバージョンの NNMi からアップグレードした環境では，引き続き JKS リポジトリが証明書の格納に使用されます。

アップグレードした環境で，PKCS #12 リポジトリに移行するには，「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従います。

デフォルトでは，ディレクトリサービス通信を有効にすると，NNMi は，ディレクトリサービスからデータを取得するときに LDAP プロトコルを使用します。ディレクトリサービスで SSL 接続が必要な場合は，SSL プロトコルを有効にして，NNMi とディレクトリサービスの間を流れるデータを暗号化する必要があります。

SSL では，ディレクトリサービスホストと NNMi 管理サーバーの間で信頼関係を確立する必要があります。この信頼関係を確立するには，証明書を NNMi トラストストアーに追加します。証明書は，ディレクトリサービスホストの識別情報を NNMi 管理サーバーに示すものです。

SSL 通信用のトラストストアー証明書をインストールするには，次の手順を実行します。

1. ディレクトリサーバーから会社のトラストストアー証明書を取得します。

   ディレクトリサービス管理者からこの証明書のテキストファイルのコピーを入手できます。

   なお，トラストストアー証明書の所有者 CN は，ディレクトリサーバーのホスト名と一致している必要があります。

2. NNMi トラストストアーが格納されているディレクトリに移動します。

   - Windows の場合：`%NnmDataDir%shared¥nnm¥certificates`
   - Linux の場合：`$NnmDataDir/shared/nnm/certificates`

   `certificates` ディレクトリから，この手順のコマンドすべてを実行します。

3. 会社のトラストストアー証明書を NNMi トラストストアーにインポートします。

   メモ

   LDAP ディレクトリサーバーのルート CA 証明書（中間証明書なし）を NNMi トラストストアーにインポートします。

   複数の LDAP ディレクトリサーバーのルート CA 証明書をインポートする必要がある場合，2 個目以降をインポートする際に，手順中の「nnmi_ldap」を，任意の名称に置き換えてください（例：nnmi_ldap2）。

a. 次のコマンドを実行します。

Windows の場合：

```
%NnmInstallDir%bin¥nnmkeytool.ovpl -import -alias nnmi_ldap -storetype PKCS12 -keystor
e nnm-trust.p12 -storepass ovpass -file <Directory_Server_Certificate.txt>
```

Linux の場合：

```
$NnmInstallDir/bin/nnmkeytool.ovpl -import -alias nnmi_ldap -storetype PKCS12 -keystor
e nnm-trust.p12 -storepass ovpass -file <Directory_Server_Certificate.txt>
```

<Directory_Server_Certificate.txt>は，会社のトラストストアー証明書です。

b. 証明書の信頼を確認するメッセージが表示されたら，y と入力します。

証明書をトラストストアーにインポートするときの出力例

このコマンドによる出力形式は次のとおりです。

```
Owner:CN=NNMi_server.example.com
Issuer:CN=NNMi_server.example.com
Serial number:494440748e5
Valid from:Tue Oct 28 10:16:21 MST 2008 until:Thu Oct 04 11:16:21 MDT 2108
Certificate fingerprints:
MD5:29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
SHA1:C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03:7E:C4:03
Trust this certificate?[no]:y
Certificate was added to keystore
```

4. トラストストアーの内容を確認します。

Windows の場合：

```
%NnmInstallDir%bin¥nnmkeytool.ovpl -list -storetype PKCS12 -keystore nnm-trust.p12 -store
pass ovpass
```

Linux の場合：

```
$NnmInstallDir/bin/nnmkeytool.ovpl -list -storetype PKCS12 -keystore nnm-trust.p12 -store
pass ovpass
```

トラストストアーの出力例

トラストストアーの出力形式は次のとおりです。

```
Keystore type:PKCS12
Keystore provider:SunJSSE
Your keystore contains 1 entry
nnmi_ldap, Nov 14, 2008, trustedCertEntry,
Certificate fingerprint (MD5):29:02:D7:D7:D7:D7:29:02:29:02:29:02:29:02:29:02
```

ヒント

トラストストアーには複数の証明書を含めることができます。

5. 次のコマンドを実行して，NNMi を再起動します。

```
ovstop
ovstart
```

重要

高可用性（HA）でファイルの変更を行うとき，クラスターの両方のノードに変更を加える必要があります。変更によって NNMi 管理サーバーを停止して再起動する必要がある場合，ノードをメンテナンスモードにしてから`ovstop` コマンドおよび`ovstart` コマンドを実行する必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

11 NNMi で使用する Telnet および SSH プロトコルの設定

NNMi コンソールを現在実行中の Web ブラウザから［アクション］＞［ノードアクセス］＞［Telnet... (クライアントから)］メニュー項目によって，選択したノードに対するtelnet コマンドが呼び出されます。［アクション］＞［ノードアクセス］＞［Secure Shell... (クライアントから)］メニュー項目によって，選択したノードに対するsecure shell（SSH）コマンドが呼び出されます。デフォルトでは，Microsoft Edge，Mozilla Firefox のいずれでもtelnet コマンドやSSH コマンドは定義されていないため，どちらのメニュー項目を使用する場合でもエラーメッセージが生成されます。システムごとに Telnet プロトコル，SSH プロトコル，または両方のプロトコルを各 NNMi ユーザーに設定して，NNMi コンソールメニュー項目を変更できます。この章では，NNMi で使用する Telnet および SSH プロトコルの設定について説明します。

11.1　Telnet または SSH メニュー項目を無効にする

導入環境の NNMi ユーザーが，NNMi コンソールから Telnet または SSH 接続する必要がない場合は，それぞれのメニュー項目を無効化して NNMi コンソールから削除できます。

NNMi コンソールのメニュー項目の無効化は，NNMi 管理サーバー上で NNMi コンソールにサインインするすべてのユーザーに適用されます。［Telnet］または［Secure Shell］メニュー項目を無効にするには，次の手順を実行します。

1. ［設定］ワークスペースで［ユーザーインタフェース］を展開して，［メニュー項目］を選択する。
2. ［メニュー項目］ビューで，［Telnet... (クライアントから)］行または［Secure Shell... (クライアントから)］行を選択して，ダブルクリックする。
3. ［メニュー項目］フォームで，［有効にする］チェックボックスをオフにしてから，［作成者］フィールドを適切な値に設定する。

   作成者値を変更すると，このメニュー項目は NNMi をアップグレードしても無効化されたままです。
4. フォームを保存し，閉じる。

詳細については，NNMi ヘルプの「NNMi コンソールメニューを制御する」を参照してください。

11.2 Windows 上のブラウザに Telnet または SSH クライアントを設定する

NNMi ユーザーの Web ブラウザにオペレーティングシステム提供の`telnet` コマンドを設定します。この手順は，［アクション］ ＞ ［ノードアクセス］ ＞ ［Telnet... (クライアントから)］メニュー項目を実行する必要がある NNMi ユーザーの各コンピュータおよび Web ブラウザで実行します。

NNMi ユーザーの Web ブラウザにサードパーティの`SSH` コマンドを設定します。この手順は，［アクション］ ＞ ［ノードアクセス］ ＞ ［Secure Shell... (クライアントから)］メニュー項目を実行する必要がある NNMi ユーザーの各コンピュータおよび Web ブラウザで実行します。

このセクションの手順を完了するには，コンピュータの管理権限が必要です。特定の手順は，ブラウザおよびオペレーティングシステムのバージョン（32 ビットまたは 64 ビット）によって異なります。

Microsoft Edge は Chromium 版のみ使用できます。Microsoft Edge が Chromium 版であることを確認するには，［設定など］ ＞ ［ヘルプとフィードバック］ ＞ ［Microsoft Edge について］をクリックします。バージョンの下にテキスト［このブラウザーは、Chromium オープン ソース プロジェクトおよび他のオープンソースソフトウェアに基づいて機能します。］が含まれる場合，この Microsoft Edge は Chromium 版です。

Firefox は 32 ビットバージョンでだけ使用できます。

次の表は，各ブラウザとオペレーティングシステムの組み合わせで使用する手順を示したものです。

表 11-1 Windows での Telnet および SSH 設定手順のマトリクス

Web ブラウザ	Windows オペレーティングシステムアーキテクチャ	適用手順
Microsoft Edge	64 ビット	• 「11.2.1 Windows オペレーティングシステム提供の Telnet クライアント」 • 「11.2.2 サードパーティ Telnet クライアント（標準 Windows）」 • 「11.2.4 サードパーティ SSH クライアント（標準 Windows および Windows on Windows）」
Firefox	32 ビット	• 「11.2.1 Windows オペレーティングシステム提供の Telnet クライアント」 • 「11.2.2 サードパーティ Telnet クライアント（標準 Windows）」 • 「11.2.4 サードパーティ SSH クライアント（標準 Windows および Windows on Windows）」
	64 ビット Windows 10, Windows 11	• 「11.2.2 サードパーティ Telnet クライアント（標準 Windows）」 • 「11.2.4 サードパーティ SSH クライアント（標準 Windows および Windows on Windows）」
	64 ビット Windows 10, Windows 11 以外	• 「11.2.3 サードパーティ Telnet クライアント（Windows on Windows）」

Web ブラウザ	Windows オペレーティングシステムアーキテクチャ	適用手順
Firefox	64 ビット Windows 10, Windows 11 以外	• 「11.2.4　サードパーティ SSH クライアント（標準 Windows および Windows on Windows）」

このセクションのタスクの多くでは Windows レジストリの編集が必要です。レジストリを直接編集せずにシステム上で各ユーザーが実行できる.reg ファイルを作成できます。.reg ファイルの例は，「11.3 Windows レジストリを変更するファイル例」を参照してください。このセクションで説明するタスクの詳細については，次の Microsoft の記事を参照してください。

- Microsoft 提供の Telnet クライアントをインストールする
  http://technet.microsoft.com/en-us/library/cc771275%28WS.10%29.aspx
- Windows レジストリの概要
  http://support.microsoft.com/kb/256986
- Windows レジストリをバックアップおよびリストアする
  http://support.microsoft.com/kb/322756

11.2.1　Windows オペレーティングシステム提供の Telnet クライアント

この手順は，次の場合に適用されます。

- 32 ビットオペレーティングシステム上の 32 ビット Firefox
- 64 ビットオペレーティングシステム上の 64 ビット Microsoft Edge

Web ブラウザで使用するオペレーティングシステム提供の Telnet クライアントを設定するには，次の手順を実行します。

1. Windows 10，Windows 11，Windows Server 2016，Windows Server 2019，または Windows Server 2022 専用）オペレーティングシステムに該当する手順に従い，コンピュータにオペレーティングシステム Telnet クライアントをインストールする。

   Windows 10，または Windows 11

   a　［コントロールパネル］で，［プログラム］をクリックしてから，［プログラムと機能］をクリックします。

   b　［タスク］で，［Windows の機能の有効化または無効化］をクリックします。

   c　［Windows の機能］ダイアログボックスで，［Telnet クライアント］チェックボックスをオンにして，［OK］をクリックします。

   Windows Server 2016，Windows Server 2019，または Windows Server 2022

   a　［サーバーマネージャー］の［ダッシュボード］で，［役割と機能の追加］をクリックします。

   b　［役割と機能の追加ウィザード］で，［Telnet クライアント］チェックボックスをオンにして，［次へ］，［インストール］の順にクリックします。

2. URL:Telnet プロトコルファイルタイプのファイル関連づけを設定する。

   a　Windows レジストリをバックアップします。

   b　Windows レジストリエディタを使用して，［HKEY_CLASSES_ROOT¥telnet¥shell¥open¥command］キーを次の値で変更します。

名前	タイプ	データ
（デフォルト）	REG_SZ	rundll32.exe url.dll,TelnetProtocolHandler %l

   %l（小文字の L）は Telnet に渡される引数で，通常はノードの IP アドレスまたは完全修飾ドメイン名。

   制御を厳しくするには，キーのバイナリへのパスを 1 行としてコード化できます。

   例を次に示します。

```
"C:¥Windows¥system32¥rundll32.exe"
"C:¥Windows¥system32¥url.dll",TelnetProtocolHandler %l
```

3. Web ブラウザを再起動してから，ブラウザのアドレスバーにtelnet コマンドを入力する。

```
telnet://<node>
```

   <*node*>は Telnet サーバーを実行するノードの IP アドレスまたは完全修飾ドメイン名です。セキュリティ警告が表示される場合は，アクションを許可します。Firefox で，［今後 telnet リンクは同様に処理する］チェックボックスをオンにします。

11.2.2　サードパーティ Telnet クライアント（標準 Windows）

この手順は，次の場合に適用されます。

- 32 ビットオペレーティングシステム上の 32 ビット Firefox
- 64 ビットオペレーティングシステム上の 64 ビット Microsoft Edge

Web ブラウザで使用するサードパーティ Telnet クライアントを設定するには，次の手順に従います。

1. サードパーティ Telnet クライアントを取得してインストールする。

   この手順では，C:¥Program Files¥PuTTY¥putty.exe にインストールした PuTTY クライアントを例に挙げます。PuTTY クライアントは，次の Web サイトから使用できます。

```
http://www.putty.org
```

2. URL:Telnet プロトコルファイルタイプのファイル関連づけを設定する。

   a　Windows レジストリをバックアップします。

   b　Windows レジストリエディタを使用して，［HKEY_CLASSES_ROOT¥telnet¥shell¥open¥command］キーを次の値で変更します。

名前	タイプ	データ
（デフォルト）	REG_SZ	"C:¥Program Files¥PuTTY¥putty.exe" %l

%l（小文字の L）は Telnet に渡される引数で，通常はノードの IP アドレスまたは完全修飾ドメイン名。

.reg ファイルでは，各引用符（”）と円記号（¥）は円記号（¥）でエスケープします。

3. Web ブラウザを再起動してから，ブラウザのアドレスバーにtelnet コマンドを入力する。

```
telnet://<node>
```

<node>は Telnet サーバーを実行するノードの IP アドレスまたは完全修飾ドメイン名です。

セキュリティ警告が表示される場合は，アクションを許可します。

Firefox で，［今後 telnet リンクは同様に処理する］チェックボックスをオンにします。

11.2.3 サードパーティ Telnet クライアント（Windows on Windows）

この手順は，次の場合に適用されます。

- 64 ビットオペレーティングシステム上の 32 ビット Firefox

Web ブラウザで使用するサードパーティ Telnet クライアントを設定するには，次の手順に従います。

1. サードパーティ Telnet クライアントを取得してインストールする。

   この手順では，C:¥Program Files¥PuTTY¥putty.exe にインストールした PuTTY クライアントを例に挙げます。PuTTY クライアントは次の Web サイトから使用できます。

```
http://www.putty.org
```

2. URL:Telnet プロトコルファイルタイプのファイル関連づけを設定する。

   a Windows レジストリをバックアップします。

   b Windows レジストリエディタを使用して，［HKEY_CLASSES_ROOT¥Wow6432Node¥telnet¥shell¥open¥command］キーを次の値で変更します。

名前	タイプ	データ
（デフォルト）	REG_SZ	"C:¥Program Files¥PuTTY¥putty.exe" %l

%l（小文字の L）は Telnet に渡される引数で，通常はノードの IP アドレスまたは完全修飾ドメイン名。

.reg ファイルでは，各引用符（”）と円記号（¥）は円記号（¥）でエスケープします。

3. Web ブラウザを再起動してから，ブラウザのアドレスバーにtelnet コマンドを入力する。

```
telnet://<node>
```

<node>は Telnet サーバーを実行するノードの IP アドレスまたは完全修飾ドメイン名です。

セキュリティ警告が表示される場合は，アクションを許可します。

Firefox で，［今後 telnet リンクは同様に処理する］チェックボックスをオンにします。

11.2.4 サードパーティ SSH クライアント（標準 Windows および Windows on Windows）

この手順は，次の場合に適用されます。

- 32 ビットまたは 64 ビットオペレーティングシステム上の 32 ビット Firefox
- 64 ビットオペレーティングシステム上の 64 ビット Microsoft Edge

Web ブラウザで使用するサードパーティ SSH クライアントを設定するには，次の手順を実行します。

1. サードパーティ SSH クライアントを取得してインストールする。

   この手順では，`C:¥Program Files¥PuTTY¥putty.exe` にインストールした PuTTY クライアントを例に挙げます。

   PuTTY は「`ssh://<node>`」入力を正しく構文解析できないため，この例には入力引数から「`ssh://`」を取り除くスクリプトが含まれています。スクリプト`C:¥Program Files¥PuTTY¥ssh.js` には，次のコマンドが含まれます。

   ```
   host = WScript.Arguments(0).replace(/ssh:/,"").replace(/¥//g,"");
   shell = WScript.CreateObject("WScript.Shell");
   shell.Run("¥"c:¥¥Program Files¥¥PuTTY¥¥putty.exe¥" -ssh " + host);
   ```

   このスクリプトはこの例のために作成されたもので，PuTTY には含まれません。

2. SSH プロトコルを定義する。

   a　Windows レジストリをバックアップします。

   b　Windows レジストリエディタを使用して，［HKEY_CLASSES_ROOT¥ssh］キーに次の値を追加します。

名前	タイプ	データ
（デフォルト）	REG_SZ	URL:SSH Protocol
EditFlags	REG_DWORD	2
FriendlyTypeName	REG_SZ	SSH
URL Protocol	REG_SZ	値なし

3. URL:SSH プロトコルファイルタイプのファイル関連づけを設定する。

   a　Windows レジストリをバックアップします。

   b　Windows レジストリエディタを使用して，［HKEY_CLASSES_ROOT¥ssh¥shell¥open¥command］キーを次の値で変更します。

名前	タイプ	データ
（デフォルト）	REG_SZ	"C:¥Windows¥System32¥WScript.exe" "C:¥Program Files¥PuTTY¥ssh.js" %l

   `%l`（小文字の L）は完全 ssh 引数で，プロトコル指定が含まれます。`ssh.js` スクリプトは SSH ターゲットを PuTTY に渡します。

.reg ファイルでは，各引用符（”）と円記号（¥）は円記号（¥）でエスケープします。

4. Web ブラウザを再起動してから，ブラウザのアドレスバーにssh コマンドを入力する。

```
ssh://<node>
```

<*node*>は Telnet サーバーを実行するノードの IP アドレスまたは完全修飾ドメイン名です。

セキュリティ警告が表示される場合は，アクションを許可します。

Firefox で，［今後 ssh リンクは同様に処理する］チェックボックスをオンにします。

11.3　Windows レジストリを変更するファイル例

多くの NNMi ユーザーが Telnet または SSH プロトコルを使用して NNMi コンソールから管理対象ノードにアクセスする必要がある場合は，Windows レジストリ更新を 1 つ以上の.reg ファイルで自動化できます。このセクションには，独自の.reg ファイル作成の基準にできる.reg ファイル例が含まれます。レジストリキーは，アプリケーションとオペレーティングシステムが一致する場合と，64 ビットの Windows バージョンで 32 ビットのアプリケーションを実行する場合では異なるパスにあります。

詳細については，http://support.microsoft.com/kb/310516 の Microsoft の記事を参照してください。

11.3.1　nnmtelnet.reg の例

このレジストリの内容例は，「11.2.1　Windows オペレーティングシステム提供の Telnet クライアント」に適用されます。

```
Windows Registry Editor Version 5.00
[HKEY_CLASSES_ROOT¥telnet¥shell¥open¥command]
@=”¥”C:¥¥Windows¥¥system32¥¥rundll32.exe¥”
¥”C:¥¥Windows¥¥system32¥¥url.dll¥”,TelnetProtocolHandler %l”
```

11.3.2　nnmputtytelnet.reg の例

このレジストリの内容例は，「11.2.2　サードパーティ Telnet クライアント（標準 Windows）」に適用されます。

```
Windows Registry Editor Version 5.00
[HKEY_CLASSES_ROOT¥telnet¥shell¥open¥command]
@=”¥”C:¥¥Program Files¥¥PuTTY¥¥putty.exe¥” %l”
```

11.3.3　nnmtelnet32on64.reg の例

このレジストリの内容例は，「11.2.3　サードパーティ Telnet クライアント（Windows on Windows）」に適用されます。

```
Windows Registry Editor Version 5.00
[HKEY_CLASSES_ROOT¥Wow6432Node¥telnet¥shell¥open¥command]
@=”¥”C:¥¥Program Files¥¥PuTTY¥¥putty.exe¥” %l”
```

11.3.4　nnmssh.reg の例

このレジストリの内容例は，「11.2.4　サードパーティ SSH クライアント（標準 Windows および Windows on Windows）」に適用されます。

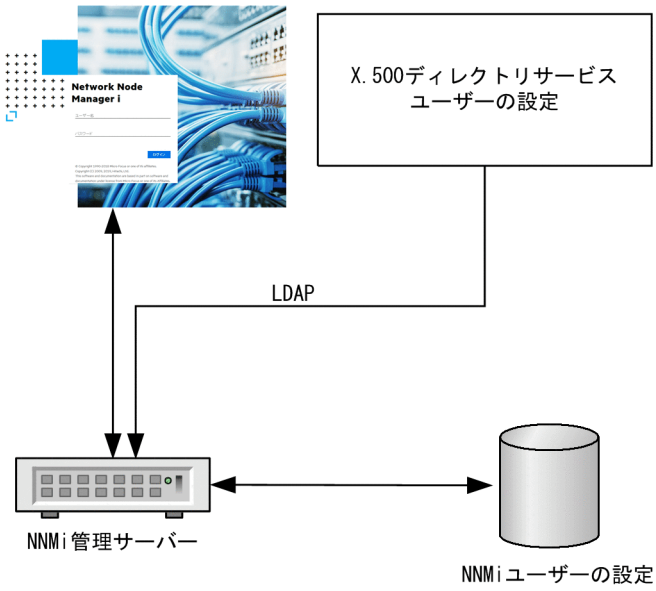
```
Windows Registry Editor Version 5.00
[HKEY_CLASSES_ROOT¥ssh]
@=”URL:ssh Protocol”
”EditFlags”=dword:00000002
”FriendlyTypeName”=”Secure Shell”
”URL Protocol”=””

[HKEY_CLASSES_ROOT¥ssh¥shell¥open¥command] @=”¥”C:¥¥Windows¥¥System32¥¥WScript.exe¥” ¥”c:¥¥P
rogram Files¥¥PuTTY¥¥ssh.js¥” %l”
```

12 NNMi と LDAP によるディレクトリサービスの統合

この章では，NNMi とディレクトリサービスを統合することで，ユーザー名，パスワード，および任意で NNMi ユーザーグループの割り当ての保存場所を統合する方法について説明します。

12.1　NNMi ユーザーのアクセス情報と設定の方法



NNMi ユーザーは，次の項目によって定義されます。

- ユーザー名は，NNMi ユーザーを一意に識別します。ユーザー名によって NNMi へのアクセスが許可され，インシデント割り当てを受け取ることができます。
- パスワードは，ユーザー名と関連づけられ，NNMi コンソールまたは NNMi コマンドへのアクセスを制御するために使用されます。
- NNMi ユーザーグループメンバーシップによって，提供する情報および NNMi コンソールでユーザーが実行可能なアクションのタイプを制御します。ユーザーグループメンバーシップに従って，ユーザーが使用可能な NNMi コマンドの制御も行われます。

重要

お使いの機器が SNMPv1 または SNMPv2c の場合，次の点にご注意ください。

SNMPv1 および SNMPv2c は，クリアテキストで情報パケットを送ります。

セキュアな環境にするため，お使いの機器から発信される SNMP トラップや情報の集まりが流れるように，SNMPv3 を使用するか，またはファイアウォールコントロールなどの保護を追加してください。

NNMi には，NNMi ユーザーアクセス情報の保存先として幾つかの方法が用意されています。

設定の方法ごとに NNMi ユーザーアクセス情報を保存するデータベースを，次に示します。

表 12-1　ユーザー情報の保存オプション

モード	ユーザーアカウント	ユーザーグループ	ユーザーグループメンバーシップ
内部（オプション 1）	NNMi	NNMi	NNMi
混合（オプション 2）	混合（NNMi のアカウント名，LDAP のアカウントのパスワード）	NNMi	NNMi
外部（オプション 3）	ディレクトリサービス	両方	ディレクトリサービス

NNMi は，LDAP（Lightweight Directory Access Protocol）を使用してディレクトリサービスと通信します。NNMi で LDAP を使用する場合は，表に示す次のどちらかの方法を使用します。

- 混合モード（元の名称は「オプション 2」）：NNMi ユーザー情報の一部を NNMi データベースに，一部をディレクトリサービスに格納します。

  混合モードを使用するには，ユーザー名，ユーザーグループ，およびユーザーグループのマッピングを NNMi データベースに格納し，ユーザー名とパスワード（ユーザーアカウントの定義）をディレクトリサービスに格納するように設定します。つまり，アカウント名の情報は NNMi と LDAP の両方に格納する必要がありますが，アカウントのパスワードは LDAP だけに格納します。
- 外部モード（元の名称は「オプション 3」）：すべての NNMi ユーザー情報をディレクトリサービスに格納します。

  外部モードを使用する場合は，すべてのユーザーアカウント情報が LDAP を使用して格納されますので，NNMi にユーザーアカウント情報を追加する必要はありません。

混合モードを使用して新規ユーザーアカウントを追加するか既存アカウントを修正する場合は，［ディレクトリサービスアカウント］のチェックボックスを選択する必要があります。ユーザーアカウントを設定する際に，内部モード，混合モードおよび外部モードを組み合わせて使用する方法として，一部のユーザーについては［ディレクトリサービスアカウント］を選択し，またほかのユーザーについては選択しないという設定は避けてください。このような設定は，サポート対象外です。

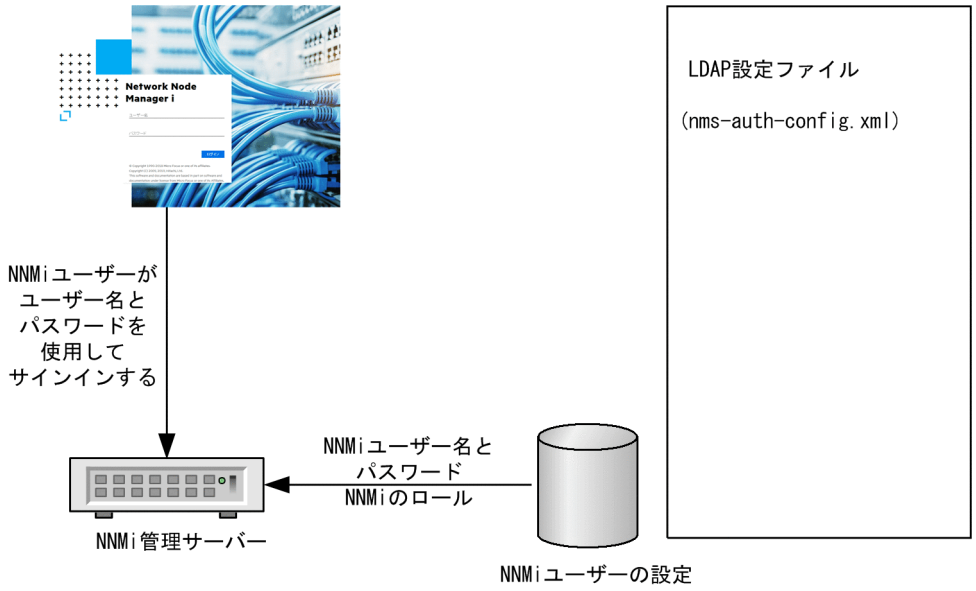
12.1.1　内部モード：NNMi データベースにすべての NNMi ユーザー情報を保存

NNMi は，すべてのユーザーアクセス情報を取得するために NNMi データベースにアクセスします。それらの情報は，NNMi 管理者が NNMi コンソールで定義およびメンテナンスします。ユーザーアクセス情報は，NNMi にとってローカルの情報となります。NNMi はディレクトリサービスにアクセスしません。また，NNMi は LDAP 設定ファイルから情報を取得するように設定されていません。

この方法での情報フローを次の図に示します。この情報フローは，次のような状況に適しています。

- NNMi ユーザーの数が少ない。
- ディレクトリサービスを使用していない。

図 12-1　内部モードの NNMi ユーザーサインインの情報フロー



12.1.2　混合モード：一部の NNMi ユーザー情報を NNMi データベースに，一部の NNMi ユーザー情報をディレクトリサービスに保存

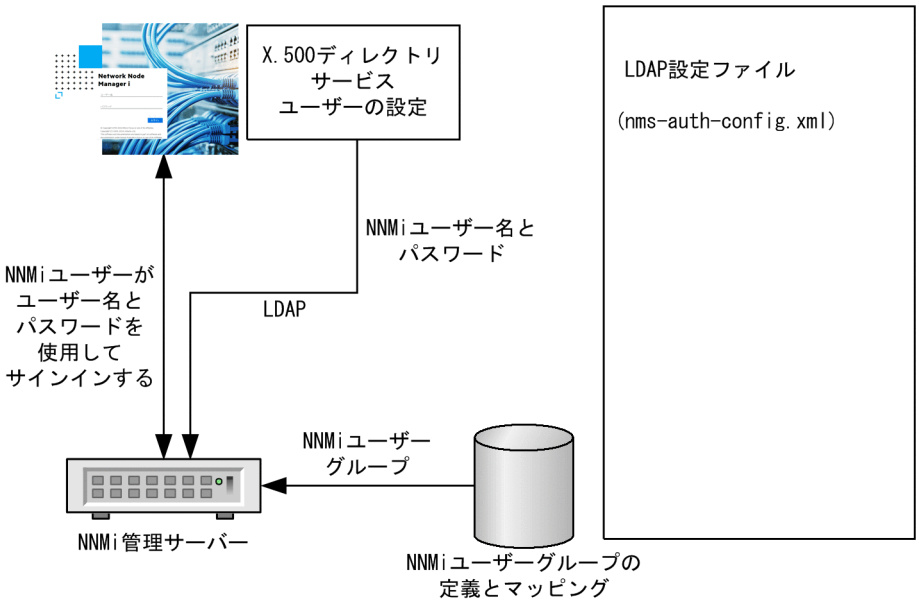
NNMi は，ユーザー名とパスワードを取得するためにディレクトリサービスにアクセスします。それらの情報は，NNMi の外部で定義され，ほかのアプリケーションでも使用できます。ユーザーから NNMi ユーザーグループへのマッピングは，NNMi コンソールでメンテナンスします。NNMi ユーザーアクセス情報の設定およびメンテナンスは，次で説明するように共同で行われます。

- ディレクトリサービス管理者は，ディレクトリサービス内のユーザー名とパスワードをメンテナンスします。
- NNMi 管理者は，（ディレクトリサービスで定義されている）ユーザー名，ユーザーグループ定義，ユーザーグループのマッピングを NNMi コンソールで入力します。
- NNMi 管理者は，NNMi に対するユーザー名のディレクトリサービスデータベーススキーマを記述する NNMi の LDAP 設定ファイルを設定します。

  次の図のコマンドラインは，NNMi が NNMi ユーザーグループ情報をディレクトリサービスから引き出さないことを示しています。

ユーザー名は，2 か所で入力する必要があるため，両方の場所でユーザー名のメンテナンスを行う必要があります。

図 12-2　混合モードの NNMi ユーザーサインインの情報フロー



この図では，この方法での情報フローを示しています。この情報フローは，次のような状況に適しています。

- NNMi ユーザーの数が少なく，ディレクトリサービスを使用できる。
- ユーザーグループの変更ごとにディレクトリサービスの変更を必要とするのではなく，NNMi 管理者がユーザーグループを管理する。
- ディレクトリサービスのグループ定義を使用できる。

ユーザー名とパスワードを保存するディレクトリサービスとの統合に関する詳細については，この章の以降の説明と，NNMi ヘルプの「NNMi アクセスを制御するための Lightweight Directory Access Protocol (LDAP)」を参照してください。

12.1.3　外部モード：すべての NNMi ユーザー情報をディレクトリサービスに保存

NNMi は，すべてのユーザーアクセス情報を取得するためにディレクトリサービスにアクセスします。それらの情報は，NNMi の外部で定義され，ほかのアプリケーションが使用できます。1 つ以上のディレクトリサービスグループでのメンバーシップで，ユーザーの NNMi ユーザーグループが決まります。

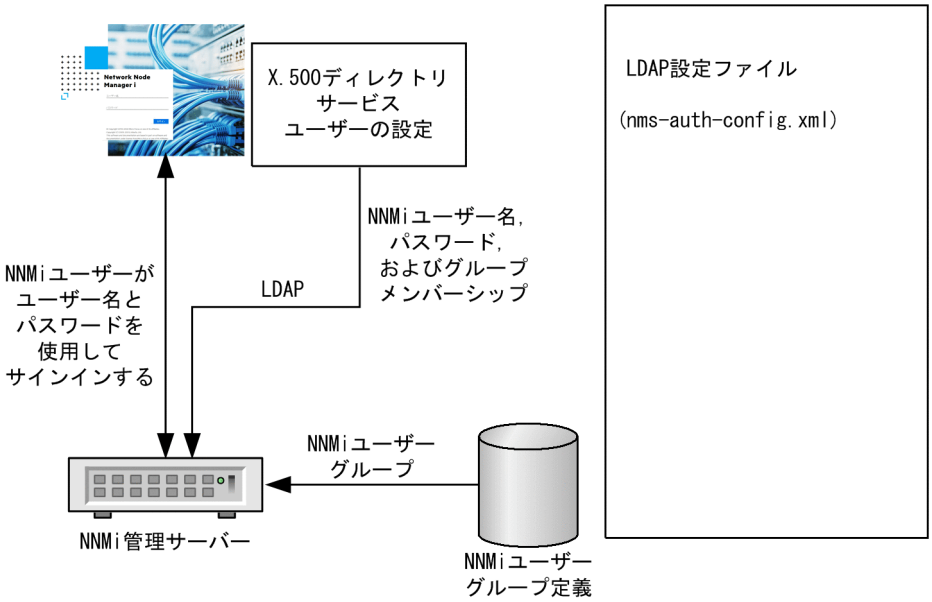
NNMi ユーザーアクセス情報の設定およびメンテナンスは，次で説明するように共同で行われます。

- ディレクトリサービス管理者は，ディレクトリサービス内のユーザー名，パスワード，グループメンバーシップをメンテナンスします。
- NNMi 管理者は，ディレクトリサービスグループを NNMi ユーザーグループに NNMi コンソールでマッピングします。

- NNMi 管理者は，NNMi に対するユーザー名およびグループのディレクトリサービスデータベーススキーマを記述する NNMi の LDAP 設定ファイルを設定します。

次の図に，この方法での情報フローを示します。これは，NNMi にアクセスする必要があるユーザーで構成されるユーザーグループを含めるようにディレクトリサービスを変更できる環境に適しています。

図 12-3　外部モードの NNMi ユーザーサインインの情報フロー



この方法は混合モードの例を拡張した形態であるため，次の設定プロセスを推奨します。

1. ディレクトリサービスから NNMi ユーザー名とパスワードを取得するよう設定して検証する。
2. ディレクトリサービスから NNMi ユーザーグループを取得するよう設定する。

すべてのユーザー情報を保存するディレクトリサービスとの統合に関する詳細については，この章の以降の説明と，NNMi ヘルプの「NNMi アクセスを制御するための Lightweight Directory Access Protocol (LDAP)」を参照してください。

12.2　ディレクトリサービスへのアクセスを設定する

ディレクトリサービスへのアクセスは，nms-auth-config.xml ファイルで設定できます。

このファイルは，次の場所にあります。

- Windows の場合：%NnmDataDir%nmsas¥NNM¥conf
- Linux の場合：$NnmDataDir/nmsas/NNM/conf

デフォルトでは，この場所にあるnms-auth-config.xml ファイルには LDAP 設定に必要な XML エレメントは含まれていません。

必要なすべての XML エレメントをこのファイルに手動で追加するには，このセクションの手順に従います。

NNMi によって，参照用のサンプルnms-auth-config.xml ファイルが別の場所に格納されます。

サンプルnms-auth-config.xml ファイルは次の場所にあります。

- Windows の場合：%NnmInstallDir%newconfig¥HPOvNnmAS¥nmsas¥conf
- Linux の場合：$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf

> **ヒント**
>
> サンプルnms-auth-config.xml ファイルから<ldapLogin>エレメント全体をコピーし，必要な変更を加えることもできます。

nms-auth-config.xml ファイルの詳細については，「12.6　LDAP 設定ファイルリファレンス」を参照してください。

ディレクトリサービスの一般的な構造の詳細については，「12.3　ディレクトリサービスのクエリー」を参照してください。

「混合モード」の設定の場合は，次のタスクを実行します。

- 12.2.1　タスク 1：現在の NNMi ユーザー情報をバックアップする
- 12.2.2　タスク 2：（任意）ディレクトリサービスへのセキュア接続を設定する
- 12.2.3　タスク 3：ディレクトリサービスからのユーザーアクセスを設定する
- 12.2.4　タスク 4：ユーザー名とパスワードの設定をテストする
- 12.2.9　タスク 9：クリーンアップして NNMi の予期せぬアクセスを防止する
- 12.2.10　タスク 10：（任意）ユーザーグループをセキュリティグループにマッピングする

「外部モード」の設定の場合は，次のタスクを実行します。

- 12.2.1　タスク 1：現在の NNMi ユーザー情報をバックアップする

- 12.2.2　タスク 2：（任意）ディレクトリサービスへのセキュア接続を設定する
- 12.2.3　タスク 3：ディレクトリサービスからのユーザーアクセスを設定する
- 12.2.4　タスク 4：ユーザー名とパスワードの設定をテストする
- 12.2.5　タスク 5：（「外部モード」の設定だけ）ディレクトリサービスからのグループの取得を設定する

> **メモ**
>
> ディレクトリサービスに NNMi ユーザーグループを保存する場合は，NNMi ユーザーグループによってディレクトリサービスを設定する必要があります。詳細については，「12.4 NNMi ユーザーグループを保存するディレクトリサービスの設定」を参照してください。

- 12.2.6　タスク 6：（「外部モード」の設定だけ）ディレクトリサービスグループを NNMi ユーザーグループにマッピングする
- 12.2.7　タスク 7：（「外部モード」の設定だけ）NNMi ユーザーグループ設定をテストする
- 12.2.8　タスク 8：（「外部モード」の設定だけ）インシデント割り当ての NNMi ユーザーグループを設定する
- 12.2.9　タスク 9：クリーンアップして NNMi の予期せぬアクセスを防止する
- 12.2.10　タスク 10：（任意）ユーザーグループをセキュリティグループにマッピングする

12.2.1　タスク 1：現在の NNMi ユーザー情報をバックアップする

NNMi データベースのユーザー情報をバックアップします。

```
nnmconfigexport.ovpl -c account -u <user> -p <password> -f NNMi_database_accounts.xml
```

12.2.2　タスク 2：（任意）ディレクトリサービスへのセキュア接続を設定する

ディレクトリサービスで Secure Socket Layer（SSL）を使用する必要がある場合は，「10.3.8　ディレクトリサービスへの SSL 接続を設定する」の説明に従って，自社の証明書を NNMi トラストストアーにインポートします。

12.2.3　タスク 3：ディレクトリサービスからのユーザーアクセスを設定する

「混合モード」および「外部モード」の場合のみ次のタスクを実行します。ディレクトリサービスに応じた適切な手順に従ってください。

1. 次のディレクトリに移動します。
   - Windows の場合：%NnmDataDir%nmsas¥NNM¥conf
   - Linux の場合：$NnmDataDir/nmsas/NNM/conf
2. NNMi に付属するnms-auth-config.xml ファイルをバックアップしてから，そのファイルを任意のテキストエディタで開きます。
3. 次のエレメントの値を指定します。

ヒント

NNMi によって，参照用のサンプルnms-auth-config.xml ファイルが別の場所に格納されます。 サンプルnms-auth-config.xml ファイルは次の場所にあります。

- Windows の場合：%NnmInstallDir%newconfig¥HPOvNnmAS¥nmsas¥conf
- Linux の場合：$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf

サンプルnms-auth-config.xml ファイルから<ldapLogin>エレメント全体をコピーし，必要な変更を加えることもできます。

表 12-2　nms-auth-config.xml ファイルの ldapLogin セクションのエレメント

<enabled> </enabled>		true に設定してnms-auth-config.xml ファイルの使用を指定します。デフォルトでは，false に設定されています。

<userRoleFilterList>
</userRoleFilterList>
		NNMi ユーザーがインシデントを割り当てることのできる NNMi ロールを指定します。
		すべてのオペレータ，管理者，およびゲストにインシデントを 割り当てるには，次を追加します。
		`<userRoleFilterList>`
		`admin guest level2 level1`
		`</userRoleFilterList>`

<connectTimeLimit> </connectTimeLimit>		接続のタイムアウト値をミリ秒単位で指定します。デフォルト値は10000（10 秒）です。NNMi ユーザーのサインイン中にタイムアウトになる場合は，この値を増やします。 例：`<connectTimeLimit>10000</connectTimeLimit>`
<searchTimeLimit> </searchTimeLimit>		検索のタイムアウト値をミリ秒単位で指定します。デフォルト値は30000（30 秒）です。NNMi ユーザーのサインイン中にタイムアウトになる場合は，この値を増やします。 例：`<searchTimeLimit>30000</searchTimeLimit>`
<server>		すべての LDAP 設定情報を含むコンテナエレメント。
	<hostname> </hostname>	LDAP サーバーの URL とポート番号。 例： • HTTP を使用する場合： ldap://hostname.domain.com:389 • HTTPS を使用する場合： ldaps://hostname.domain.com:636

	<hostname> </hostname>	注：HTTP を使用する場合は，ldap://と指定してください。HTTPS を使用する場合は，ldap://またはldaps://と指定してください。
	<secure> </secure>	HTTPS を使用する場合は，true に設定します。HTTPS を使用しない場合は，false に設定します。
</server>		
注：server エレメントは，冗長化された LDAP サーバー構成で，複数台から同じ情報が取得できる場合，複数指定することができます。その場合，上に書いた接続先から順に接続を試行します。		
<bindCredential>		バインド資格証明が含まれているコンテナエレメント（匿名ログオンをサポートしていないディレクトリサービスでは必須）。
	<bindDN> </bindDN>	バインド DN を指定します。
	<bindCredential> </bindCredential>	バインド DN のパスワードを暗号化された形式で指定します。 ”nnmldap.ovpl -encrypt <mypassword>”コマンドを実行してパスワードを暗号化します。
</bindCredential>		
<users>		すべてのユーザー設定情報が含まれているコンテナエレメント。
	<userSearch>	ユーザーを検索するための設定情報が含まれているコンテナエレメント。 <userSearch></userSearch>の設定は 1 つだけ設定してください。複数設定することはサポートしていません。
	<base> </base>	例： • Active Directory の場合： <base> CN={0} </base> • その他の LDAP 技術の場合： <base> SAMAccountName={0} </base>
	<baseContextDN> </baseContextDN>	Active Directory の場合，ディレクトリサービスのドメインでユーザーレコードを保存する部分を指定します。 例： • Active Directory の場合： OU=Users,OU=Accounts,DC=mycompany,DC=com • その他の LDAP 技術の場合： ou=People,o=example.com
	</userSearch>	
</users>		
注：混合モードの場合は，<roleSearch></roleSearch>を次のように 1 つだけ設定してから，手順 4 を実行してください。 <roleSearch> <roleBase></roleBase> <roleContextDN></roleContextDN> </roleSearch>		

4. nms-auth-config.xml ファイル（%NnmDataDir%nmsas¥NNM¥conf（Windows）または$NnmDataDir/nmsas/NNM/conf（Linux））の編集後，次のコマンドを実行します。

- Windows の場合：

```
%NnmInstallDir%bin¥nnmldap.ovpl -reload
```

- Linux の場合：

```
$NnmInstallDir/bin/nnmldap.ovpl -reload
```

12.2.4　タスク 4：ユーザー名とパスワードの設定をテストする

1. LDAP 設定ファイルで，テスト用にdefaultRole をguest と設定する。

   この値はいつでも変更できます。

   nms-auth-config.xml ファイルで，次の内容をusersearch エレメントよりも前に追加します。

   ```
   <defaultRoles>
   <role>guest</role>
   </defaultRoles>
   ```

2. LDAP 設定ファイルを保存する。
3. 次のコマンドを実行して，NNMi に LDAP 設定ファイルを再読み込みさせる。

   ```
   nnmldap.ovpl -reload
   ```

4. ディレクトリサービスで定義されているユーザー名とパスワードを使用して，NNMi コンソールにサインインする。

   このテストは，NNMi データベースでまだ定義されていないユーザー名を使用して実行してください。

5. NNMi コンソールのタイトルバーで，ユーザー名と NNMi ロール（ゲスト）を確認する。
   - ユーザーサインインが正しく動作したら，このタスクの手順 8.に進みます。
   - ユーザーサインインが正しく動作しない場合は，次は手順 6.に進みます。

   各テストのあとで，NNMi コンソールからサインアウトしてセッション資格証明をクリアします。

6. 次のコマンドを実行し，あるユーザーの設定をテストする。

   ```
   nnmldap.ovpl -diagnose <NNMi_user>
   ```

   *<NNMi_user>*は，ディレクトリサービスで定義した NNMi ユーザーのサインイン名で置き換えます。

   コマンド出力を検討し，適切に応答します。推奨事項は次のとおりです。

   - 「12.2.3　タスク 3：ディレクトリサービスからのユーザーアクセスを設定する」が正常に完了したことを確認します。
   - 「12.3.4　ユーザー識別」の詳細な設定プロセスに従います。

メモ

混合モードの場合，次のようなメッセージが出力されますが，混合モードでは LDAP グループを参照しないため，動作に問題ありません。次のメッセージは無視してください。

```
!!!!!!!!!!!!!!!!!!!!!!!! NOTE !!!!!!!!!!!!!!!!!!!!!!!!
!  このユーザー識別名のLDAPグループが見つかりません。
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


!!!!!!!!!!!!!!!!!!!!!!!! NOTE !!!!!!!!!!!!!!!!!!!!!!!!
!  LDAPの設定が誤っているようです。詳細は、上記を参照してください。
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

7. NNMi コンソールへのサインイン時に期待する結果が表示されるまで，手順 1.から手順 5.を繰り返す。
8. サインインできたら，設定方法を選択する。
   - NNMi ユーザーグループメンバーシップを NNMi データベースに保存する（「混合モード」の設定）場合は，「12.2.9　タスク 9：クリーンアップして NNMi の予期せぬアクセスを防止する」に進みます。
   - NNMi ユーザーグループメンバーシップをディレクトリサービスに保存する（「外部モード」の設定）場合は，次はタスク 5 に進みます。

12.2.5　タスク 5：（「外部モード」の設定だけ）ディレクトリサービスからのグループの取得を設定する

このタスクは，「外部モード」の場合に実行します。ディレクトリサービスに応じた適切な手順に従ってください。

1. 次のディレクトリに移動します。
   - Windows の場合：`%NnmDataDir%nmsas¥NNM¥conf`
   - Linux の場合：`$NnmDataDir/nmsas/NNM/conf`
2. `nms-auth-config.xml` ファイルのバックアップを作成し，このファイルをテキストエディタで開きます。
3. 次のエレメントの値を指定します。

ヒント

NNMi によって，参照用のサンプル`nms-auth-config.xml` ファイルが別の場所に格納されます。 サンプル`nms-auth-config.xml` ファイルは次の場所にあります。

- Windows の場合：`%NnmInstallDir%newconfig¥HPOvNnmAS¥nmsas¥conf`
- Linux の場合：`$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf`

サンプル`nms-auth-config.xml` ファイルから<ldapLogin>エレメント全体をコピーし，必要な変更を加えることもできます。

表 12-3　nms-auth-config.xml ファイルの ldapLogin セクションのエレメント

<roleSearch>		ユーザーロール情報が含まれているプレースホルダーエレメント。 `<roleSearch></roleSearch>`の設定は 1 つだけ設定してください。複数設定することはサポートしていません。
	<roleBase>member= {1} </roleBase>	member を，ディレクトリサービスドメインのディレクトリサービスユーザー ID を保存するグルー プ属性の名前で置き換えます。

	<roleContextDN>
	</roleContextDN>
		ディレクトリサーバードメインの中でグループレコードを保存する部分を指定します。
		形式は，ディレクトリサービスの属性名と値のカンマ区切りリストです。
		例：
		• Active Directory の場合：
		`CN=Users,DC=ldapserver,DC=mycompany,DC=com`
		• その他の LDAP 技術の場合：
		`ou=Groups,o=example.com`

</roleSearch>		

4. ファイルを保存します。
5. 次のコマンドを実行します。

```
nnmldap.ovpl -reload
```

12.2.6　タスク 6：（「外部モード」の設定だけ）ディレクトリサービスグループを NNMi ユーザーグループにマッピングする

1. NNMi コンソールで，定義済みの NNMi ユーザーグループをディレクトリサービスのユーザーグループにマッピングする。
   a. **[ユーザーグループ]** ビューを開きます。
      [設定] ワークスペースで **[セキュリティ]** を展開してから **[ユーザーグループ]** をクリックします。
   b. **[admin]** 行をダブルクリックします。
   c. **[ディレクトリサービス名]** フィールドに，NNMi 管理者のディレクトリサービスグループの完全識別名を入力します。
   d. **[保存して閉じる]** をクリックします。
   e. `guest`，`level1`，`level2` の行ごとに手順 b から手順 d を繰り返します。

   このマッピングによって，NNMi コンソールにアクセスできるようになります。NNMi コンソールにアクセスするすべてのユーザーは，この手順で指定した，定義済みの NNMi ユーザーグループのうちどれかにマッピングされているディレクトリサービスグループに含まれている必要があります。

2. ディレクトリサービスで 1 人以上の NNMi ユーザーを含むそのほかのグループに，NNMi コンソールで新しいユーザーグループを作成する。

   a. **[ユーザーグループ]** ビューを開きます。

      [設定] ワークスペースで **[セキュリティ]** を展開してから **[ユーザーグループ]** をクリックします。

   b. **[新規作成]** をクリックしてから，グループの情報を入力します。

      - **[名前]** は一意の値に設定します。短い名前にすることをお勧めします。
      - **[表示名]** は，ユーザーに表示される値に設定します。
      - **[ディレクトリサービス名]** は，ディレクトリサービスグループの完全識別名に設定します。
      - **[説明]** は，この NNMi ユーザーグループの目的を説明するテキストに設定します。

   c. **[保存して閉じる]** をクリックします。

   d. NNMi ユーザーのディレクトリサービスグループごとに手順 b と手順 c を繰り返します。

   このマッピングによって，NNMi コンソールのトポロジオブジェクトにアクセスできるようになります。各ディレクトリサービスグループは，複数の NNMi ユーザーグループにマッピングできます。

12.2.7 タスク 7：（「外部モード」の設定だけ）NNMi ユーザーグループ設定をテストする

1. NNMi の LDAP 設定ファイル（`nms-auth-config.xml` ファイル）を保存する。
2. 次のコマンドを実行して，NNMi に LDAP 設定ファイルを再読み込みさせる。

```
nnmldap.ovpl -reload
```

3. ディレクトリサービスで定義されているユーザー名とパスワードを使用して，NNMi コンソールにサインインする。

   NNMi データベースでまだ定義されていないで，`admin`，`level1`，`level2` の NNMi ユーザーグループにマッピングされているディレクトリサービスグループのメンバーであるユーザー名で，このテストを実行します。

4. ユーザー名と NNMi ロール（**[ユーザーグループ]** ビューの **[表示名]** フィールドで定義したもの）を NNMi コンソールのタイトルバーで，確認する。

   - ユーザーサインインが正しく動作したら，タスク 8 に進みます。
   - ユーザーサインインが正しく動作しない場合は，次は手順 5.に進みます。

   各テストのあとで，NNMi コンソールからサインアウトしてセッション資格証明をクリアします。

5. 次のコマンドを実行し，ユーザーの設定をテストする。

```
nnmldap.ovpl -diagnose <NNMi_user>
```

   *<NNMi_user>*は，ディレクトリサービスで定義した NNMi ユーザーのサインイン名で置き換えます。

   コマンド出力を検討し，適切に応答します。推奨事項は次のとおりです。

- 「12.2.5　タスク 5：(「外部モード」の設定だけ）ディレクトリサービスからのグループの取得を設定する」が正常に完了したことを確認します。
- 定義済みの NNMi ユーザーグループごとに，「12.2.6　タスク 6：(「外部モード」の設定だけ）ディレクトリサービスグループを NNMi ユーザーグループにマッピングする」が正常に完了したことを確認します。
- 「12.3.5　ユーザーグループ識別」の詳細な設定プロセスに従います。

6. NNMi コンソールへのサインイン時に期待する結果が表示されるまで，手順 1.から手順 4.を繰り返す。

12.2.8　タスク 8：(「外部モード」の設定だけ）インシデント割り当ての NNMi ユーザーグループを設定する

1. ディレクトリサービスで定義されているユーザー名とパスワードを使用して，NNMi コンソールにサインインする。
2. 任意のインシデントビューでインシデントを選択し，［アクション］ ＞ ［割り当て］ ＞ ［インシデントの割り当て］ をクリックする。

   `userRoleFilterList` パラメーターによって指定されている各 NNMi ロールのユーザーに，インシデントを割り当てることができることを確認します。

12.2.9　タスク 9：クリーンアップして NNMi の予期せぬアクセスを防止する

1. (任意）LDAP 設定ファイルで，`defaultRole` パラメーターの値を変更するか，またはコメントにする。

   `defaultRole` パラメーターの値を変更した，またはコメントにした場合は，次のコマンドを実行して，NNMi に LDAP 設定ファイルを再読み込みさせてください。

   ```
   nnmldap.ovpl -reload
   ```

2. (「混合モード」の設定だけ）NNMi データベースにユーザーグループメンバーシップを保存するには，次の手順を実行して，NNMi データベースのユーザーアクセス情報をリセットする。

   a. 既存のユーザーアクセス情報すべてを削除します（［ユーザーアカウント］ ビューのすべての行を削除します)。

      詳細については，NNMi ヘルプの「ユーザーアカウントを削除する」を参照してください。

   b. NNMi ユーザーごとに，ユーザー名の ［ユーザーアカウント］ ビューに新しいオブジェクトを作成します。

      – ［名前］ フィールドに，ディレクトリサービスに定義されているユーザー名を入力します。
      – ［ディレクトリサービスアカウント］ チェックボックスを選択します。
      –パスワードは指定しないでください。

      詳細については，NNMi ヘルプの「ユーザーアカウントタスク」を参照してください。

c. NNMi ユーザーごとに，1 つ以上の NNMi ユーザーグループにユーザーアカウントをマッピングします。

詳細については，NNMi ヘルプの「ユーザーアカウントをユーザーグループにマップする ([ユーザーアカウントのマッピング] フォーム)」を参照してください。

d. インシデント所有権を更新して，各割り当てインシデントが有効なユーザー名と関連づけられるようにします。

詳細については，NNMi ヘルプの「インシデント割り当てを管理する」を参照してください。

3. (「外部モード」の設定だけ）ディレクトリサービスからのユーザーグループメンバーシップを使用するには，次の手順を実行して，NNMi データベースのユーザーアクセス情報をリセットする。

a. 既存のユーザーアクセス情報すべてを削除します（**[ユーザーアカウント]** ビューのすべての行を削除します)。

詳細については，NNMi ヘルプの「ユーザーアカウントを削除する」を参照してください。

b. インシデント所有権を更新して，各割り当てインシデントが有効なユーザー名と関連づけられるようにします。

詳細については，NNMi ヘルプの「インシデント割り当てを管理する」を参照してください。

12.2.10 タスク 10：(任意）ユーザーグループをセキュリティグループにマッピングする

詳細については，NNMi ヘルプの「セキュリティグループマッピングタスク」を参照してください。

12.3　ディレクトリサービスのクエリー

NNMi は，LDAP を使用してディレクトリサービスと通信します。NNMi が要求を送信すると，ディレクトリサービスは保存されている情報を返します。NNMi は，ディレクトリサービスに保存されている情報を変更できません。

12.3.1　ディレクトリサービスアクセス

LDAP は，次の形式でディレクトリサービスに対してクエリーを実行します。

```
ldap://<directory_service_host>:<port>/<search_string>
```

- ldap はプロトコル指定子です。この指定子は，ディレクトリサービスへの標準接続と SSL 接続の両方で使用してください。
- *<directory_service_host>*は，ディレクトリサービスをホストするコンピュータの完全修飾名です。
- *<port>*は，LDAP 通信でディレクトリサービスが使用するポートです。非 SSL 接続のデフォルトポートは 389 です。SSL 接続のデフォルトポートは 636 です。
- *<search_string>*には要求情報が指定されます。詳細については，「12.3.2　ディレクトリサービスの情報」と，次のサイトにある RFC 1959「An LDAP URL Format」を参照してください。

  ```
  http://www.ietf.org/rfc/rfc1959.txt
  ```

Web ブラウザで LDAP クエリーを URL として入力し，アクセス情報が正しく，検索文字列の構造が正しいことを確認できます。

ディレクトリサービス（例えば，Active Directory）が匿名アクセスを許可しない場合，そのディレクトリは Web ブラウザからの LDAP クエリーを拒否します。この場合は，サードパーティ製の LDAP ブラウザ（Apache Directory Studio に含まれる LDAP ブラウザなど）を使用し，設定パラメーターの有効性を検証できます。

12.3.2　ディレクトリサービスの情報

ディレクトリサービスには，ユーザー名，パスワード，およびグループメンバーシップなどの情報が保存されています。ディレクトリサービス内の情報にアクセスするには，情報の保存場所を参照する識別名を知っている必要があります。サインインアプリケーションの場合の識別名は，可変情報（ユーザー名など）と固定情報（ユーザー名の保存場所など）の組み合わせです。識別名を構成するエレメントは，ディレクトリサービスの構造と内容によって決まります。

次の例は，USERS-NNMi-Admin というユーザーグループの場合に考えられる定義を示しています。このグループは，NNMi への管理アクセス権を持つディレクトリサーバーのユーザー ID のリストで構成されます。次の情報は，これらの例に関係しています。

- Active Directory の例は，Windows オペレーティングシステムの場合です。
- ほかのディレクトリサービスの例は，Linux オペレーティングシステムの場合です。
- それぞれの例に示すファイルは，LDIF（lightweight directory interchange format）ファイルの一部です。LDIF ファイルによって，ディレクトリサービスの情報を共有できます。
- それぞれの例の図は，ディレクトリサービスドメインをグラフィカルに表現したものです。この図は，引用した LDIF ファイルに含まれる情報を拡張して表示したものです。

Active Directory の情報構造例

この例での関心の対象は次の項目です。

- ユーザー John Doe の識別名：

```
CN=john.doe@example.com,OU=Users,OU=Accounts,DC=example,DC=com
```

- USERS-NNMi-Admin グループの識別名：

```
CN=USERS-NNMi-Admin,OU=Groups,OU=Accounts,DC=example,DC=com
```

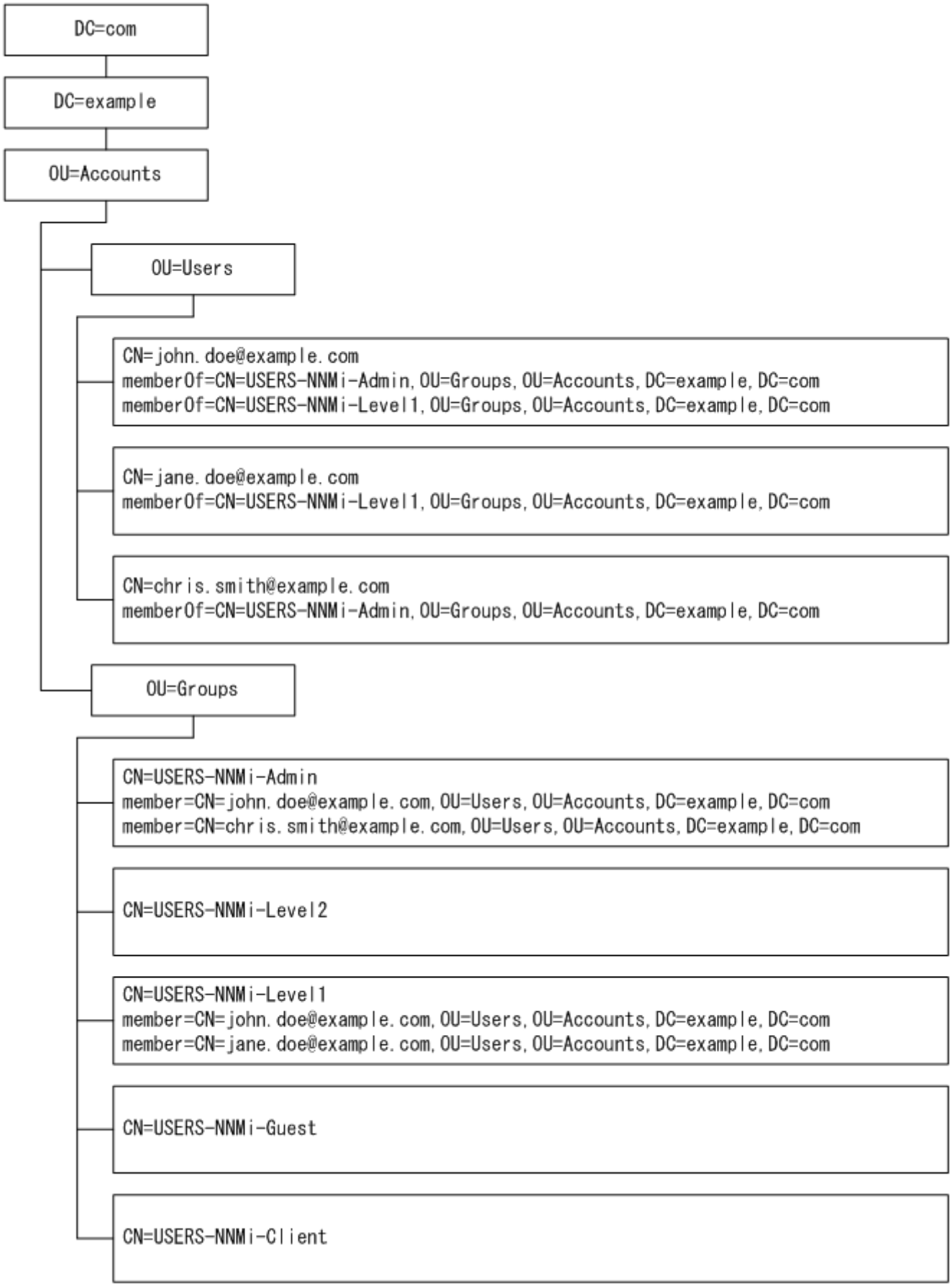
- ディレクトリサービスユーザー ID を保存するグループ属性：member

LDIF ファイルの引用例：

```
groups |USERS-NNMi-Admin
dn: CN=USERS-NNMi-Admin,OU=Groups,OU=Accounts,DC=example,DC=com
cn: USERS-NNMi-Admin
description: Group of users for NNMi administration.
member: CN=john.doe@example.com,OU=Users,OU=Accounts,
  DC=example,DC=com
member: CN=chris.smith@example.com,OU=Users,OU=Accounts,
  DC=example,DC=com
```

次の図に，このディレクトリサービスドメインの例を示します。

図 12-4　Active Directory のドメイン例



ほかのディレクトリサービスの情報構造例

この例での関心の対象は次の項目です。

- ユーザー John Doe の識別名：

```
uid=john.doe@example.com,ou=People,o=example.com
```

- USERS-NNMi-Admin グループの識別名：

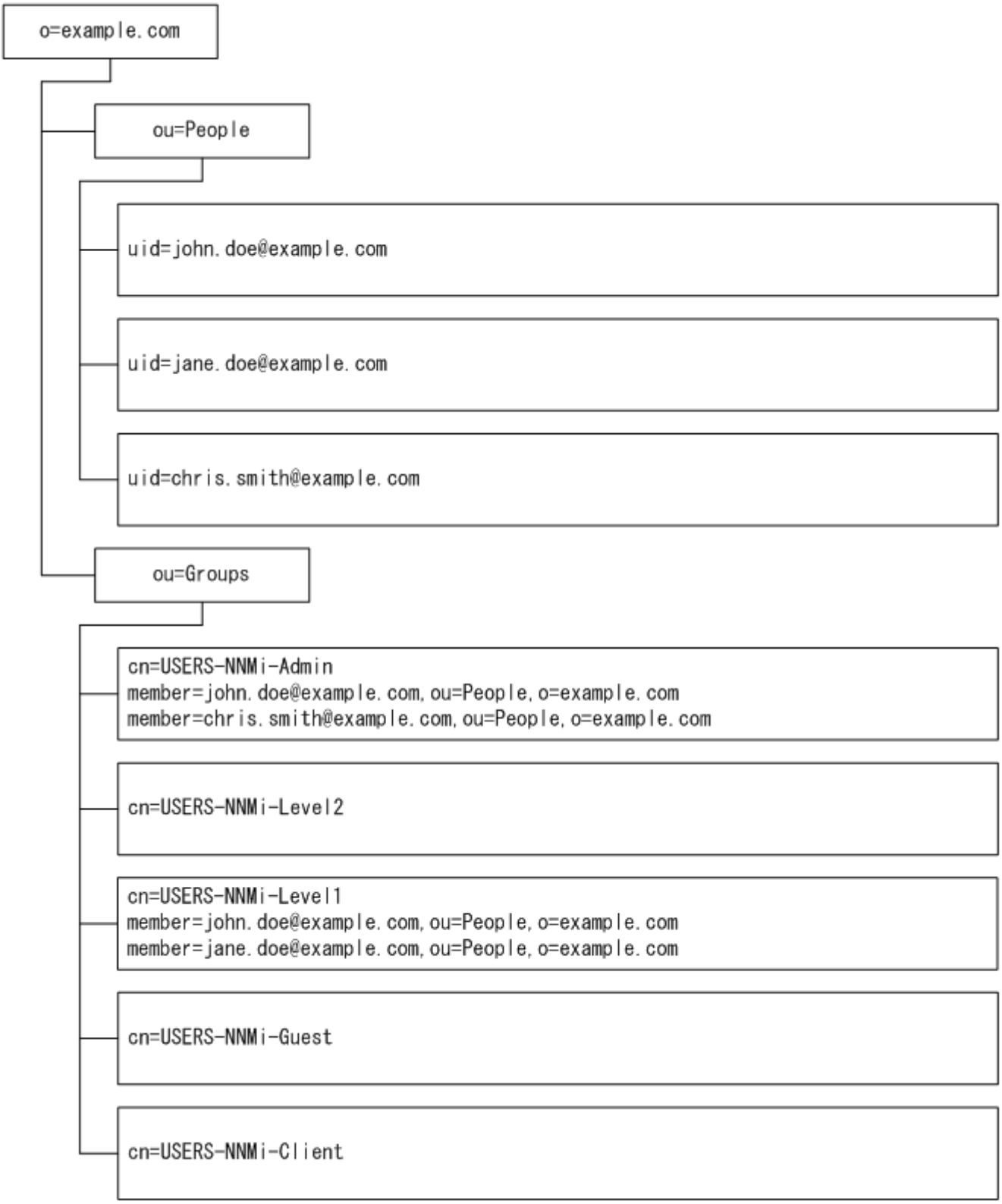
```
cn=USERS-NNMi-Admin,ou=Groups,o=example.com
```

- ディレクトリサービスユーザー ID を保存するグループ属性：member

LDIF ファイルの引用例：

```
groups |USERS-NNMi-Admin
dn: cn=USERS-NNMi-Admin,ou=Groups,o=example.com
cn: USERS-NNMi-Admin
description: Group of users for NNMi administration.
member: uid=john.doe@example.com,ou=People,o=example.com
member: uid=chris.smith@example.com,ou=People,o=example.com
```

図 12-5　ほかのディレクトリサービスのドメインの例



12.3.3　ディレクトリサービス管理者が所有する情報

「表 12-4　ディレクトリサービスからユーザー名およびパスワードを取得する場合の情報」および「表 12-5　ディレクトリサービスからグループメンバーシップを取得する場合の情報」に，LDAP を使用してディレクトリサービスにアクセスするように NNMi を設定する前に，ディレクトリサービス管理者から入手する情報を示します。

- ユーザー名とパスワードについてだけディレクトリサービスを使用する場合（「混合モード」の設定）は，「表 12-4　ディレクトリサービスからユーザー名およびパスワードを取得する場合の情報」の情報を収集します。
- すべての NNMi アクセス情報についてディレクトリサービスを使用する場合（「外部モード」の設定）は，「表 12-4　ディレクトリサービスからユーザー名およびパスワードを取得する場合の情報」および「表 12-5　ディレクトリサービスからグループメンバーシップを取得する場合の情報」の情報を収集します。

表 12-4　ディレクトリサービスからユーザー名およびパスワードを取得する場合の情報

情報	Active Directory 例	そのほかのディレクトリサービス例
ディレクトリサービスをホストするコンピュータの完全修飾名	directory_service_host.example.com	
LDAP 通信でディレクトリサービスが使用するポート	• 非 SSL 接続の場合は 389 • SSL 接続の場合は 636	
ディレクトリサービスでの SSL 接続情報	SSL 接続が必要な場合は，会社のトラストストアー証明書のコピーを取得し，「10.3.8　ディレクトリサービスへの SSL 接続を設定する」を参照します。	
ディレクトリサービスに保存される 1 つのユーザー名の識別名（ディレクトリサービスドメインを示す）	CN=john.doe@example.com, OU=Users,OU=Accounts, DC=example,DC=com	uid=john.doe@example.com, ou=People,o=example.com

表 12-5　ディレクトリサービスからグループメンバーシップを取得する場合の情報

情報	Active Directory 例	そのほかのディレクトリサービス例
ユーザーが割り当てられているグループを識別する識別名	memberOf ユーザー属性でグループを識別します。	• ou=Groups,o=example.com • cn=USERS-NNMi-*, ou=Groups,o=example.com
グループ内のユーザーを識別する方法	• CN=john.doe@example.com, OU=Users,OU=Accounts, DC=example,DC=com • CN=john.doe@example.com	• cn=john.doe@example.com, ou=People,o=example.com • cn=john.doe@example.com
ディレクトリサービスユーザー ID を保存するグループ属性	member	member
NNMi アクセスに適用するディレクトリサービスのグループの名前	• CN=USERS-NNMi-Admin, OU=Groups,OU=Accounts, DC=example,DC=com	• cn=USERS-NNMi-Admin, ou=Groups,o=example.com • cn=USERS-NNMi-Level2,

情報	Active Directory 例	そのほかのディレクトリサービス例
NNMi アクセスに適用するディレクトリサービスのグループの名前	• CN=USERS-NNMi-Level2, OU=Groups,OU=Accounts, DC=example,DC=com • CN=USERS-NNMi-Level1, OU=Groups,OU=Accounts, DC=example,DC=com • CN=USERS-NNMi-Client, OU=Groups,OU=Accounts, DC=example,DC=com • CN=USERS-NNMi-Guest, OU=Groups,OU=Accounts, DC=example,DC=com	ou=Groups,o=example.com • cn=USERS-NNMi-Level1, ou=Groups,o=example.com • cn=USERS-NNMi-Client, ou=Groups,o=example.com • cn=USERS-NNMi-Guest, ou=Groups,o=example.com

12.3.4 ユーザー識別

ユーザー識別は，「混合モード」および「外部モード」に適用されます。

ユーザー識別のための識別名は，1 人のユーザーをディレクトリサービスで特定するための完全に修飾する方法です。NNMi は，ユーザー識別名を LDAP 要求でディレクトリサービスに渡します。

LDAP 設定ファイルにおけるユーザー識別名は，nms-auth-config.xml ファイル内の<base>エレメントと<baseContextDN>エレメントを連結したものです。ディレクトリサービスによって返されたパスワードが，NNMi コンソールにユーザーが入力したサインインパスワードと一致する場合，ユーザーサインインが続行されます。

「混合モード」の場合は，次の情報が適用されます。

- NNMi コンソールアクセスの場合，NNMi は次の情報を検討し，できるだけ高い権限をユーザーに与えます。
  －LDAP 設定ファイルのdefaultRole パラメーターの値
  －NNMi コンソールで定義済みの NNMi ユーザーグループでの，このユーザーのメンバーシップ
- NNMi トポロジオブジェクトアクセスの場合，NNMi は，NNMi コンソールでこのユーザーが属する NNMi ユーザーグループのセキュリティグループマッピングに従ってアクセス権を与えます。

「外部モード」の場合は，次の情報が適用されます。

- NNMi コンソールアクセスの場合，NNMi は次の情報を基に，できるだけ高い権限をユーザーに与えます。
  －LDAP 設定ファイルのdefaultRole パラメーターの値
  －NNMi コンソールで定義済みの NNMi ユーザーグループにマッピングされている（**[ディレクトリサービス名]** フィールド）ディレクトリサービスグループでの，このユーザーのメンバーシップ

- NNMi トポロジオブジェクトアクセスの場合，NNMi は，このユーザーがディレクトリサービス（NNMi コンソールで NNMi ユーザーがマッピングされている）で属するグループのセキュリティグループマッピングに従ってアクセス権を与えます。

Active Directory でのユーザー識別例

nms-auth-config.xml ファイルの内容が<base>CN={0}</base><baseContextDN>OU=Users,OU=Accounts,DC=example,DC=com</baseContextDN>で，ユーザーが NNMi に john.doe としてサインインする場合，ディレクトリサービスに渡される文字列は次のとおりです。

CN=john.doe,OU=Users,OU=Accounts,DC=example,DC=com

そのほかのディレクトリサービスでのユーザー識別例

nms-auth-config.xml ファイルの内容が<base>uid={0}@example.com</base><baseContextDN>ou=People,o=example.com</baseContextDN>で，ユーザーが NNMi に john.doe としてサインインする場合，ディレクトリサービスに渡される文字列は次のとおりです。

uid=john.doe@example.com,ou=People,o=example.com

12.3.5 ユーザーグループ識別

ユーザーグループ識別は，「外部モード」の設定に適用されます。

NNMi は，NNMi ユーザーのユーザーグループを次のように判断します。

1. NNMi コンソールで設定されているすべてのユーザーグループの外部名の値をディレクトリサービスグループの名前と比較する。
2. ユーザーグループが一致する場合，NNMi ユーザーがディレクトリサービスのそのグループのメンバーであるかどうかを判断する。

NNMi コンソールで，短いテキスト文字列によって，NNMi コンソールアクセスを許可する，定義済みの NNMi ユーザーグループの一意の名前が識別されます。LDAP 設定ファイルの defaultRole および userRoleFilterList パラメーターも，このテキスト文字列を必要とします。次の表では，このグループの一意の名前を表示名にマッピングしています。

表 12-6 NNMi ユーザーグループ名のマッピング

NNMi コンソールの NNMi ロール名	NNMi 設定ファイルのユーザーグループの一意の名前およびテキスト文字列
管理者	admin
グローバルオペレーター	globalops
オペレータレベル 2	level2
オペレータレベル 1	level1
ゲスト	guest

NNMi コンソールの NNMi ロール名	NNMi 設定ファイルのユーザーグループの一意の名前およびテキスト文字列
Web サービスクライアント	client

NNMi グローバルオペレータユーザーグループ（globalops）では，すべてのトポロジオブジェクトだけにアクセス権が与えられます。ユーザーが NNMi コンソールにアクセスするには，ユーザーをほかのどれかのユーザーグループ（admin，level2，level1，またはguest）に割り当てる必要があります。

globalops ユーザーグループはデフォルトですべてのセキュリティグループにマッピングされるため，管理者はこのユーザーグループをセキュリティグループにマッピングしないようにする必要があります。

(1)　ディレクトリサービスからのユーザーグループ取得の設定（詳細な方法）

「12.2　ディレクトリサービスへのアクセスを設定する」の「12.2.5　タスク 5：(「外部モード」の設定だけ）ディレクトリサービスからのグループの取得を設定する」の説明にある簡単な方法では正常に機能しない場合は，次の手順を実行します。

1. 必要なユーザー情報をディレクトリサービス管理者から取得する。
2. 適切な手順を完了し，ディレクトリサービスでのグループ名およびグループメンバーの形式を確認する。
   - Active Directory の場合に LDAP ブラウザを使用する方法：以降の「(2)　ディレクトリサービスでグループおよびグループメンバーシップを識別する方法の判別（Active Directory の場合に LDAP ブラウザを使用する方法)」を参照してください。
   - ほかのディレクトリサービスの場合に LDAP ブラウザを使用する方法：以降の「(3)　ディレクトリサービスでグループおよびグループメンバーシップを識別する方法の判別（ほかのディレクトリサービスの場合に LDAP ブラウザを使用する方法)」を参照してください。
   - ほかのディレクトリサービスの場合に Web ブラウザを使用する方法：以降の「(4)　ディレクトリサービスでグループを識別する方法の判別（Web ブラウザを使用する方法)」を参照してください。
3. LDAP 設定ファイルを設定する。
   a. nms-auth-config.xml ファイルを任意のテキストエディタで開く。
   b. ディレクトリサービスでグループにユーザー名が保存されるときの方法とユーザー名が相関するように，role エレメントを設定する。
      実際のユーザー名を次の式のどちらかで置き換えます。
      - サインインのために入力されたユーザー名を意味する場合は{0}を使用します（たとえば，john.doe)。
      - ディレクトリサービスによって返された認証済みユーザーの識別名を意味する場合は，{1}を使用します（たとえば，uid=john.doe@example.com,ou=People,o=example.com)。
   c. ディレクトリサービスのドメインでグループレコードを保存する部分を roleContextDN エレメントに設定する。
      形式は，ディレクトリサービスの属性名と値のカンマ区切りリストです。
      例：

- Active Directory の場合：
  `CN=Users,DC=ldapserver,DC=mycompany,DC=com`
- その他の LDAP 技術の場合：
  `ou=Groups,o=example.com`

4.「12.2　ディレクトリサービスへのアクセスを設定する」の説明に従って設定をテストする。

(2)　ディレクトリサービスでグループおよびグループメンバーシップを識別する方法の判別（Active Directory の場合に LDAP ブラウザを使用する方法）

サードパーティの LDAP ブラウザで，次の手順を実行します。

1. ディレクトリサーバードメインの中でユーザー情報を保存する領域にナビゲートする。
2. NNMi にアクセスする必要があるユーザーを識別し，そのユーザーに関連づけられているグループの識別名の形式を調べる。
3. ディレクトリサーバードメインの中でグループ情報を保存する領域にナビゲートする。
4. NNMi ユーザーグループに対応するグループを識別して，グループに関連づけられているユーザーの名前の形式を調べる。

(3)　ディレクトリサービスでグループおよびグループメンバーシップを識別する方法の判別（ほかのディレクトリサービスの場合に LDAP ブラウザを使用する方法）

サードパーティの LDAP ブラウザで，次の手順を実行します。

1. ディレクトリサーバードメインの中でグループ情報を保存する領域にナビゲートする。
2. NNMi ユーザーグループに対応するグループを識別して，それらのグループの識別名の形式を調べる。
3. グループに関連づけられているユーザーの名前の形式も調べる。

(4)　ディレクトリサービスでグループを識別する方法の判別（Web ブラウザを使用する方法）

1. サポートされる Web ブラウザで，次の URL を入力する。

```
ldap://<directory_service_host>:<port>/<group_search_string>
```

- *<directory_service_host>*は，ディレクトリサービスをホストするコンピュータの完全修飾名です。
- *<port>*は，LDAP 通信でディレクトリサービスが使用するポートです。
- *<group_search_string>*は，ディレクトリサービスに保存されるグループ名の識別名です（例：`cn=USERS-NNMi-Admin,ou=Groups,o=example.com`）。

2. ディレクトリサービスのアクセステストの結果を評価する。
   - ディレクトリサービスに要求されたエントリが存在しないことを示すメッセージが表示された場合は，<*group_search_string*>の値を確認してから，手順 1.の操作を繰り返してください。
   - 該当するグループのリストが表示された場合，そのアクセス情報は正しいことになります。
3. グループのプロパティを調べ，そのグループに関連づけられているユーザーの名前の形式を判断する。

12.4 NNMi ユーザーグループを保存するディレクトリサービスの設定

NNMi ユーザーグループをディレクトリサービスに保存する場合（「外部モード」の設定）は，NNMi ユーザーグループ情報を使用してディレクトリサービスを設定する必要があります。原則として，ディレクトリサービスには適切なユーザーグループがすでに含まれています。含まれていない場合，ディレクトリサービス管理者は，特に NNMi ユーザーグループ割り当て用の新規ユーザーグループを作成できます。

ディレクトリサービスの設定およびメンテナンス手順は，特定のディレクトリサービスソフトウェアと企業のポリシーに応じて異なるため，ここではそれらの手順について説明していません。

12.5　ディレクトリサービス統合のトラブルシューティング

1. 次のコマンドを実行して，NNMi LDAP 設定を検証する。

```
nnmldap.ovpl -info
```

報告された設定が期待どおりの設定ではない場合は，LDAP 設定ファイルで設定を確認してください。

2. 次のコマンドを実行して，NNMi に LDAP 設定ファイルを再読み込みさせる。

```
nnmldap.ovpl -reload
```

3. 次のコマンドを実行して，ユーザーの設定をテストする。

```
nnmldap.ovpl -diagnose <NNMi_user>
```

*<NNMi_user>*は，ディレクトリサービスで定義した NNMi ユーザーのサインイン名で置き換えます。

コマンド出力を検討し，適切に応答します。

メモ

混合モードの場合，次のようなメッセージが出力されますが，混合モードでは LDAP グループを参照しないため，動作に問題ありません。次のメッセージは無視してください。

```
!!!!!!!!!!!!!!!!!!!!!!!! NOTE !!!!!!!!!!!!!!!!!!!!!!!!
!  このユーザー識別名のLDAPグループが見つかりません。
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


!!!!!!!!!!!!!!!!!!!!!!!! NOTE !!!!!!!!!!!!!!!!!!!!!!!!
!  LDAPの設定が誤っているようです。詳細は、上記を参照してください。
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

4. ディレクトリサービスに期待されるレコードが含まれていることを確認する。

Web ブラウザまたはサードパーティの LDAP ブラウザ（Apache Directory Studio に含まれる LDAP ブラウザなど）を使用して，ディレクトリサービスの情報を調べます。

ディレクトリサービスに対するクエリーの形式に関する詳細については，次のサイトの RFC 1959「An LDAP URL Format」を参照してください。

```
http://www.ietf.org/rfc/rfc1959.txt
```

5. %NnmDataDir%log¥nnm¥nnm.log（Windows）または/var/opt/OV/log/nnm/nnm.log（Linux）のログファイルを表示し，サインイン要求が正しいことを確認して，エラーが発生しているかどうかを判断する。

- 次の行のようなメッセージは，ディレクトリサービスで HTTPS 通信が必要であることを示しています。この場合は，「10.3.8　ディレクトリサービスへの SSL 接続を設定する」の説明に従って SSL を有効にします。

```
javax.naming.AuthenticationNotSupportedException : [LDAP:error code 13 - confidentiality
required]
```

- 次の行のようなメッセージは，ディレクトリサービスとのやり取り中にタイムアウトが発生したことを示します。この場合は，LDAP 設定ファイルのsearchTimeLimit の値を増やします。

```
javax.naming.TimeLimitExceededException : [LDAP: error code 3 - Timelimit Exceeded]
```

12.6 LDAP 設定ファイルリファレンス

12.6.1 nms-auth-config.xml ファイル

nms-auth-config.xml ファイルには，ディレクトリサービスと通信して，それに対する LDAP 照会を作成するための設定が XML 形式で保存されています。このセクションでは，LDAP 設定に関連するエレメントのみをリファレンス 用として提供します。

このファイルは次の場所にあります。

- Windows の場合：%NnmDataDir%nmsas¥NNM¥conf
- Linux の場合：$NnmDataDir/nmsas/NNM/conf

デフォルトでは，この場所にあるnms-auth-config.xml ファイルには LDAP 設定に必要な XML エレメントは含まれていません。

必要なすべての XML エレメントをこのファイルに手動で追加するには，このセクションの手順に従います。

NNMi によって，参照用のサンプルnms-auth-config.xml ファイルが別の場所に格納されます。

サンプルnms-auth-config.xml ファイルは次の場所にあります。

- Windows の場合：%NnmInstallDir%newconfig¥HPOvNnmAS¥nmsas¥conf
- Linux の場合：$NnmInstallDir/newconfig/HPOvNnmAS/nmsas/conf

ヒント

サンプルnms-auth-config.xml ファイルから<ldapLogin>エレメント全体をコピーし，必要な変更を加えることもできます。

nms-auth-config.xml ファイル（%NnmDataDir%nmsas¥NNM¥conf（Windows）または$NnmDataDir/nmsas/NNM/conf（Linux））の編集後，次のコマンドを実行して NNMi に LDAP 設定を再度読み込ませます。

- Windows の場合：

```
%NnmInstallDir%bin¥nnmldap.ovpl -reload
```

- Linux の場合：

```
$NnmInstallDir/bin/nnmldap.ovpl -reload
```

```
<ldapLogin>
<!-- これはLDAP認証をオン/オフするスイッチです。trueのときLDAPベースの認証が使用されます-->
  <enabled>true</enabled>
<!-- このエレメントにより、インシデントの割り当てが行えるユーザーを指定できます。-->
  <userRoleFilterList>admin guest level2 level1</userRoleFilterList>
```

```
<!-- <enabled>がtrueの場合、<configuration>エレメントを定義してLDAPパラメーターを指定します-
->
  <configuration>
<!-- ログオンを試みたユーザーとフィルター(オプション) を照合して、適切な設定が使用されている
か確認します。複数の設定が指定されている場合、これにより適用外のLDAPサーバーをスキップしてロ
グオン時間を短縮できます。-->
    <filter>
      <usernamePattern>.*@hpe¥.com</usernamePattern>
    </filter>
<!-- LDAPサーバーの検索を実行するときの時間制限-->
    <searchTimeLimit>30000</searchTimeLimit>
    <connectTimeLimit>10000</connectTimeLimit>
<!-- サーバーURLを定義-->
    <server>
      <hostname>ldaps://ldap.domain1.com</hostname>
      <secure>true</secure>
    </server>
<!---オプション。匿名アクセスをサポートしていないLDAPサーバーに接続するためのバインド資格証
明と暗号化パスワード。"nnmldap.ovpl -encrypt" を使用して暗号化パスワードを作成します。--->
    <bindCredential>
      <bindDN>someUser@some.com</bindDN>
      <bindCredential>someEncryptedPassword</bindCredential>
    </bindCredential>
<!-- このエレメントは、このLDAP設定でユーザーを検索するためのルールを定義します-->
    <users>
<!-- オプション。ログオンを試みたユーザーと照合されるフィルター。適用外のLDAP設定をスキップ
してログオン時間を短縮するために使用されます。これはJavaの正規表現です。-->
      <filter>
        <usernamePattern>.*some¥.com</usernamePattern>
      </filter>
<!-- オプション。NNMiコンソールに表示される表示名の式。-->
      <displayName>${sn},${givenName} (HPE)</displayName>
<!-- オプション。この設定で認証されるすべてのユーザーに付与されるデフォルトのロール-->
      <defaultRoles>
        <role>guest</role>
      </defaultRoles>
<!-- ユーザーアカウントを見つけるための検索設定。文字列中のパターン"{0}" は、ユーザーがログ
オン画面に入力したログオン名に置き換えられます。-->
      <userSearch>
        <base>uid={0}</base>
        <baseContextDN>ou=People,o=domain.com</baseContextDN>
      </userSearch>
    </users>
<!-- このLDAP設定でユーザーのロールまたはグループを検索するためのルールを定義します-->
    <roles>
<!-- オプション。この設定でロールを検索するための対象となるユーザーを定義するフィルター。こ
れはJavaの正規表現です。-->
      <filter><usernamePattern>x</usernamePattern></filter>
<!-- 認証されたユーザーDNが含まれているLDAPグループを見つけるための検索設定。ユーザーのDNが
含まれる箇所で文字列"{1}" を使用します。-->
      <roleSearch>
        <roleBase>member={1}</roleBase>
        <roleContextDN>ou=Groups,o=some.com</roleContextDN>
      </roleSearch>
      <roleSearch>
        <roleBase>GroupMember={1}</roleBase>
        <roleContextDN>CN=Groups,DC=mycompany,DC=com</roleContextDN>
      </roleSearch>
```

```
    </roles>
  </configuration>
</ldapLogin>
```

12.7 nms-auth-config.xml ファイルへの切り替え

11-10 から 11-50 以降へバージョンアップすると自動的にldap.properties ファイルの設定がnms-auth-config.xml ファイルへ移行されます。

重要

ldap.properties ファイルでdefaultRole を次のようにコメントアウトしている状態で，11-50 以降へバージョンアップした場合，バージョンアップ時に生成されたnms-auth-config.xml ファイルはそのままご利用いただけません。

```
#defaultRole=guest
```

nms-auth-config.xml ファイルの次の設定個所をコメントアウトするか削除して，nnmldap.ovpl -reload コマンドを実行し，LDAP 設定ファイルを再読み込みしてください。

```
<defaultRoles>
 <role/>
</defaultRoles>
```

13 NAT 環境の重複 IP アドレスの管理

NAT（ネットワークアドレス変換）では，多数のローカルネットワークを 1 つの動的な外部（パブリック）IP アドレスを使用してグローバルインターネットに接続することで IP アドレスを節約できます。また，内部アドレスを外部ネットワークから隠ぺいすることで，プライベートネットワークのセキュリティが強化できます。この章では，NNMi で使用する NAT の設定および重複する IP アドレスの設定について説明します。

13.1 NAT とは

通常，ネットワークアドレス変換は，ローカルネットワークを外部インターネットと相互接続するために使用します。このテクノロジは，より多くの IPv4 アドレスを求めるニーズの高まりに対応するソリューションとして開発されました。また，IP アドレスの特定範囲（RFC1918 を参照）は，内部専用として設計されていた（インターネット上でルーティングできない）ため，NAT のようなテクノロジを求める声が強くなっていました。

NAT では IP ヘッダー情報を変換します。パブリックネットワークを通過する必要がある IP パケットの内部アドレスを外部アドレスに置き換えます。NAT では，静的または動的な外部アドレスを使用することで内部アドレスを外部アドレスに変換します。

13.2 NAT の利点

NAT には，次のような利点があります。

- 多数のホストが 1 つの動的な外部 IP アドレスを使用してグローバルインターネットに接続するため，IP アドレス空間を節約できる
- プライベート IP アドレスを再利用できる
- 内部アドレスを外部ネットワークから隠ぺいすることで，プライベートネットワークのセキュリティが強化される

13.3 サポートされる NAT タイプ

NNMi では，次のタイプの NAT プロトコルがサポートされます。

- 静的 NAT：

  内部 IP アドレスが，常に同じ外部 IP アドレスにマップされる NAT タイプ（各ノードは静的な内部／外部アドレスペアを持つ）。このタイプでは，Web サーバーなどの内部ホストに未登録（プライベート）IP アドレスを割り当てたまま，インターネット上で到達可能な状態にすることができます。

- 動的 NAT：

  外部アドレスと内部アドレスのバインドをセッションごとに変更できる NAT スキーム。この NAT スキームでは，利用可能な登録済み（パブリック）IP アドレスのプールから得られるパブリック IP アドレスに内部 IP アドレスがマップされます。通常，ネットワーク内の NAT ルーターで登録済み IP アドレスのテーブルが保持されています。内部 IP アドレスからインターネットへのアクセスが要求されると，別の内部 IP アドレスで現在使用されていない IP アドレスがルーターによってテーブルから選択されます。

- 動的ポートアドレス変換（動的 PAT）（ネットワークアドレスおよびポート変換（NAPT）とも呼ばれる）：

  このタイプの NAT では，IP アドレスだけでなくポート番号も変換されます。アドレスとポート番号を変換することで，複数の内部アドレスが 1 つの外部アドレスを使用してインターネット上で同時に通信できるようになります。

13.4　NNMi に NAT を実装する方法

NNMi では，テナント／IP アドレスのペアを使用して各ノードを識別することによって，NAT 環境を管理します。NNMi 管理者は，NAT アドレスドメインごとにテナント定義を作成します。テナントによって，ノードの論理グループが識別されます。例えば，インターネットプロバイダーのネットワークに，プライベート IP アドレスを実装した顧客が複数存在するとします。インターネットプロバイダーは，NNMi 内で各顧客のノードを，個々の顧客を識別する特定のテナント名に割り当てることができます。そのテナントの論理グループ内では，次のようになります。

- NNMi 管理者は，検出シードを使用して，テナント／IP アドレスのペアを使用するテナントメンバーのノードを識別します。
- サブネット接続ルールは，各テナントのノードグループ内で独立して適用されます。
- ルーター冗長グループは，ほかのテナントノードグループから独立し，各テナント内でモニタリングされます。
- NNMi は，各テナントのノードグループ内，および定義済みのそのテナントのノードとデフォルトテナントに割り当てられたノード間だけで L2 接続を検出します。
- 複数の NAT ドメイン（NAT ゲートウェイルーターなど）と相互接続するインフラストラクチャーデバイスは，すべてデフォルトテナントに割り当てます。これによって，ワークグループ（および顧客）が確認する必要があるレイヤー 2 接続が NNMi に表示されるようになります。
- NNMi ユーザーが表示できるテナント数は，セキュリティグループによって決まります。割り当てられたセキュリティグループには，複数のテナントのノードを含めることができます。詳細については，「14.　NNMi のセキュリティおよびマルチテナント」を参照してください。

> **メモ**
>
> ネットワーク管理環境のすべての NAT ドメインで，ドメインネームシステム（DNS）名が重複しないようにすることを推奨します。

使用している NAT プロトコルによって，NNMi の実装方法や要件が異なる場合があります。例えば，動的 NAT または動的 PAT を使用している場合，追加のハードウェアおよびライセンスが必要になります。NAT プロトコルのタイプに基づいて，適切な節を参照してください。

- 「13.5　静的 NAT の考慮事項」
- 「13.6　動的 NAT および動的 PAT の考慮事項」

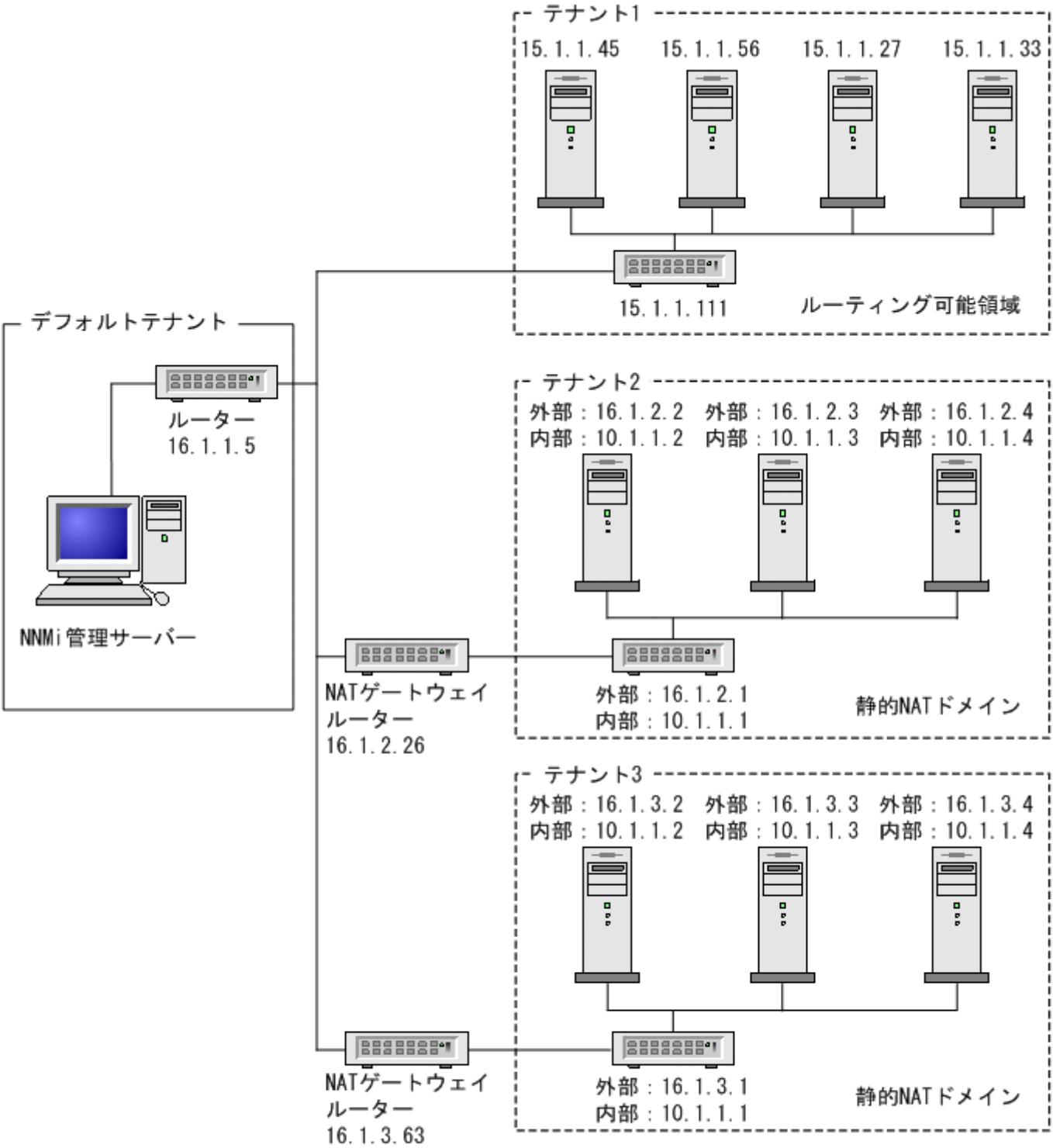
詳細については，「13.6.6　ネットワークアドレス変換（NAT）環境での NNMi の配備」および「13.6.7　状態とステータスの NNMi 計算」を参照してください。

13.5　静的 NAT の考慮事項

各インスタンスが一意のテナントで設定されていれば，1 つの NNMi 管理サーバーで任意の数の静的 NAT インスタンスを監視できます。テナントの詳細については，「14.　NNMi のセキュリティおよびマルチテナント」および NNMi ヘルプの「テナントを設定する」を参照してください。

静的 NAT の設定例として次の図を参照してください。

図 13-1　静的 NAT の設定例



（凡例）

外部：外部アドレス

内部：内部アドレス

デフォルトテナントに属するノードは，任意のテナントの任意のノードにレイヤー 2 接続できます。デフォルトテナント以外のテナント内のノードは，同じテナントかデフォルトテナント内のデバイスにしかレイヤー 2 接続できません。

サブネットはテナントに固有です（サブネットは複数のテナントにまたがらない）。このメリットは，同じサブネットを異なるテナントで使用できる点にあります。

ルーター冗長グループ（RRG）はテナントをまたがることができません。

複数の NAT ドメイン（NAT ゲートウェイなど）と相互接続するインフラストラクチャーデバイスは，すべてデフォルトテナントに割り当てます。これによって，ワークグループ（および顧客）が確認する必要があるレイヤー 2 接続が NNMi に表示されるようになります。

デフォルトのセキュリティグループ内のデバイスはすべてのビューで表示されます。デバイスへのアクセスを制御するには，該当するデバイスをデフォルトのセキュリティグループ以外のセキュリティグループに割り当てます。

13.5.1　静的 NAT のハードウェアとソフトウェアの要件

静的 NAT では，特別なハードウェアまたはソフトウェアの要件はありません。1 つの NNMi 管理サーバーで，NNMi，NNMi Advanced を使用する静的 NAT ドメインを幾つでも管理できます。

13.5.2　静的 NAT での通信

NNMi では，使用可能な重複するアドレスマッピングを自動的に使用して静的 NAT 通信用のテナント／外部 IP アドレスのペアを識別することによって，静的 NAT ファイアウォールを通して正常な通信が行われます。この利点については，「13.7　重複する IP アドレスマッピング」を参照してください。

(1)　静的 NAT 環境での管理アドレスの ICMP ポーリングの管理

NAT 環境では，ファイアウォールによって，NNMi がノードの IP アドレス（プライベート IP アドレス）を使用して NAT ノードとのやり取りがブロックされます。これを解決するには，NAT アドレス（パブリック IP アドレス）を使用して NNMi と通信します。

NAT 環境では，ノードの管理アドレスが，ノードでホストされる IP アドレスと異なることがあります。NNMi が NAT 環境でノードを検出できるようにするには，NAT アドレスを検出シードとして NNMi に追加する必要があります。NNMi は，この NAT アドレスがノードの `ipAddressTable` に存在しなくても，それを通信に使用します。

NNMi はこの機能を提供することで，誤ったノード停止中インシデントの生成を回避し，根本原因分析をより正確にします。

(2) NAT 環境での管理アドレスの ICMP ポーリングの概要

(a) NAT 環境の管理アドレスの ICMP ポーリング

NAT 環境がある場合，この設定を無効にしないことをお勧めします。

管理アドレスの ICMP ポーリングが無効になっている場合，有効にするには，次の手順を実行します。

1. ワークスペースのナビゲーションパネルで，［設定］ワークスペースを選択して［モニタリング］フォルダーを展開し，［モニタリングの設定］を選択して［デフォルト設定］タブを探す。
2. ［ICMP 障害モニタリング］セクションの［管理アドレスポーリングを有効にする］を有効にする。
   NNMi ヘルプの「モニタリングのデフォルト設定」を参照してください。
   SNMP エージェントに対して［アクション］＞［モニタリングの設定］を実行したあとに NNMi が表示する情報を確認します。表示される情報に，NNMi が管理アドレスのポーリングを有効にしているかどうかが示されます。

ICMP 管理アドレスポーリングが有効になっていると，NNMi が次のように変更されます。

- ［管理アドレス ICMP の状態］フィールドが次のフォームおよびテーブルビューに表示されます。
  - ［ノード］フォーム
  - ［SNMP エージェント］フォーム
  - ［SNMP エージェント］テーブルビュー
- NNMi は，管理アドレス ICMP 状態の表示場所と SNMP エージェントステータスの判断方法を変更します。

管理アドレス ICMP および IP アドレスの状態ポーリングアクションを次の表に示します。NNMi は，ICMP 管理アドレスポーリング設定および ICMP 障害ポーリング設定に対応して，これらのアクションを実行します。

表 13-1　ICMP 設定および結果の状態ポーリング

ICMP 管理アドレスポーリング	ICMP 障害ポーリング	管理 ICMP アドレス状態	IP アドレス状態
有効※	無効※	ポーリング※	ポーリングなし※
有効	有効	ポーリング	ポーリング
無効	無効	ポーリングなし	ポーリングなし
無効	有効	ポーリングなし	ポーリング

注※　デフォルトの設定

SNMP エージェントと管理アドレス ICMP の応答のために APA が判断する SNMP エージェントステータス，および生成されるインシデントの変化を次の表に示します。APA は，管理アドレスの ICMP ポー

リングで，結論とインシデントの生成時に，管理アドレス ICMP 応答と SNMP エージェント応答を考慮します。

表 13-2　SNMP エージェントステータスの判断および生成されるインシデント

SNMP エージェント応答	管理アドレス ICMP 応答	SNMP エージェントステータス	生成されるインシデント
応答	応答	正常域	なし
応答	無応答	警戒域	そのほかのネットワークの問題で，生成されるインシデントは次のとおりです。 • なし • AddressNotResponding
無応答	応答	危険域	SNMPAgentNotResponding
無応答	無応答	危険域	そのほかのネットワークの問題で，生成されるインシデントは次のとおりです。 • なし • NodeDown

13.5.3　検出と静的 NAT

NNMi 管理者は，ネットワーク管理環境内の各静的 NAT ドメインを識別するために，テナント定義を作成する必要があります。スパイラル検出では，NNMi が各ノードを検出して監視する前に各ノードを識別するための検出シード（テナントとアドレスのペア）が必要です。

NNMi 管理者は，静的 NAT ドメインのノードごとに検出シードを作成する必要があります。検出シードでは，ノードごとに次の情報を指定します。

- 外部 IP アドレス（外部/内部 IP アドレスペアのパブリックアドレス）
- テナント名

詳細については，NNMi ヘルプを参照してください。

重要

検出シードを静的 NAT 環境内に追加する場合（nnmloadseeds.ovpl コマンドまたは NNMi コンソールを使用），必ずノードの外部（パブリック）IP アドレスを使用してください。詳細については，nnmloadseeds.ovpl リファレンスページを参照してください。

ドメインネームシステム（DNS）名が重複しないようにすることをお勧めします。

13.5.4　静的 NAT のモニタリングの設定

ネットワーク環境によって，NNMi 管理者は ICMP 障害モニタリングの設定を使用するかどうかを選択できます。「13.6.7　状態とステータスの NNMi 計算」も参照してください。

- ［モニタリングの設定］＞［ノードの設定］タブ

  ノードグループのモニタリングを設定します。［ICMP 障害モニタリング］セクションで選択します。詳細については，NNMi オンラインヘルプを参照してください。

  - 管理アドレスポーリング（デフォルトで有効な，強く推奨される機能）
  - IP アドレス障害ポーリング（省略可能）

- ［モニタリングの設定］＞［デフォルト設定］タブ

  ［ICMP 障害モニタリング］セクションで選択します。詳細については，NNMi オンラインヘルプを参照してください。

> **重要**
>
> ネットワーク環境に動的 NAT ドメインも設定されている場合，動的 NAT ドメインとは異なる設定が静的 NAT ドメインで必要になることがあるため，デフォルト設定が適切でない可能性があります。

13.5.5　トラップと静的 NAT

NNMi 管理サーバーで NAT ゲートウェイの背後にあるノードから SNMP トラップを受信するには，管理対象ノードを変更する必要があります。この項では，SNMPv2c と SNMPv1 の 2 種類の SNMP トラップについて説明します。

NNMi では，受信した各トラップのソースアドレスを一義的に解決する必要があります。

(1)　SNMPv2c トラップ

次の図に，SNMPv2c トラップの形式を示します。この図の上部のセクションは IP ヘッダー，下部のセクションは SNMP トラップの`Protocol Data Unit (PDU)`で構成されています。

SNMPv2cトラップの形式

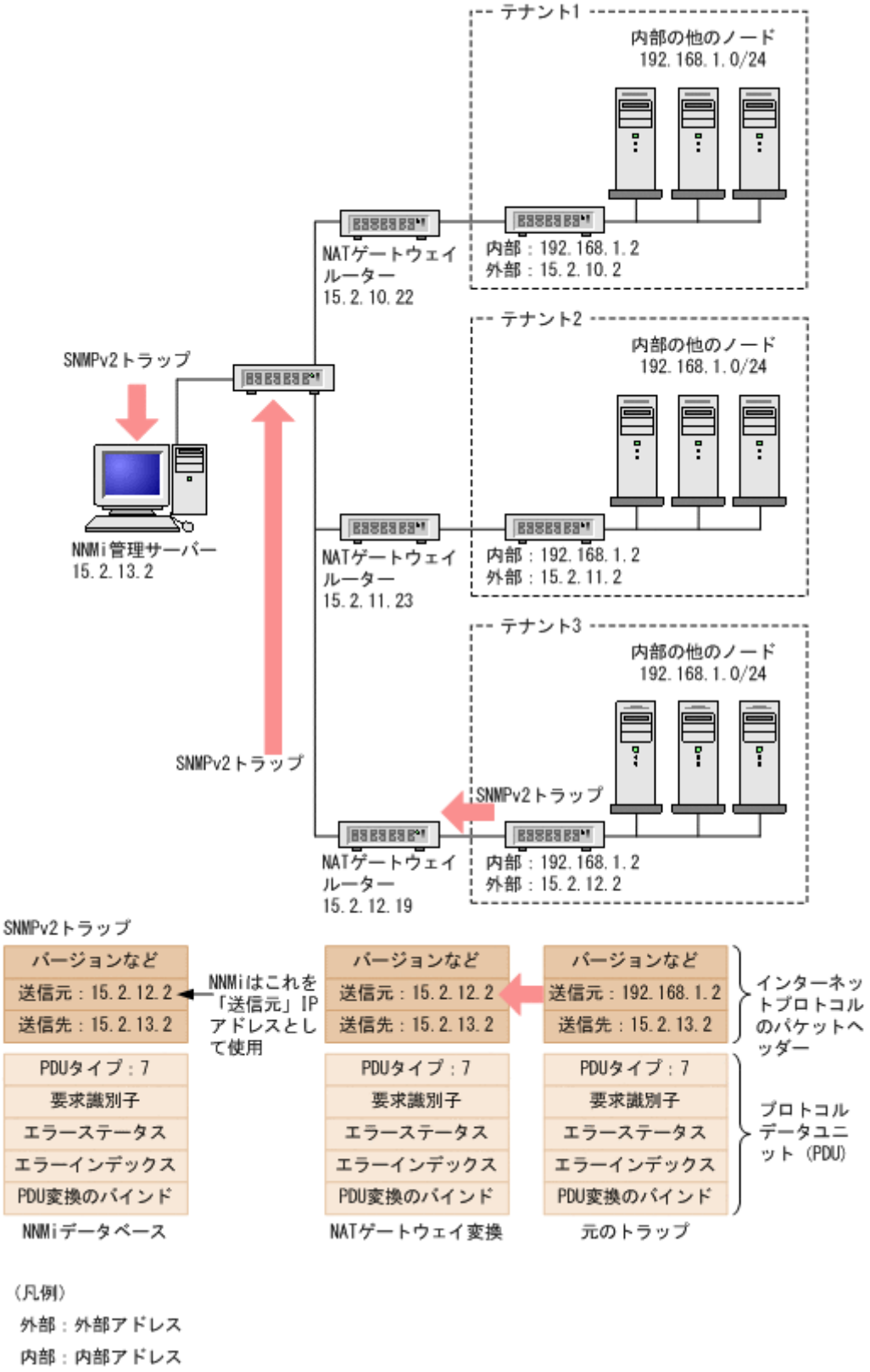
バージョンおよびその他の情報
ソースアドレス
デスティネーションアドレス
PDUタイプ: 7
要求識別子
エラーステータス
エラーインデックス
PDU変数のバインド

SNMPv2c トラップの PDU には，エージェントアドレスフィールドがありません。そのため，トラップの唯一のソースフィールドが IP パケットヘッダー内に存在します。ソースフィールドは，NAT ルーターによって適切に変換されます。

ソースノードのプライベート内部 IP アドレスに関連づけられているインタフェースで，NAT ルーターの背後にあるデバイスのすべてのトラップのソースが明らかになっていることを確認します。これで，NAT ゲートウェイがトラップを適切なパブリックアドレスに変換できます。

次の図に，NAT ゲートウェイからの適切な変換の例を示します。NAT ゲートウェイによって，192.168.1.2 のソースアドレスで始まるトラップのアドレスが 15.2.12.2 に適切に変換されます。次に，NNMi 管理サーバーによってこのアドレスが適切に解決されます。

図 13-2　SNMPv2c の例



(2) SNMPv1 トラップ

SNMPv1 トラップの場合，SNMP トラップの PDU 内にエージェントアドレスが組み込まれています。次の図に，SNMPv1 トラップの形式を示します。上部のセクションは IP ヘッダー，下部のセクションは SNMP トラップの PDU で構成されています。

SNMPv1 トラップの形式

バージョンおよびその他の情報
ソースアドレス
デスティネーションアドレス
PDUタイプ: 4
エンタープライズ
エージェントアドレス
汎用トラップコード
固有トラップコード
タイムスタンプ
PDU変数のバインド

エージェントアドレスはヘッダーではなく PDU に組み込まれているため，通常，この値は NAT ルーターによって変換されません。ヘッダーのアドレスを認識して，ペイロードのエージェントアドレスを無視するように NNMi を設定するには，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-jboss.properties`
   - Linux：`$NNM_PROPS/nms-jboss.properties`
2. 次の行を探す。

```
#!com.hp.nnm.trapd.useUdpHeaderIpAddress=false
```

3. 次のように値を true に変更して#!文字を削除する。

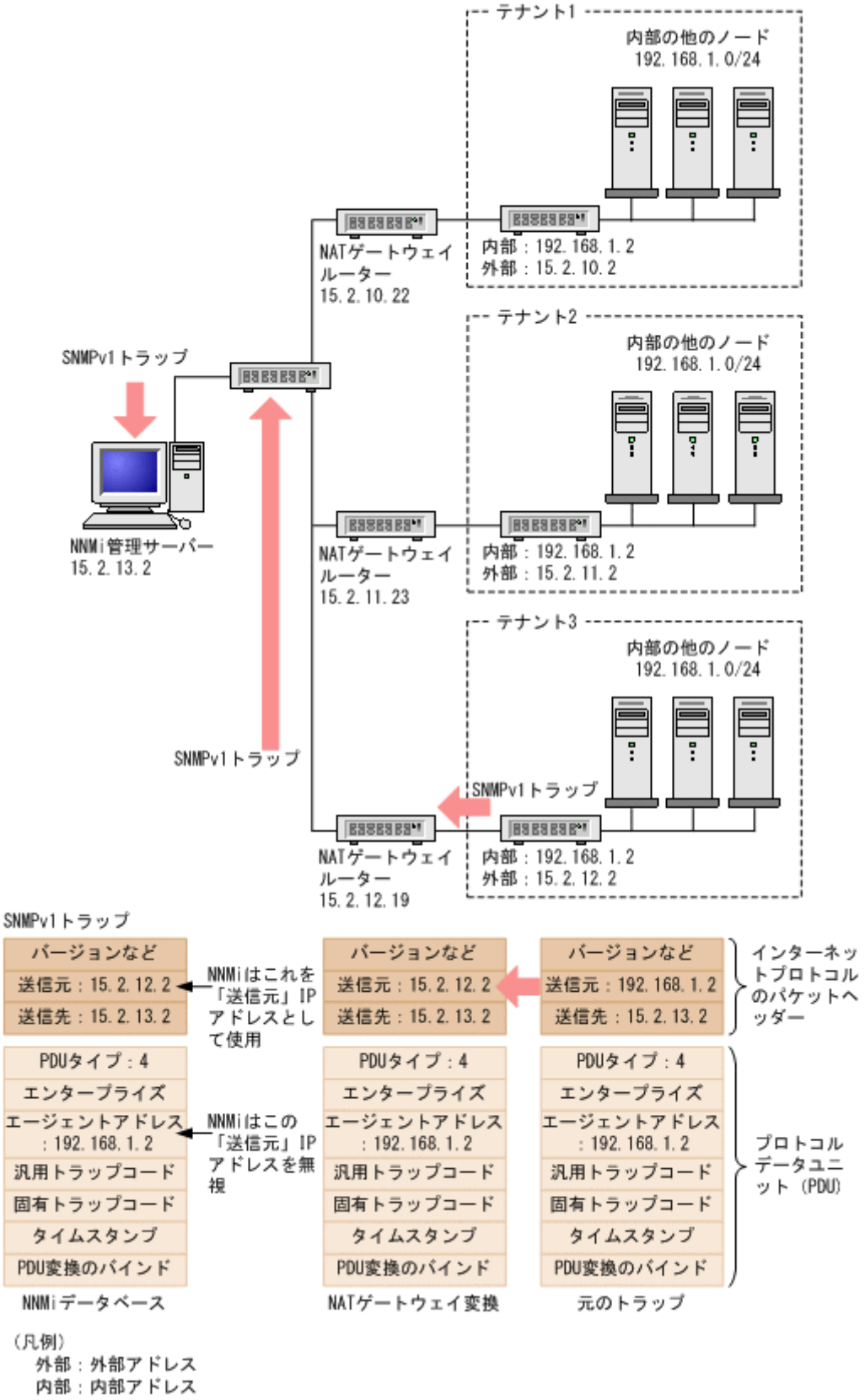
```
com.hp.nnm.trapd.useUdpHeaderIpAddress=true
```

4. 変更を保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

次の図に，競合するエージェントアドレスフィールドが NNMi で無視される SNMPv1 トラップの例を示します。

図 13-3　SNMPv1 の例



NNMi では，関連する次のカスタムインシデント属性（CIA）が提供されます。

- cia.agentAddress－トラップを生成した SNMP エージェントの SNMPv1 トラップデータに保存される IP アドレス。
- cia.internalAddress－静的 NAT がネットワーク管理ドメインに含まれている場合，NNMi 管理者は，選択したインシデントのソースノードの外部管理アドレスにマップされる内部 IP アドレスを表示するようにこの属性を設定できます。

[重複する IP アドレスマッピング] フォームを使用して，この内部アドレス（プライベートアドレス）に外部管理 IP アドレス（パブリックアドレス）をマップする必要があります。詳細については，NNMi ヘルプを参照してください。

13.5.6　サブネットと静的 NAT

サブネットおよび NAT に関しては，次の点に注意してください。

- サブネットはテナントに固有です（サブネットは複数のテナントにまたがらない）。このメリットは，同じサブネットを異なるテナントで使用できる点にあります。
- サブネットフィルタではテナントとアドレスのペアが使用されます。
- サブネット接続ルールを設定する場合，そのルールはすべてのテナントに適用されます。サブネットのメンバーは，すべてのテナントで一意である必要があります（各ノードは 1 つのテナントにだけ割り当てられます）。サブネット接続ルールで，デフォルトテナントと別のテナント間にリンクを確立できます。ただし，2 つのテナント間のリンクは，どちらかのテナントがデフォルトテナントである場合にだけ使用できます。

13.5.7　グローバルネットワーク管理と静的 NAT

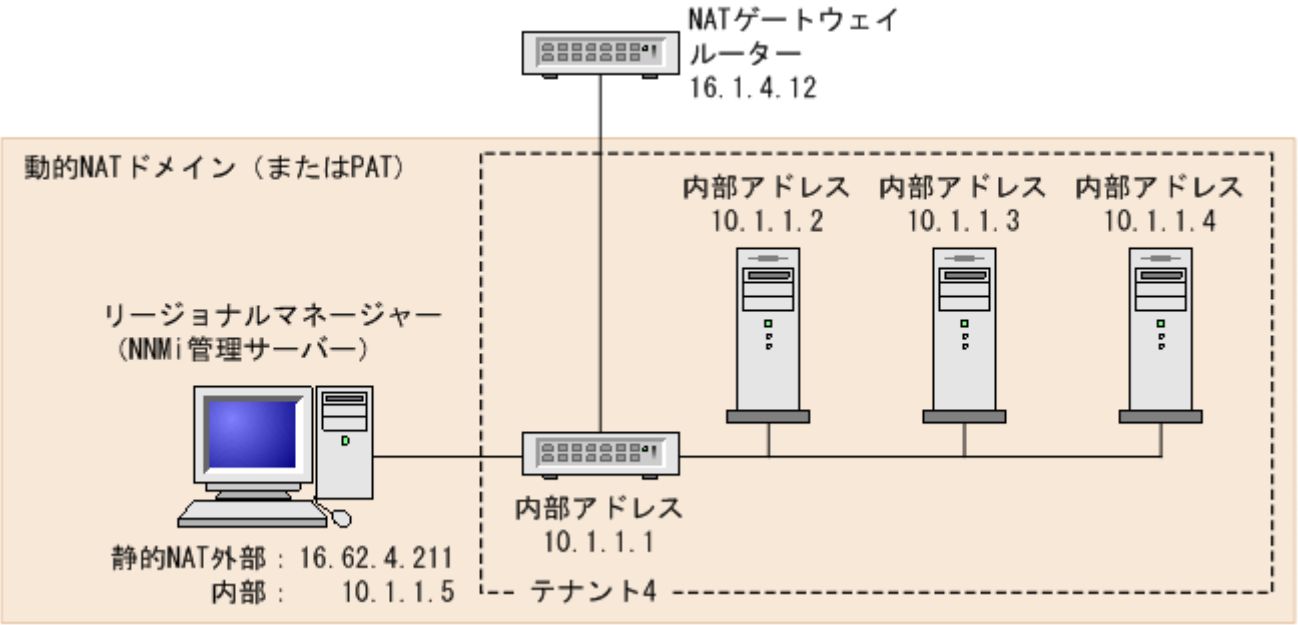
リージョナルマネージャーごとに，少なくとも 1 つの静的またはルーティング可能（非変換）アドレスがある必要があります。これによって，NNMi 管理サーバーが相互に通信ができ，通信を隠ぺいしてセキュリティを確保できます。グローバルネットワーク管理の詳細については，「15.　グローバルネットワーク管理」を参照してください。

13.6　動的 NAT および動的 PAT の考慮事項

1 つの NNMi 管理サーバーで 1 つの動的 NAT ドメインまたは動的 PAT ドメインを管理できます。このドメイン内にあるすべてのノードは一意の同じテナントに属している必要があります。NNMi 管理サーバーは，リージョナルマネージャーとしてグローバルネットワーク管理環境に参加している必要があります。動的 NAT の設定例として次の図を参照してください。

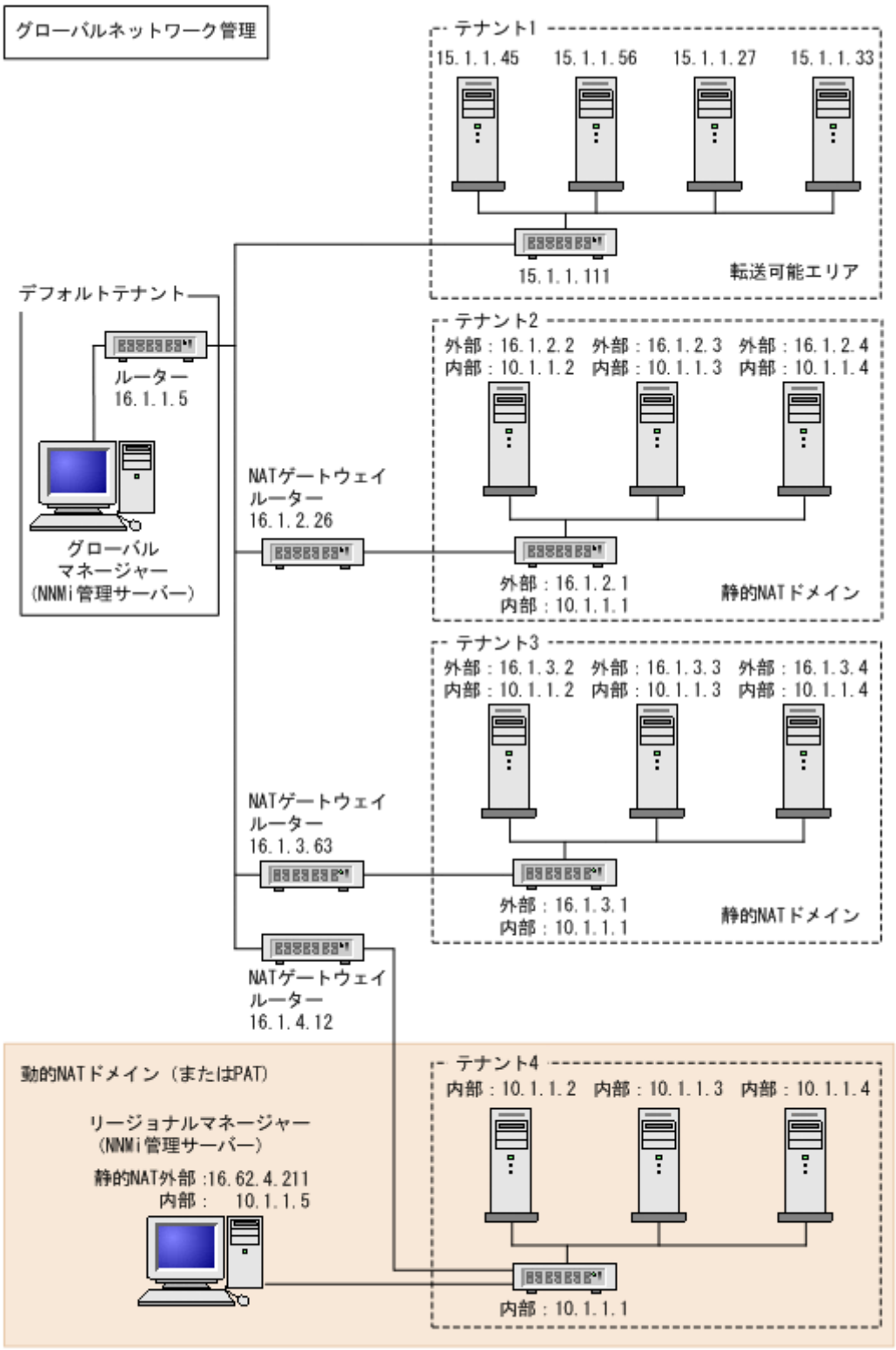
リージョナルマネージャーが NAT ファイアウォールの背後にある場合，その外部（パブリック）アドレスは静的アドレスである必要があります。

図 13-4　動的 NAT の設定例



複数の動的 NAT ドメイン，および動的 PAT ドメインを監視するには，NNMi のグローバルネットワーク管理機能を使用します。テナントは，NNMi グローバルネットワーク管理設定全体で一意である必要があります。NAT 環境内のグローバルネットワーク管理設定の例として次の図を参照してください。

図 13-5　NAT 環境内のグローバルネットワーク管理設定の例



（凡例）
外部：外部アドレス
内部：内部アドレス

デフォルトテナントに属するデバイスは，任意のテナントの任意のデバイスにレイヤー 2 接続できます。デフォルトテナント以外のテナント内のデバイスは，同じテナントかデフォルトテナント内のデバイスにしかレイヤー 2 接続できません。

複数の NAT ドメイン（NAT ゲートウェイなど）と相互接続するインフラストラクチャーデバイスは，すべてデフォルトテナントに割り当てます。これによって，ワークグループ（および顧客）が確認する必要があるレイヤー 2 接続が NNMi に表示されるようになります。

デフォルトのセキュリティグループ内のデバイスはすべてのビューで表示されます。デバイスへのアクセスを制御するには，該当するデバイスをデフォルトのセキュリティグループ以外のセキュリティグループに割り当てます。

グローバルネットワーク管理の詳細については，「15.　グローバルネットワーク管理」を参照してください。テナントの設定の詳細については，NNMi ヘルプの「テナントを設定する」を参照してください。

13.6.1　動的 NAT および動的 PAT のハードウェアとソフトウェアの要件

動的 NAT および動的 PAT 環境では，NNMi Advanced が必要になります。動的 NAT または動的 PAT で設定されたアドレスドメインごとに NNMi リージョナルマネージャーが必要です。

13.6.2　検出と動的 NAT および動的 PAT

NNMi では，テナントを使用して重複アドレスドメインを含むネットワークに対応します。重複アドレスドメインは，ネットワーク管理ドメインの動的 NAT または動的 PAT 領域内に存在することがあります。そのようなネットワークの場合，重複アドレスドメインを異なるテナントに配置します（これはシード検出を使用して行います）。詳細については，NNMi ヘルプを参照してください。

動的 NAT または動的 PAT 環境内に検出シードを追加する場合（`nnmloadseeds.ovpl` コマンドまたは NNMi コンソールを使用），必ずノードの内部 IP アドレスを使用してください。

詳細については，`nnmloadseeds.ovpl` リファレンスページ，または NNMi ヘルプを参照してください。

13.6.3　動的 NAT のモニタリングの設定

ネットワーク環境によって，NNMi 管理者は ICMP 障害モニタリングの設定を使用するかどうかを選択できます。「13.6.7　状態とステータスの NNMi 計算」も参照してください。

- ［モニタリングの設定］＞［ノードの設定］タブ

  ノードグループのモニタリングを設定します。［ICMP 障害モニタリング］セクションで選択します。詳細については，NNMi オンラインヘルプを参照してください。

  - 管理アドレスポーリング（デフォルトで有効な，強く推奨される機能）

- IP アドレス障害ポーリング（省略可能）

- ［モニタリングの設定］＞［デフォルト設定］タブ

  ［ICMP 障害モニタリング］セクションで選択します。詳細については，NNMi オンラインヘルプを参照してください。

> **重要**
>
> ネットワーク環境に静的 NAT ドメインも設定されている場合，動的 NAT ドメインとは異なる設定が静的 NAT ドメインで必要になることがあるため，デフォルト設定が適切でない可能性があります。

13.6.4　サブネットと動的 NAT および動的 PAT

サブネット，動的 NAT および動的 PAT に関しては，次の点に注意してください。

- サブネットはテナントに固有です（サブネットは複数のテナントにまたがらない）。このメリットは，同じサブネットを異なるテナントで使用できる点にあります。
- サブネットフィルタではテナントとアドレスのペアが使用されます。
- サブネット接続ルールを設定する場合，そのルールはすべてのテナントに適用されます。サブネットのメンバーは，すべてのテナントで一意である必要があります（各ノードは 1 つのテナントにだけ割り当てられます）。サブネット接続ルールで，デフォルトテナントと別のテナント間にリンクを確立できます。ただし，2 つのテナント間のリンクは，どちらかのテナントがデフォルトテナントである場合にだけ使用できます。

13.6.5　グローバルネットワーク管理と動的 NAT および動的 PAT

リージョナルマネージャーごとに，少なくとも 1 つの静的またはルーティング可能（非変換）アドレスがある必要があります。これによって，NNMi 管理サーバーが相互に通信ができ，通信を隠ぺいしてセキュリティを確保できます。

リージョナルマネージャーが NAT ファイアウォールの背後にある場合，その外部アドレスは静的アドレスである必要があります。

グローバルネットワーク管理の詳細については，「15.　グローバルネットワーク管理」を参照してください。NNMi ヘルプの「グローバルネットワーク管理のためのテナントのベストプラクティス」も参照してください。

13.6.6　ネットワークアドレス変換（NAT）環境での NNMi の配備

NAT 環境で NNMi を配備するには，次の手順を実行します。

1. ネットワーク管理環境の各 NAT ドメインのリストを特定して作成する。
2. 各 NAT ドメイン内で使用されるサポート対象 NAT のタイプを調べる。
3. 各 NAT ドメイン（NAT ドメインの内部 IP アドレス領域内外）に関して，必要に応じて各 NNMi 管理サーバーを配備する。

   次の特別な考慮事項を参照してください。

   - 「13.5　静的 NAT の考慮事項」
   - 「13.6　動的 NAT および動的 PAT の考慮事項」
4. NNMi の［設定］＞［検出］＞［テナント］ワークスペースを使用して，各 NAT ドメインで一意のテナント名を定義する。

   重要

   配備でグローバルネットワーク管理を使用している場合，この名前はすべての NNMi 管理サーバー（リージョナルマネージャーとグローバルマネージャー）で一意である必要があります。
5. NNMi でモニタリングする必要のある各 NAT ドメイン内のノードを決定する。
6. 静的 NAT ドメインのみ: 重複するアドレスマッピングを作成して，各ノードの割り当てられた NAT 外部/内部 IP アドレスのペアを識別する。

   重複するアドレスマッピングを作成する利点については，「13.7　重複する IP アドレスマッピング」を参照してください。

   次の情報を入力します。

   - テナント名
   - 外部 IP アドレス
   - 内部 IP アドレス

   NNMi の［設定］＞［検出］＞［重複するアドレスマッピング］ワークスペースまたは `nnmloadipmappings.ovpl` コマンドラインツールのどちらかを使用します。

   詳細については，NNMi オンラインヘルプを参照してください。
7. ネットワーク環境の NNMi 管理サーバーの配備先によっては，NNMi でノードの内部アドレスを使用する場合に，ファイアウォールによって NNMi と NAT ドメイン内のノードの通信がブロックされる可能性がある。そのため，［設定］＞［通信の設定］設定で，適切な［優先管理アドレス］設定（NAT の外部または内部 IP アドレス）を使用する。
8. ネットワーク環境の NAT の［モニタリングの設定］設定を確認する。

   - 「13.5.4　静的 NAT のモニタリングの設定」
   - 「13.6.3　動的 NAT のモニタリングの設定」

   ［モニタリングの設定］の詳細については，NNMi オンラインヘルプを参照してください。
9. 各ノードの検出シードを設定する。

重要

複数の NAT ドメイン（NAT ゲートウェイルーターなど）と相互接続するインフラストラクチャーデバイスは，すべてデフォルトテナントに割り当てます。

NNMi の［設定］＞［検出］＞［シード］ワークスペースまたは`loadseeds.ovpl` コマンドラインツールのどちらかを使用します。

- NNMi 管理サーバーが内部 IP アドレス領域内にある場合，内部 IP アドレスを使用して検出シードを設定します。
  - ・ホスト名/IP（内部 IP アドレスを使用）
  - ・テナント名
- NNMi 管理サーバーが内部 IP アドレス領域外にある場合，外部 IP アドレスを使用して検出シードを設定します。
  - ・ホスト名/IP（外部 IP アドレスを使用）
  - ・テナント名

詳細については，NNMi オンラインヘルプを参照してください。

10. NNMi 検出で，期待どおりノードが検出されることを確認する。

    検出されない場合，設定（上記）をダブルチェックします。

11. NNMi 設定がチームのニーズを満たしていることを確認する。
    - 各ノードのセキュリティグループの割り当てを微調整して，NNMi コンソールで各ノードを表示できるチームメンバー/顧客を制御します。NNMi の［設定］＞［セキュリティ］＞［セキュリティグループ］ワークスペースを使用します。
    - これらのノードに適用される［モニタリングの設定］設定を確認して，必要に応じて微調整します。NNMi の［設定］＞［モニタリング］＞［モニタリングの設定］ワークスペースを使用します。

12. NNMi マップにノード間の接続が期待どおりに表示されることを確認する。

    表示されない場合，次の作業を行います。

    - 接続に含まれる両方のノードのテナントの割り当て（デフォルトテナントまたはその他のテナント）が正しいことを確認します。
    - ［設定］＞［検出］＞［検出の設定］の［サブネット接続ルール］タブの設定が正しいことを確認します。
    - 自動的に検出されない接続を NNMi で強制的に追加するには，`nnmconnedit.ovpl` コマンドラインツールを使用します。詳細については，`nnmconnedit.ovpl` リファレンスページを参照してください。

13. 適切な NNMi 管理サーバーの IP アドレスが含まれるように各ノードの SNMP エージェントの SNMP トラップ転送ルールが設定されていることを確認する。

14. 静的 NAT ドメインのみ: NNMi の［重複するアドレスマッピング］の［内部アドレス］に関連づけられたインタフェースが，NNMi 管理サーバーに送信されるすべてのトラップのソースになるように，各静的 NAT ノードの SNMP エージェントを設定する。

15. ネットワーク環境に SNMPv1 が含まれている場合，NNMi 設定で必要な変更を適切に行う。
「13.5.5　トラップと静的 NAT」を参照してください。

13.6.7　状態とステータスの NNMi 計算

デフォルトの NNMi では，NAT 環境に存在するノードを含め，各ノードの管理アドレスの ICMP ポーリングが自動的に有効になります（[設定] ＞ [モニタリング] ＞ [モニタリングの設定]，[デフォルト設定] タブ，[ICMP 障害モニタリング] セクションの [管理アドレスポーリングを有効にする] 設定）。NAT 環境がある場合，この設定を無効にしないことをお勧めします。

重要

[インベントリ] ＞ [ノード] ビューでノードを選択し，[アクション] ＞ [設定の詳細] ＞ [モニタリングの設定] コマンドを使用します。表示される情報に，NNMi でこの管理アドレスポーリングが有効になっているかどうかが示されます。

管理アドレスポーリングが有効の場合は，[エージェント ICMP 状態] フィールドが，次の場所に表示されます。

- [ノード] フォーム
- [SNMP エージェント] フォーム
- [SNMP エージェント] テーブルビュー

次の表に，[ICMP 障害モニタリング] 設定に基づいて NNMi の動作がどのように変化するかを示します。

表 13-3　モニタリングの設定の内容および結果としての State Poller 動作

ICMP 障害モニタリングの設定		管理 ICMP アドレス状態 IP アドレス状態	
管理アドレスポーリングを有効にする	IP アドレス障害ポーリングを有効にする	エージェント ICMP 状態	IP アドレス状態
有効※	無効※	ポーリング	未ポーリング
有効	有効	ポーリング	ポーリング
無効	無効	未ポーリング	未ポーリング
無効	有効	未ポーリング	ポーリング

注※　デフォルトの設定

管理アドレスポーリングを有効にすると，結果の計算時とインシデントの生成時に，管理アドレスの ICMP 応答と SNMP エージェントの応答の両方が NNMi で考慮されます。

次の表に，ICMP 応答と SNMP 応答の組み合わせによって決定される，SNMP エージェントステータスの計算を示します。

表 13-4　SNMP エージェントステータスの判断

SNMP エージェントの応答	管理アドレスの ICMP 応答	結果としての SNMP エージェントステータス
応答	応答	正常域
応答	無応答	警戒域
無応答	応答	危険域
無応答	無応答	危険域

13.7　重複する IP アドレスマッピング

ネットワーク管理環境に重複アドレスドメインが含まれている場合，一意のテナントとして各ドメインを設定する必要があります。詳細については，NNMi ヘルプの「テナントを設定する」および「14.　NNMi のセキュリティおよびマルチテナント」を参照してください。

静的 NAT がネットワーク管理ドメインに含まれていて，NNMi 管理サーバーが静的 NAT ドメイン外にある場合，識別されたテナント／NAT 内部 IP アドレス（プライベート IPv4 アドレスなど）ペアの［IP アドレス］フォームの［マップされたアドレス］属性に NAT 外部 IP アドレス（パブリックアドレス）が表示されるように NNMi を設定できます。

動的 NAT および動的 PAT を使用しているネットワーク管理ドメインの領域に対して NNMi を設定している場合，［重複する IP アドレスマッピング］フォームは使用しないでください。「13.6　動的 NAT および動的 PAT の考慮事項」を参照してください。

ネットワークドメインの静的 NAT 設定は，パブリック IP アドレス，プライベート IP アドレスまたはその両方に適用されることがあります。

識別されたテナントと NAT 内部 IP アドレスペアの［IP アドレス］フォームの［マップされたアドレス］属性に静的 NAT 外部 IP アドレスが表示されるように NNMi を設定するには，次のどちらかを実行します。

- NNMi コンソールで，［重複する IP アドレスマッピング］フォームを使用します。
- `nnmloadipmappings.ovpl` コマンドを使用します。

詳細については，NNMi ヘルプ，または`nnmloadipmappings.ovpl` のリファレンスページを参照してください。

13.7.1　プライベート IP アドレスの範囲

Internet Engineering Task Force（IETF）および Internet Assigned Numbers Authority（IANA）では，次の IP アドレス範囲をプライベートネットワーク（企業のローカルエリアネットワーク（LAN），企業のオフィス，または住宅用のネットワークなど）用に予約しています。

IPv4 プライベートアドレス範囲（RFC1918）：

- 10.0.0.0～10.255.255.255（24 ビットブロック）
- 172.16.0.0～172.31.255.255（20 ビットブロック）
- 192.168.0.0～192.168.255.255（16 ビットブロック）

IPv6 プライベートアドレス範囲：

- fc00::/7 アドレスブロック=RFC4193 ユニークローカルアドレス（ULA）
- fec0::/10 アドレスブロック=非推奨（RFC3879）

14 NNMi のセキュリティおよびマルチテナント

NNMi セキュリティおよびマルチテナントでは，NNMi データベースのオブジェクトに関する情報へのユーザーアクセスを制限できます。この制限は，ネットワークオペレータのビューをその責任範囲に合わせてカスタマイズする場合やサービスプロバイダが NNMi を組織ごとに設定する場合に役立ちます。この章では，NNMi セキュリティおよびテナントモデルについて説明し，設定の推奨事項について記載します。デフォルトでは，NNMi コンソールユーザーが NNMi データベースのすべてのオブジェクトを参照できます。使用環境でデフォルト設定を許容できる場合，この章は必要ありません。

14.1 オブジェクトのアクセス制限による影響

NNMi セキュリティを設定すると次のような影響があります。

トポロジインベントリオブジェクト：

- NNMi コンソールユーザーには，そのユーザーの NNMi ユーザーアカウント設定に対応するノードだけが表示されます。
- インタフェースなどのサブノードオブジェクトは，そのノードからアクセス制御を継承します。
- 接続などのノード間オブジェクトは，NNMi コンソールユーザーが関連するノードを 1 つ以上表示できる場合にだけ表示されます。
- NNMi コンソールユーザーには，ノードグループの中の 1 つ以上のノードにそのユーザーがアクセスできるノードグループだけが表示されます。

マップおよびパスビュー：

- マップには，関与している両方のノードを表示する権限を NNMi コンソールユーザーが持っている接続が表示されます。
- パスビューでは，NNMi コンソールユーザーがアクセスできないすべての中間ノードは省略されるか，クラウドとして表示されます。

インシデント：

- ソースノードが NNMi トポロジ内にあるインシデントについては，NNMi コンソールユーザーがソースノードにアクセスできるインシデントだけが表示されます。
- NNMi の稼働状態およびライセンス管理イベントのインシデントなど，ソースノードが含まれないインシデントは，1 つのグループとして処理されます。NNMi 管理者は，ユーザーに［未解決のインシデント］セキュリティグループを関連づけることで，どの NNMi コンソールユーザーにそれらのインシデントが表示されるかを決定します。
- ソースノードが NNMi トポロジ内にないトラップから生じたインシデントは，ソースノードが含まれないインシデントと同様に処理されます。これらのインシデントを生成するように NNMi が設定されている場合，NNMi 管理者は，ユーザーに［未解決のインシデント］セキュリティグループを関連づけることで，どの NNMi コンソールユーザーにそれらのインシデントが表示されるかを決定します。

インシデントの割り当てアクションでは，ユーザーのアクセス権はチェックされません。NNMi 管理者によって，あるインシデントがそのインシデントを表示する権限を持たない NNMi コンソールユーザーに割り当てられるおそれがあります。

NNMi コンソールアクション：

- 何も選択しないで実行されるアクションの場合，NNMi コンソールユーザーが実行する権限を持っているアクションだけが表示されます。
- 選択された 1 つ以上のオブジェクトに対して実行されるアクションの場合，NNMi コンソールユーザーは，選択されたオブジェクトに対する適切なアクセスレベルを持っている必要があります。セ

キュリティ設定によっては，NNMi コンソールビューに表示されている一部のオブジェクトに対して有効ではないアクションが NNMi コンソールに表示される場合もあります。これらの無効なアクションを実行すると，この制限に関するエラーメッセージが表示されます。

- マップビューについては，NNMi は，不明なノードと，NNMi トポロジ内に存在するが現在のユーザーがアクセスできないノードの区別ができません。

MIB ブラウザおよび線グラフ：

- NNMi コンソールユーザーは，ユーザーがアクセスできるノードの MIB データとグラフを表示できます。
- NNMi コンソールユーザーは，ユーザーが SNMP コミュニティ文字列を認識しているノードの MIB データを表示できます。

NNMi コンソール URL：

ダイレクト URL から NNMi コンソールビューにアクセスするには，NNMi にサインインする必要があります。NNMi は，NNMi セキュリティ設定に応じてユーザーのアクセス権を適用し，それに従って，使用できるトポロジを制限します。

14.2 NNMi のセキュリティモデル

NNMi セキュリティモデルでは，NNMi データベースのオブジェクトへのユーザーアクセスを制御できます。このモデルは，NNMi ユーザーのアクセスを特定のオブジェクトやインシデントに制限するネットワーク管理組織で使用する場合に適しています。NNMi セキュリティモデルには，次の利点があります。

- NNMi コンソールオペレータのネットワークのビューを制限できます。オペレータは特定のデバイスタイプまたはネットワーク領域に集中できます。
- NNMi トポロジへのオペレータアクセスをカスタマイズできます。オペレータアクセスのレベルは，ノードごとに設定できます。
- ［ノード（すべての属性）］ビューをセキュリティグループでフィルタリングできます。
- セキュリティ設定で構成されるノードグループの設定およびメンテナンスが簡素化されます。
- NNMi テナントモデルとは独立して使用できます。

NNMi セキュリティは，次のような場合に使用されます。

- NNMi オペレータがサイト（カスタムマップ）内の機器タイプに集中できるようにします。
- 特定のサイト（カスタムマップ）のノードだけが表示される各サイトビューを NNMi オペレータに提供します。
- 導入時にノードをステージングします。NNMi 管理者にはすべてのノードが表示されますが，NNMi オペレータには導入したノードだけが表示されます。
- すべての NOC オペレータにフルアクセスを付与し，NOC ユーザーのアクセスを制限します。
- 中央の NOC オペレータに完全なネットワークビューを提供し，地域の NOC オペレータのビューを制限します。

14.2.1 セキュリティグループ

NNMi セキュリティモデルでは，ノードへのユーザーアクセスはユーザーグループおよびセキュリティグループを介して間接的に制御されます。NNMi トポロジ内の各ノードは，1 つのセキュリティグループだけに関連づけられます。セキュリティグループは複数のユーザーグループに関連づけることができます。

各ユーザーアカウントは，次のユーザーグループにマッピングされます。

次に示す事前設定された 1 つ以上の NNMi ユーザーグループ

- NNMi 管理者
- NNMi グローバルオペレーター
- NNMi レベル 1 オペレーター
- NNMi レベル 2 オペレーター

- NNMi ゲストユーザー

マッピングは NNMi コンソールのアクセスに必要です。これによって，NNMi コンソール内で使用できるアクションが決まります。ユーザーアカウントがこれらの複数の NNMi ユーザーグループにマッピングされている場合，許可されるアクションのスーパーセットがユーザーに付与されます。

[NNMi Web サービスクライアント] ユーザーグループでは，NNMi コンソールへのアクセス権は付与されませんが，すべての NNMi オブジェクトへの管理者レベルのアクセス権が付与されます。

NNMi グローバルオペレータユーザーグループ（`globalops`）では，すべてのトポロジオブジェクトだけにアクセス権が与えられます。ユーザーが NNMi コンソールにアクセスするには，ユーザーをほかのどれかのユーザーグループ（`level2`，`level1`，または`guest`）に割り当てる必要があります。

`globalops` ユーザーグループはデフォルトですべてのセキュリティグループにマッピングされるため，管理者はこのユーザーグループをセキュリティグループにマッピングしないようにする必要があります。

セキュリティグループにマッピングされるカスタムユーザーグループ

これらのマッピングでは，NNMi データベースのオブジェクトへのアクセスが提供されます。各マッピングには，セキュリティグループのノードに適用されるオブジェクトアクセス権レベルが含まれています。オブジェクトアクセス権レベルは，インタフェースやインシデントなどの関連するデータベースオブジェクトにも適用されます。例えば，インタフェース X および Y を含むノード A へのオブジェクトオペレータレベル 1 のアクセス権があるユーザーには，次のすべてのデータベースオブジェクトへのオブジェクトオペレータレベル 1 のアクセス権があります。

- ノード A
- インタフェース X および Y
- ソースオブジェクトがノード A，インタフェース X，またはインタフェース Y のインシデント

NNMi には，次のセキュリティグループがあります。

デフォルトのセキュリティグループ

新しい NNMi インストール済み環境では，**[デフォルトのセキュリティグループ]** がすべてのノードに対する初期セキュリティグループとして割り当てられます。デフォルトでは，すべてのユーザーに，**[デフォルトのセキュリティグループ]** 内のすべてのオブジェクトが表示されます。NNMi 管理者は，**[デフォルトのセキュリティグループ]** に関連づけられるノードと，**[デフォルトのセキュリティグループ]** 内のオブジェクトにアクセスできるユーザーを設定できます。

未解決のインシデント

[未解決のインシデント] セキュリティグループは，ソースノードが NNMi トポロジ内にない受信トラップから NNMi が作成するインシデントへのアクセス権を提供します。デフォルトでは，すべてのユーザーに，**[未解決のインシデント]** セキュリティグループに関連づけられたすべてのインシデントが表示されます。NNMi 管理者は，**[未解決のインシデント]** セキュリティグループに関連づけられたインシデントにアクセスできるユーザーを設定できます。

すべてのセンサーは，ノードのセキュリティグループの割り当てを継承します。

ベストプラクティス

次のベストプラクティスが NNMi セキュリティ設定に適用されます。

- 各ユーザーアカウントを事前設定された 1 つの NNMi ユーザーグループだけにマッピングします。
- 事前設定された NNMi ユーザーグループをセキュリティグループにマッピングしないでください。
- ［NNMi 管理者］ユーザーグループにマッピングされたすべてのユーザーアカウントには，NNMi データベースのすべてのオブジェクトに対する管理者レベルのアクセス権が付与されるため，このユーザーアカウントをほかのユーザーグループにマッピングしないでください。
- Web サービスクライアントロール専用のユーザーアカウントを別個に作成します。このユーザーアカウントは NNMi トポロジ全体にアクセスできるため，このユーザーアカウントは［NNMi Web サービスクライアント］ユーザーグループにだけマッピングしてください。

14.2.2　セキュリティグループ構造の例

次の図に示すユーザーの枠は，NNMi トポロジの例で，ユーザーに表示する必要のあるノードのプライマリグループを示しています。ユーザーアクセスを完全に制御するには，サブグループが一意のセキュリティグループに対応している必要があります。一意の各セキュリティグループを 1 つ以上のユーザーグループにマッピングして，そのセキュリティグループ内のオブジェクトに対して使用できるユーザーアクセスのレベルを表すことができます。

表 14-1 に，トポロジでのセキュリティグループと考えられるカスタムユーザーグループ間のマッピングを示します。セキュリティモデルを実際に実装する場合，これらのカスタムユーザーグループの一部は不要になることがあります。

表 14-2 に，このトポロジでの幾つかのユーザーアカウントとユーザーグループのマッピングを示します。

図 14-1　ユーザーアクセス要件に対応するトポロジの例

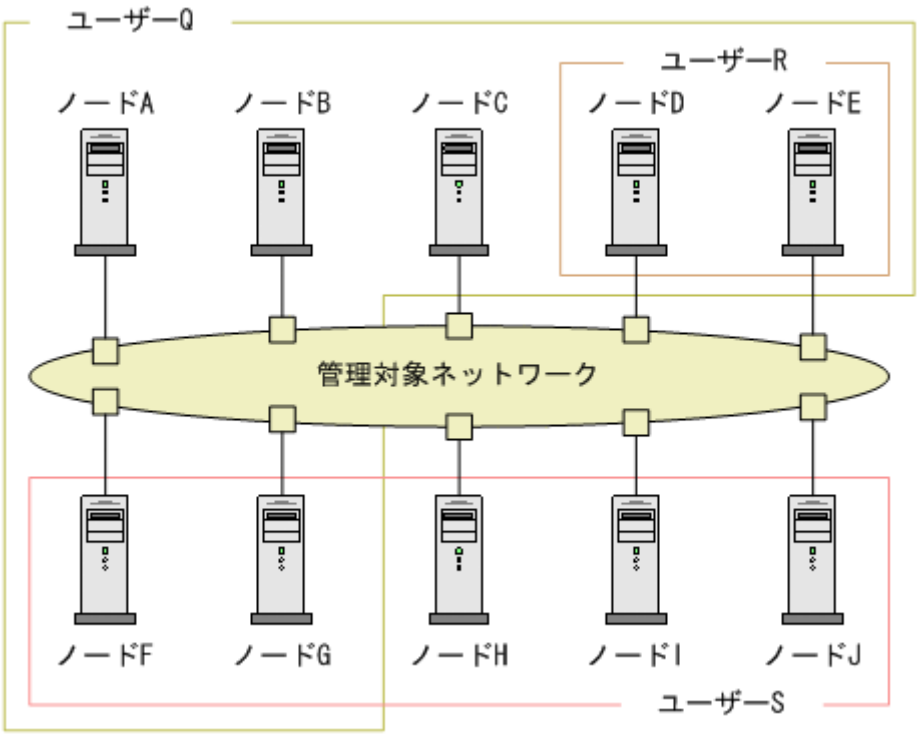


表 14-1　セキュリティグループマッピングの例

セキュリティグループ	セキュリティグループのノード	ユーザーグループ	オブジェクトアクセス権
SG1	A，B，C	UG1 管理者	オブジェクト管理者
		UG1 レベル 2	オブジェクトオペレータレベル 2
		UG1 レベル 1	オブジェクトオペレータレベル 1
		UG1 ゲスト	オブジェクトゲスト
SG2	D，E	UG2 管理者	オブジェクト管理者
		UG2 レベル 2	オブジェクトオペレータレベル 2
		UG2 レベル 1	オブジェクトオペレータレベル 1
		UG2 ゲスト	オブジェクトゲスト
SG3	F，G	UG3 管理者	オブジェクト管理者
		UG3 レベル 2	オブジェクトオペレータレベル 2
		UG3 レベル 1	オブジェクトオペレータレベル 1
		UG3 ゲスト	オブジェクトゲスト
SG4	H，I，J	UG4 管理者	オブジェクト管理者
		UG4 レベル 2	オブジェクトオペレータレベル 2
		UG4 レベル 1	オブジェクトオペレータレベル 1
		UG4 ゲスト	オブジェクトゲスト

表 14-2　ユーザーアカウントマッピングの例

ユーザーアカウント	ユーザーグループ	ノードアクセス	注
ユーザー Q	NNMi レベル 2 オペレータ	なし	ユーザー Q の枠に含まれるノードへのオペレータレベル 2 のアクセス権があります。
	UG1 レベル 2	A，B，C	
	UG2 レベル 2	D，E	
	UG3 レベル 2	F，G	
ユーザー R	NNMi レベル 1 オペレータ	なし	ユーザー R の枠に含まれるノードへのオペレータレベル 1 のアクセス権があります。
	UG2 レベル 1	D，E	
ユーザー S	NNMi レベル 2 オペレータ	なし	ユーザー S の枠に含まれるノードへのオペレータレベル 2 のアクセス権があります。
	UG3 レベル 2	F，G	
	UG4 レベル 2	H，I，J	

ユーザーアカウント	ユーザーグループ	ノードアクセス	注
ユーザー T	NNMi レベル 2 オペレータ	なし	ユーザー T は，トポロジの例に含まれるすべてのノードに（各権限レベルで）アクセスできます。 このユーザーには，ノード D および E への管理アクセス権がありますが，管理アクセス権が必要なツールのメニュー項目は表示できません。ユーザーに NNMi 管理サーバーへのアクセス権がある場合は，ノード D および E に対してだけ，管理アクセス権が必要なコマンドラインツールを実行できます。
	UG1 ゲスト	A，B，C	
	UG2 管理者	D，E	
	UG3 レベル 2	F，G	
	UG4 レベル 1	H，I，J	

14.3 NNMi のテナントモデル

NNMi テナントモデルでは，トポロジ検出とトポロジデータが各テナント（組織または顧客とも呼ばれる）で完全に分離されます。このモデルは，サービスプロバイダ（特に管理対象サービスプロバイダ）や大規模エンタープライズに適しています。

NNMi テナントモデルには，次の利点があります。

- 各ノードが属する組織が明確になります。
- ［ノード（すべての属性）］インベントリビューを，テナントとセキュリティグループでフィルタリングできます。
- 顧客データへのオペレータアクセスを分離する規制要件に適合します。
- テナント設定で構成されるノードグループの設定およびメンテナンスが簡素化されます。
- NNMi セキュリティの設定が簡素化されます。

NNMi マルチテナントを使用すると，同じ NNMi 管理サーバーで複数の顧客（テナント）を管理するサービスプロバイダに，異なる顧客ビューを提供できます。

14.3.1 テナント

NNMi テナントモデルでは，組織という概念がセキュリティ設定に加わります。NNMi トポロジ内の各ノードが属するテナントは 1 つだけです。テナントによって，NNMi データベースが論理的に分離されます。オブジェクトアクセスはセキュリティグループで管理されます。

ノードが最初に検出されて NNMi データベースに追加されるときに，各ノードで初期検出テナントの割り当てが発生します。シード済みのノードで，各ノードに割り当てるテナントを指定できます。NNMi によって，検出されたほかのすべてのノード（自動検出ルールに含まれているが直接シードされないノード）がデフォルトテナントに割り当てられます。NNMi 管理者は，検出後にいつでもノードのテナントを変更できます。

各テナント定義には，初期検出セキュリティグループが含まれます。NNMi によって，初期検出セキュリティグループが初期検出テナントとともにノードに割り当てられます。NNMi 管理者は，検出後にいつでもノードのセキュリティグループを変更できます。

ノードのテナントの割り当てを変更しても，セキュリティグループの割り当ては自動的に変更されません。

NNMi には，デフォルトテナントが備わっています。デフォルトでは，すべての NNMi ユーザーが，［デフォルトのセキュリティグループ］を介して，テナントに関連づけられたすべてのオブジェクトにアクセスできます。

すべてのセンサーは，ノードのテナントおよびセキュリティグループの割り当てを継承します。

ベストプラクティス

次のベストプラクティスが NNMi テナント設定に適用されます。

- 小規模な組織の場合，テナントごとに 1 つのセキュリティグループで十分です。
- 大規模な組織を複数のセキュリティグループに分割できます。
- ユーザーが組織を超えてノードにアクセスできないようにするには，各セキュリティグループに，1 つのテナントだけに対応するノードしか含まれないようにします。

14.3.2　テナント構造の例

次の図では，NNMi トポロジ内に 2 つのテナントが含まれている様子を示します。ユーザー L，M，N の枠は，ユーザーにノードを表示する必要があるプライマリグループを表しています。テナント 1 のトポロジは 1 つのグループとして管理されるため，1 つのセキュリティグループだけが必要です。テナント 2 のトポロジは重複しているセットで管理されるため，3 つのセキュリティグループに分割されます。

表 14-3 に，トポロジでのセキュリティグループと考えられるカスタムユーザーグループ間のマッピングを示します（このセキュリティモデルを実際に実装する場合，これらのカスタムユーザーグループの一部は不要になることがあります）。

表 14-4 に，このトポロジでの幾つかのユーザーアカウントとユーザーグループのマッピングを示します。

図 14-2　複数のテナントのトポロジの例

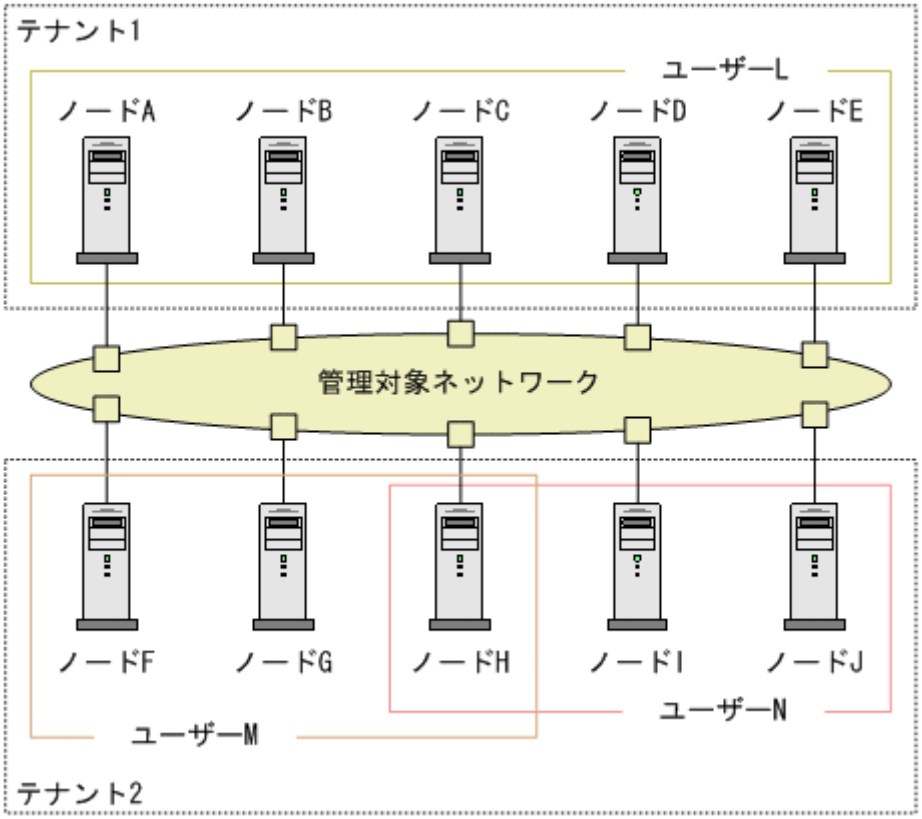


表 14-3　複数のテナントのセキュリティグループマッピングの例

セキュリティグループ	セキュリティグループのノード	ユーザーグループ	オブジェクトアクセス権
T1 SG	A，B，C，D，E	T1 管理者	オブジェクト管理者
		T1 レベル 2	オブジェクトオペレータレベル 2
		T1 レベル 1	オブジェクトオペレータレベル 1
		T1 ゲスト	オブジェクトゲスト
T2 SGa	F，G	T2_a 管理者	オブジェクト管理者
		T2_a レベル 2	オブジェクトオペレータレベル 2
		T2_a レベル 1	オブジェクトオペレータレベル 1
		T2_a ゲスト	オブジェクトゲスト
T2 SGb	H	T2_b 管理者	オブジェクト管理者
		T2_b レベル 2	オブジェクトオペレータレベル 2
		T2_b レベル 1	オブジェクトオペレータレベル 1
		T2_b ゲスト	オブジェクトゲスト
T2 SGc	I，J	T2_c 管理者	オブジェクト管理者
		T2_c レベル 2	オブジェクトオペレータレベル 2
		T2_c レベル 1	オブジェクトオペレータレベル 1
		T2_c ゲスト	オブジェクトゲスト

表 14-4　複数のテナントのユーザーアカウントマッピングの例

ユーザーアカウント	ユーザーグループ	ノードアクセス	注
ユーザー L	NNMi レベル 2 オペレータ	なし	ユーザー L には，テナント 1 のすべてのノードをグループ化する，ユーザー L の枠に含まれるノードへのオペレータレベル 2 のアクセス権があります。
	T1 レベル 2	A，B，C，D，E	
ユーザー M	NNMi レベル 1 オペレータ	なし	ユーザー M には，テナント 2 のノードのサブセットをグループ化する，ユーザー M の枠に含まれるノードへのオペレータレベル 1 のアクセス権があります。
	T2_a レベル 1	F，G	
	T2_b レベル 1	H	
ユーザー N	NNMi レベル 2 オペレータ	なし	ユーザー N には，テナント 2 のノードのサブセットをグループ化する，ユーザー N の枠に含まれるノードへのオペレータレベル 2 のアクセス権があります。
	T2_b レベル 2	H	
	T2_c レベル 2	I，J	

14.4 NNMi のセキュリティおよびマルチテナントを設定する

NNMi のセキュリティおよびマルチテナント設定は，NNMi データベース全体に適用されます。NNMi 管理者であれば，すべてのテナントのすべてのオブジェクトへのオペレータアクセス権を表示および設定できます。

NNMi 管理者が 1 つ以上のカスタムセキュリティグループを定義すると，［セキュリティグループ］がすべての［ノード］フォームに表示されます。また，［ノード］および［ノード（すべての属性）］インベントリビューの列としても表示されます。

NNMi 管理者が 1 つ以上のカスタムテナントを定義すると，［テナント］フィールドがすべての［ノード］フォームに表示されます。また，［ノード］および［ノード（すべての属性）］インベントリビューの列としても表示されます。

ノードグループ

セキュリティ設定またはマルチテナント設定の一部と適合するようにノードグループを作成するには，セキュリティグループ UUID，セキュリティグループ名，テナント UUID，またはテナント名に基づいて，ノードグループの追加のフィルターを指定します。これらのノードグループを使用して，監視アクションおよびインシデントライフサイクル移行アクション用のポーリングサイクルを，セキュリティグループまたはテナントごとに設定します。

ヒント

セキュリティグループとテナントの名前は変更できるため，追加のフィルターにはセキュリティグループまたはテナントの UUID を指定します。この情報は，設定フォームと，`nnmsecurity.ovpl` コマンド出力で使用できます。

ユーザーグループ：NNMi コンソールアクセス

事前に定義された NNMi ユーザーグループの 1 つにユーザーアカウントをマッピングすると，NNMi ロールと，NNMi コンソールで表示されるメニュー項目が設定されます。各ユーザーアカウントには，そのユーザーのトポロジオブジェクトに対する最も高いオブジェクトのアクセス権に対応する NNMi ロールを付与することをお勧めします。

ただし，NNMi 管理者はすべてのトポロジオブジェクトへのアクセス権を持つため，管理者レベルの権限を付与することは避けてください。NNMi トポロジ内の一部のノードに対してだけ，NNMi コンソールユーザーを管理者として設定するには，そのユーザーを NNMi レベル 1 オペレータまたは NNMi レベル 2 オペレータのユーザーグループに割り当てます。また，オブジェクト管理者オブジェクトアクセス権を使用して，トポロジ内のノードのサブセットを含むセキュリティグループにマッピングされたカスタムユーザーグループを作成し，ユーザーをそのグループに割り当てます。

ユーザーグループ：ディレクトリサービス

ユーザーグループメンバーシップを NNMi データベースに保存する場合，すべてのオブジェクトアクセス設定は，NNMi 設定エリア内で，ユーザーグループ，ユーザーアカウントマッピング，セキュリティグループ，およびセキュリティグループマッピングを使用します。

ユーザーグループメンバーシップをディレクトリサービスに保存する場合，オブジェクトアクセス設定は，NNMi 設定（セキュリティグループおよびセキュリティグループマッピング）と，ディレクトリサービスコンテンツ（ユーザーグループメンバーシップ）の間で共有されます。NNMi データベースに，ユーザーアカウントまたはユーザーアカウントマッピングを作成しないでください。ディレクトリサービス内の適用可能なグループごとに，NNMi データベースに 1 つ以上のユーザーグループを作成してください。NNMi で，各ユーザーグループ定義の［ディレクトリサービス名］フィールドに，ディレクトリサービス内のそのグループの識別名を設定します。

詳細については，「12. NNMi と LDAP によるディレクトリサービスの統合」を参照してください。

14.4.1 セキュリティおよびマルチテナントの設定ツール

NNMi には，マルチテナントとセキュリティを設定するための幾つかのツールが備わっています。

セキュリティウィザード

NNMi コンソールの［セキュリティウィザード］は，セキュリティ設定の可視化に役立ちます。NNMi コンソール内でノードをセキュリティグループに割り当てるには，このウィザードを使用する方法が最も簡単です。［変更概要の表示］ページには，現在のウィザードセッションで保存されていない変更点のリストが表示されます。また，セキュリティ設定に関する潜在的な問題も示されます。

［セキュリティウィザード］の使用法の詳細については，ウィザード内の NNMi ヘルプリンクをクリックしてください。

> **メモ**
>
> ［セキュリティウィザード］は，NNMi セキュリティ設定に関してだけ使用できます。テナント情報は含まれていません。

NNMi コンソールフォーム

NNMi コンソール内の個々のセキュリティオブジェクトおよびマルチテナントオブジェクトのフォームは，設定の 1 つの側面を同時に集中的に捉える場合に便利です。これらのフォームの使用法の詳細については，各フォームの NNMi ヘルプを参照してください。

［テナント］ビューには NNMi マルチテナント設定情報が含まれています。このビューは，［設定］ワークスペースの［検出］の下に表示されます。各［テナント］フォームには 1 つの NNMi テナントが記述され，現在そのテナントに割り当てられているノードが表示されます。ノードの割り当て情報は読み取り専用です。

ノードに割り当てられているテナントまたはセキュリティグループを変更するには，［ノード］フォームまたは`nnmsecurity.ovpl` コマンドを使用します。

次の NNMi コンソールビューは，［設定］ワークスペースの［セキュリティ］の下に表示されます。これらのビューには，次の NNMi セキュリティ設定情報が含まれています。

ユーザーアカウント

・各［ユーザーアカウント］フォームには 1 つの NNMi ユーザーが記述され，そのユーザーが属するユーザーグループが表示されます。メンバーシップ情報は読み取り専用です。

・ユーザーグループメンバーシップをディレクトリサービスに保存すると，ユーザーアカウントはNNMi コンソールに表示されません。

ユーザーグループ

各［**ユーザーグループ**］フォームには 1 つの NNMi ユーザーグループが記述され，そのユーザーグループにマッピングされたユーザーアカウントとセキュリティグループが表示されます。マッピング情報は読み取り専用です。

ユーザーアカウントのマッピング

・各［**ユーザーアカウントのマッピング**］フォームには，1 つのユーザーアカウントとユーザーグループの関連づけが表示されます。

・ユーザーアカウントマッピングを変更しても，現在の NNMi コンソールユーザーにその変更は反映されません。現在のユーザーは，NNMi コンソールに次回のサインインで，変更を受け取ります。

・ユーザーグループメンバーシップをディレクトリサービスに保存すると，ユーザーアカウントマッピングは NNMi コンソールに表示されません。

セキュリティグループ

各［**セキュリティグループ**］フォームには 1 つの NNMi セキュリティグループが記述され，そのセキュリティグループに現在割り当てられているノードが表示されます。ノードの割り当て情報は読み取り専用です。

セキュリティグループのマッピング

・各［**セキュリティグループのマッピング**］フォームには，1 つのユーザーグループとセキュリティグループの関連づけが表示されます。

・初期設定のあと，セキュリティグループマッピングに関連づけられたオブジェクトのアクセス権は読み取り専用になっています。セキュリティグループマッピングのオブジェクトアクセス権を変更するには，そのマッピングを削除して，再度作成します。

コマンドライン

`nnmsecurity.ovpl` コマンドラインインタフェースは，自動操作や一括操作する場合に便利です。このツールは，セキュリティ設定に関する潜在的な問題のレポートも提供します。

`nnmsecurity.ovpl` オプションの多くは，コンマ区切り値（CSV）ファイルからの入力データのロードをサポートしています。設定データは，`nnmsecurity.ovpl` コマンドで使用するために，CSV 出力を生成できるファイルまたはシステムに保持できます。このコマンドは，NNMi の外部で生成された UUID も受け入れます。

ヒント

セキュリティグループとテナントの名前は一意である必要はないため，`nnmsecurity.ovpl` コマンドへの入力値としてセキュリティグループまたはテナントの UUID を指定します。

次のスクリプト例では，`nnmsecurity.ovpl` コマンドを使用して，2 つのユーザーアカウントと 5 つのノードにセキュリティ設定を作成しています。

```
#!/bin/sh
# ユーザーを2つ作成する
```

```
nnmsecurity.ovpl -createUserAccount user1 -password password -role level1
nnmsecurity.ovpl -createUserAccount user2 -password password -role level2
# グループを2つ作成する
nnmsecurity.ovpl -createUserGroup local1
nnmsecurity.ovpl -createUserGroup local2
# 新しいユーザーグループにユーザーアカウントを割り当てる
nnmsecurity.ovpl -assignUserToGroup -user user1 -userGroup local1
nnmsecurity.ovpl -assignUserToGroup -user user2 -userGroup local2
# セキュリティグループを2つ作成する
nnmsecurity.ovpl -createSecurityGroup secgroup1
nnmsecurity.ovpl -createSecurityGroup secgroup2
# 新しいセキュリティグループに新しいユーザーグループを割り当てる
nnmsecurity.ovpl -assignUserGroupToSecurityGroup -userGroup local1 -securityGroup secgrou
p1 -role level1
nnmsecurity.ovpl -assignUserGroupToSecurityGroup -userGroup local2 -securityGroup secgrou
p2 -role level2
# セキュリティグループをノードに割り当てる
nnmsecurity.ovpl -assignNodeToSecurityGroup -node mplspe01 -securityGroup secgroup1
nnmsecurity.ovpl -assignNodeToSecurityGroup -node vwan_router-1 -securityGroup secgroup1
nnmsecurity.ovpl -assignNodeToSecurityGroup -node vwan_router-2 -securityGroup secgroup1
nnmsecurity.ovpl -assignNodeToSecurityGroup -node data_center_1 -securityGroup secgroup2
nnmsecurity.ovpl -assignNodeToSecurityGroup -node mplspe03 -securityGroup secgroup2
```

14.4.2 マルチテナントを設定する

次の方法でマルチテナントを設定できます。

- NNMi コンソールの［テナント］フォーム
  個々のテナントを処理する際に役立ちます。
- nnmsecurity.ovpl コマンドラインインタフェース
  自動操作や一括操作する場合に便利です。このツールは，テナント設定に関する潜在的な問題のレポートも提供します。

それぞれの NNMi トポロジオブジェクトをテナント（組織）に割り当てるため，NNMi マルチテナントを定義および設定するプロセスは循環的なプロセスです。

NNMi マルチテナントの設定に関しては，次の点に注意してください。

- 検出されたノードに割り当てられるセキュリティグループは，そのノードに関連づけられたテナントの［初期検出セキュリティグループ］の値によって設定されます。
- NNMi テナントを設定しないで，NNMi セキュリティモデルを使用すると，すべてのノードにデフォルトテナントが割り当てられます。
- NNMi 検出用にノードをシードするときに，そのノードが属するテナントを指定できます。自動検出ルールを使用して NNMi でノードが検出されると，NNMi によって，そのノードはデフォルトテナントに割り当てられます。検出後，ノードに対するテナントの割り当てを変更できます。

NNMi マルチテナントを計画および設定するための概略的な方法を次に示します。この概略的な手順では，NNMi マルチテナントを設定するための 1 つの方法を説明します。

1. ユーザー要件を分析して，NNMi 環境で必要なテナントの数を判別する。

   1 つの NNMi 管理サーバーで複数のネットワークを個々に管理する場合だけ，テナントを使用することをお勧めします。

2. 管理対象のネットワークトポロジを分析して，各テナントにどのノードが属するかを判別する。
3. 各テナントのトポロジを分析して，NNMi ユーザーがアクセスする必要のあるノードのグループを判別する。
4. 事前に定義された NNMi ユーザーグループと，［デフォルトのセキュリティグループ］および［未解決のインシデント］セキュリティグループの間のデフォルトの関係を削除する。

   この手順によって，ユーザーが管理してはならないノードへのアクセス権が，そのユーザーに誤って付与されないようにします。この時点では，NNMi トポロジ内のオブジェクトにアクセスできるのは NNMi 管理者だけです。

5. 特定されたテナントを設定する。

   a　特定されたセキュリティグループを作成します。

   b　特定されたテナントを作成します。

   テナントごとに，［デフォルトのセキュリティグループ］，またはアクセスが制限されたテナント固有のセキュリティグループのどれかに，［初期検出セキュリティグループ］を設定します。これを行うことで，NNMi 管理者がアクセス権を設定するまで，テナントの新しいノードが全体に表示されることはなくなります。

6. テナントをシードに割り当てて，検出の準備をする。

   ノードのグループを検出したあと，［初期検出セキュリティグループ］の値を変更できます。これを行うことで，ノードをセキュリティグループに手動で再割り当てする処理が制限されます。

7. 検出が完了したら，次を実行する。

   - ノードごとにテナントを確認し，必要に応じて変更します。
   - ノードごとにセキュリティグループを確認し，必要に応じて変更します。

8. カスタムユーザーグループを設定する。

   カスタムユーザーグループの設定については，「14.4.4　セキュリティ設定を確認する」を参照してください。

14.4.3　セキュリティグループを設定する

ディレクトリサービスと NNMi を統合して，ユーザー名，パスワード，およびオプションとして NNMi ユーザーグループの割り当ての保管場所を統合する場合は，NNMi セキュリティを設定する前に，その統合の設定を実行してください。

NNMi では，次の方法でセキュリティを設定できます。

- NNMi コンソールの［セキュリティウィザード］

  セキュリティ設定の可視化に役立ちます。［変更概要の表示］ページには，現在のウィザードセッションで保存されていない変更点のリストが表示されます。また，セキュリティ設定に関する潜在的な問題も示されます。

- 個々のセキュリティオブジェクトに対応した NNMi コンソールのフォーム

  セキュリティ設定の 1 つの側面を同時に集中的に捉える場合に便利です。

- `nnmsecurity.ovpl` コマンドラインインタフェース

  自動操作や一括操作する場合に便利です。このツールは，セキュリティ設定に関する潜在的な問題のレポートも提供します。

NNMi トポロジ内のオブジェクトに対するユーザーのアクセス権を制限するために，NNMi セキュリティを定義および設定するプロセスは，循環的なプロセスです。

> **メモ**
>
> この設定方法は，セキュリティグループからユーザーアカウントに移動します。例えば，ユーザーアカウントからセキュリティグループに NNMi セキュリティを設定する場合，NNMi ヘルプで「セキュリティの設定例」を検索してください。

NNMi セキュリティの設定に関しては，次の点に注意してください。

- 検出されたノードに割り当てられるセキュリティグループは，そのノードに関連づけられたテナントの［初期検出セキュリティグループ］の値によって設定されます。
- NNMi テナントを設定しないで，NNMi セキュリティモデルを使用すると，すべてのノードがデフォルトテナントに割り当てられます。

NNMi セキュリティを計画および設定するための概略的な方法を次に示します。この概略的な手順では，NNMi セキュリティを設定するための 1 つの方法を説明します。

1. 管理対象のネットワークトポロジを分析して，NNMi ユーザーがアクセスする必要のあるノードのグループを判別する。
2. 事前に定義された NNMi ユーザーグループと，［デフォルトのセキュリティグループ］および［未解決のインシデント］セキュリティグループの間のデフォルトの関係を削除する。

   この手順によって，ユーザーが管理してはならないノードへのアクセス権が，そのユーザーに誤って付与されることがないようにします。この時点では，NNMi トポロジ内のオブジェクトにアクセスできるのは NNMi 管理者だけです。
3. ノードの各サブセットのセキュリティグループを設定する。

   特定のノードは 1 つのセキュリティグループにだけ属することができます。

   a　セキュリティグループを作成します。

   b　適切なノードを各セキュリティグループに割り当てます。
4. カスタムユーザーグループを設定する。

a　セキュリティグループごとに，NNMi ユーザーアクセスの各レベルに対応するユーザーグループを設定します。

- ユーザーグループメンバーシップを NNMi データベースに保存しても，それらのユーザーグループにユーザーはマッピングされません。
- ユーザーグループメンバーシップをディレクトリサービスに保存する場合は，各ユーザーグループの［**ディレクトリサービス名**］フィールドに，ディレクトリサービス内のそのグループの識別名を設定します。

b　各カスタムユーザーグループを，適切なセキュリティグループにマッピングします。マッピングごとに適切なオブジェクトアクセス権を設定します。

5. ユーザーアカウントを設定する。

ユーザーグループメンバーシップを NNMi データベースに保存する場合は，次の手順を実行します。

- NNMi コンソールにアクセスできるユーザーごとに，ユーザーアカウントオブジェクトを作成します。ユーザーアカウントを設定するプロセスは，NNMi コンソールログオンにディレクトリサービスを使用しているかどうかによって異なります。
- 各ユーザーアカウントを NNMi コンソールにアクセスするために，事前に定義した NNMi ユーザーグループの 1 つにマッピングします。
- 各ユーザーアカウントをトポロジオブジェクトにアクセスするために，1 つ以上のカスタム NNMi ユーザーグループにマッピングします。

ユーザーグループメンバーシップをディレクトリサービスに保存する場合，各ユーザーが，事前に定義された NNMi ユーザーグループの 1 つ，および 1 つ以上のカスタムユーザーグループに属していることを確認します。

6. 「14.4.4　セキュリティ設定を確認する」の説明に従って，設定を確認する。
7. セキュリティ設定を管理する。

- ［**デフォルトのセキュリティグループ**］に追加されたノードに注目し，これらのノードを適切なセキュリティグループに移動します。
- 新しい NNMi コンソールユーザーを適切なユーザーグループに追加します。

14.4.4　セキュリティ設定を確認する

セキュリティ設定が適切であるかを確認するために，設定のそれぞれの側面を個別に確認することが必要です。ここでは，設定を確認するための幾つかの方法を説明します。ここに記載されていない方法も使用できます。

メモ

NNMi には，潜在的なセキュリティ設定エラーのレポートが備わっています。これらのレポートには，NNMi コンソールの［**ツール**］＞［**セキュリティレポート**］からアクセスします。ま

たは，-displayConfigReport オプションをnnmsecurity.ovpl コマンドに指定して使用することもできます。

セキュリティグループとノード間の割り当てを確認する

各ノードが適切なセキュリティグループに割り当てられていることを次の方法で確認します。

- セキュリティグループごとに［ノード］または［ノード（すべての属性）］インベントリビューをソートし，グループ分けを調べます。
- -listNodesInSecurityGroup オプションをnnmsecurity.ovpl コマンドに指定して使用します。

ユーザーグループとセキュリティグループ間の割り当てを確認する

どのユーザーグループが各セキュリティグループにマッピングされているかを次の方法で確認します。

- ユーザーグループまたはセキュリティグループごとに［セキュリティグループのマッピング］ビューをソートして，グループ分けを調べます。また，各マッピングのオブジェクトアクセス権も確認します。
- ［セキュリティウィザード］の［ユーザーグループとセキュリティグループのマップ］ページで，同時に 1 つのユーザーグループまたはセキュリティグループを選択して，そのオブジェクトに対する現在のマッピングを確認します。
- -listUserGroupsForSecurityGroup オプションをnnmsecurity.ovpl コマンドに指定して使用します。

各ユーザーが NNMi コンソールアクセス権を持っているかを確認する

NNMi コンソールアクセス権について，事前に設定された NNMi ユーザーグループの 1 つに各ユーザーが割り当てられていることを確認します。

- NNMi 管理者
- NNMi レベル 1 オペレータ
- NNMi レベル 2 オペレータ
- NNMi ゲストユーザー

そのほかのすべてのユーザーグループ割り当てで，NNMi データベースのオブジェクトへのアクセス権が付与されます。

NNMi コンソールアクセス権を持たないユーザーは，［セキュリティウィザード］の［変更概要の表示］ページに表示されます。［ツール］＞［セキュリティレポート］メニュー項目や，-displayConfigReport usersWithoutRoles オプションをnnmsecurity.ovpl コマンドに設定して，この情報を得ることもできます。

ユーザーとユーザーグループ間の割り当てを確認する

ユーザーグループメンバーシップを次の方法で確認します。

- ユーザーアカウントまたはユーザーグループごとに［ユーザーアカウントのマッピング］ビューをソートして，グループ分けを調べます。

- ［セキュリティウィザード］の［ユーザーアカウントとユーザーグループのマップ］ページで，同時に 1 つのユーザーアカウントまたはユーザーグループを選択して，そのオブジェクトに対する現在のマップングを確認します。
- -listUserGroups オプションと-listUserGroupMembers オプションをnnmsecurity.ovpl コマンドに指定して使用します。

テナントとノード間の割り当てを確認する

各ノードが適切なテナントに割り当てられていることを確認する方法として，テナントごとに［ノード］または［ノード（すべての属性）］インベントリビューをソートし，グループ分けを調べる方法があります。

現在のユーザー設定を確認する

現在ログオンしているユーザーの NNMi コンソールアクセス権を確認するには，［ヘルプ］＞［システム情報］をクリックします。［製品］タブの［ユーザー情報］セクションに，現在の NNMi セッションに関する次の情報が表示されます。

- NNMi データベースのユーザーアカウント，またはアクセス対象のディレクトリサービスに定義されているユーザー名。
- NNMi ロール。これは，ユーザーがマッピングされる，事前に定義された NNMi ユーザーグループ（NNMi 管理者，NNMi レベル 1 オペレータ，NNMi レベル 2 オペレータ，および NNMi ゲストユーザー）の中で最も高い権限を持つものに対応します。マッピングによって，NNMi コンソールで使用できるアクションが決まります。
- ユーザー名にマッピングされたユーザーグループ。このリストには，NNMi ロールの設定前に設定された NNMi ユーザーグループと，NNMi データベース内のオブジェクトへのアクセス権を付与するそのほかのすべてのユーザーグループが含まれています。

14.4.5　セキュリティおよびマルチテナントの設定をエクスポートする

次の表は，NNMi のセキュリティおよびマルチテナント設定をエクスポートするための設定エリアを示しています。nnmconfigexport.ovpl -c コマンドで使用できます。エクスポートエリアは，特にグローバルネットワーク管理環境で，複数の NNMi 管理サーバーにわたって設定を管理するのに役立ちます。

表 14-5　NNMi のセキュリティおよびマルチテナント設定のエクスポートエリア

設定エリア	説明
account	ユーザーアカウント，ユーザーグループ，およびユーザーアカウントとユーザーグループ間のマッピングをエクスポートします。 複数の NNMi データベースにわたってユーザー定義を共有するのに便利です。
security	テナントおよびセキュリティグループをエクスポートします。 複数の NNMi データベースにわたってセキュリティ定義を共有するのに便利です。 この情報をインポートすると，新しいオブジェクトが作成され，既存のオブジェクトが更新されますが，現在のエクスポートに含まれていないオブジェクトは削除されません。このた

設定エリア	説明
security	め，ローカルで定義されたオブジェクトが NNMi データベースに含まれている場合でも，このオプションは安全に使用できます。
securitymappings	ユーザーグループとセキュリティグループ間のマッピングをエクスポートします。 セキュリティとマルチテナント設定を完全にエクスポートするには，`account`，`security`，および`securitymappings` 設定エリアの同時エクスポートを実行してください。

14.5 NNMi セキュリティとマルチテナントをグローバルネットワーク管理に定義する

グローバルネットワーク管理環境では，ノードのテナントは，そのノードを管理する NNMi 管理サーバーに設定されます。グローバルネットワーク管理環境では，指定されたノードのテナント UUID は各グローバルマネージャーとリージョナルマネージャーで同じです。

ノードのセキュリティグループは，トポロジにそのノードが含まれる各 NNMi 管理サーバーに設定されます。したがって，トポロジ内のオブジェクトへのユーザーアクセスは，グローバルネットワーク管理環境の各 NNMi 管理サーバーに別個に設定されます。グローバルマネージャーとリージョナルマネージャーが使用するセキュリティグループ定義は，同じである場合も，異なる場合もあります。

グローバルマネージャーとリージョナルマネージャーに同様のユーザーアクセスを設定する場合，幾つかの方法を使用して設定することもできますが，大部分の場合，各 NNMi 管理サーバーにカスタム設定する必要があります。

ヒント

- グローバルマネージャーにすべてのテナントとセキュリティグループを定義します。`nnmconfigexport.ovpl -c security` を使用して，テナントとセキュリティグループ定義をエクスポートします。各リージョナルマネージャーで，`nnmconfigimport.ovpl` を使用してテナントとセキュリティグループ定義をインポートします。あるいは，`nnmsecurity.ovpl` コマンドを使用して，別の NNMi 管理サーバーの UUID と同じ UUID を使用して，テナントおよびセキュリティグループを作成できます。この推奨手順に従うことで，グローバルネットワーク管理環境内で，各テナントとセキュリティグループの UUID を同じにできます。

  ユーザーがグローバルマネージャーから NPS レポートを開始する場合，このベストプラクティスは設定の必須部分になります。

  テナント UUID は一意である必要がありますが，テナント名は再利用できます。NNMi は，名前が同じで UUID が異なる 2 つのテナントを，共有設定を持たない 2 つの別個のテナントであると見なします。

- 組織ごとに 1 つのリージョナルマネージャーをセットアップする場合は，リージョナルマネージャーのすべてのノードを 1 つのテナントに入れられます。ただし，各リージョナルマネージャーに一意のテナントを設定し，グローバルマネージャーでトポロジデータが確実に分離されるようにしてください。

  リージョナルマネージャーからグローバルマネージャーに転送されたインシデントに，セキュリティ情報とテナント情報を伝達する幾つかの追加カスタムインシデント属性（CIA）が含まれる場合があります。

  このようなインシデントのソースオブジェクトがデフォルトテナント以外のテナントに属している場合，転送されるインシデントには次の CIA が含まれます。

  cia.tenant.name

  cia.tenant.uuid

このようなインシデントのソースオブジェクトが［デフォルトのセキュリティグループ］以外のセキュリティグループに属している場合，転送されるインシデントには次の CIA が含まれます。

cia.securityGroup.name

cia.securityGroup.uuid

14.5.1 グローバルネットワーク管理にセキュリティおよびマルチテナントの初期設定をする

グローバルネットワーク管理の初期設定後，リージョナルマネージャーは，グローバルネットワーク管理の設定に従って，リージョナルトポロジ内のノードに関する情報を使用して，グローバルマネージャーを更新します。

デフォルトテナントだけとのトポロジの同期

カスタムセキュリティグループとデフォルトテナントを持つグローバルネットワーク管理環境の場合，グローバルマネージャーでは，リモートで管理されているすべてのノードが，次の設定でグローバルマネージャートポロジに追加されます。

- デフォルトテナント
- デフォルトテナントの［初期検出セキュリティグループ］として設定されるセキュリティグループ。

カスタムテナントとのトポロジの同期

カスタムセキュリティグループとカスタムテナントを持つグローバルネットワーク管理環境の場合，グローバルマネージャーでは，リモートで管理されているすべてのノードが，そのノードに割り当てられているテナントの UUID を使用して，グローバルマネージャートポロジに追加されます。そのテナント UUID がグローバルマネージャーにない場合，次のように，グローバルネットワーク管理プロセスによってグローバルマネージャーの NNMi 設定にテナントが作成されます。

- テナント UUID は，リージョナルマネージャーの場合と同じ値です。
- テナント名は，リージョナルマネージャーの場合と同じ値です。
- ［初期検出セキュリティグループ］の値は，テナントと同じ名前のセキュリティグループに設定されます。なお，セキュリティグループがグローバルマネージャーにない場合，NNMi によってそのセキュリティグループが作成されます。

グローバルマネージャーのトポロジにノードが追加されると，そのノードは，グローバルマネージャーに設定されたテナント UUID に対応する［初期検出セキュリティグループ］に割り当てられます。このため，グローバルマネージャー上でのセキュリティグループの関連づけは，リージョナルマネージャー上でのセキュリティグループの関連づけから独立しています。

ヒント

グローバルマネージャーでのセキュリティ設定を簡素化するための推奨を次に示します。

- 各リージョナルマネージャーによって管理されるノードのスプレッドシート，またはそのほかのレコードを保持します。ノードごとに，リージョナルマネージャーとグローバルマネージャーのそれぞれに必要なセキュリティグループをメモしておきます。グローバルネットワーク管理の設定が完了したら，nnmsecurity.ovpl コマンドを使用して，セキュリティグループの割り当ての確認および更新をします。
- グローバルネットワーク管理環境で，複数のリージョナルマネージャーによって 1 つのグローバルマネージャーが更新されている場合，そのグローバルマネージャーに対してグローバルネットワーク管理の設定を有効にするには，各リージョナルマネージャーから 1 つずつ設定してください。
- 各リージョナルマネージャーをグローバルネットワーク管理の設定に追加する前に，デフォルトテナント（またはカスタムテナント）の［初期検出セキュリティグループ］の値を変更できます。これを実行した場合，以前に設定されたリージョナルマネージャーのトポロジに新しいノードが追加されると，さまざまな結果が生じるおそれがあることに注意してください。
- グローバルネットワーク管理を有効にする前に，グローバルマネージャー上で，リージョナルマネージャーで使用される各テナントの［初期検出セキュリティグループ］を，オペレータがアクセスできない専用セキュリティグループに設定してください。これによって，グローバルマネージャー上の管理者は，ほかの NNMi コンソールオペレータのために，ノードを適切なセキュリティグループに明示的に移動しなくてはならなくなります。

14.5.2 セキュリティおよびマルチテナントの割り当てのグローバルネットワーク管理への影響

次の表は，リージョナルマネージャーでのノードのテナントまたはセキュリティグループの割り当てへの変更が，グローバルマネージャーにどのように影響を及ぼすかを示しています。

表 14-6 リージョナルマネージャーでの設定変更がグローバルマネージャーに及ぼす影響

アクション	影響
リージョナルマネージャーで，ノードを別のテナントに割り当てる。	グローバルマネージャーのノードは，その別のテナントに割り当てられるように変更されます。テナント UUID がグローバルマネージャーにない場合は作成されます。
リージョナルマネージャーで，ノードを別のセキュリティグループに割り当てる。	グローバルマネージャーでは変更されません。NNMi 管理者は，その変更を手動で複製するように選択できます。
リージョナルマネージャーで，テナントの設定（名前，説明，または初期検出セキュリティグループ）を変更する。	グローバルマネージャーでは変更されません。NNMi 管理者は，その変更を手動で複製するように選択できます。
リージョナルマネージャーで，セキュリティグループの設定（名前または説明）を変更する。	グローバルマネージャーでは変更されません。NNMi 管理者は，その変更を手動で複製するように選択できます。

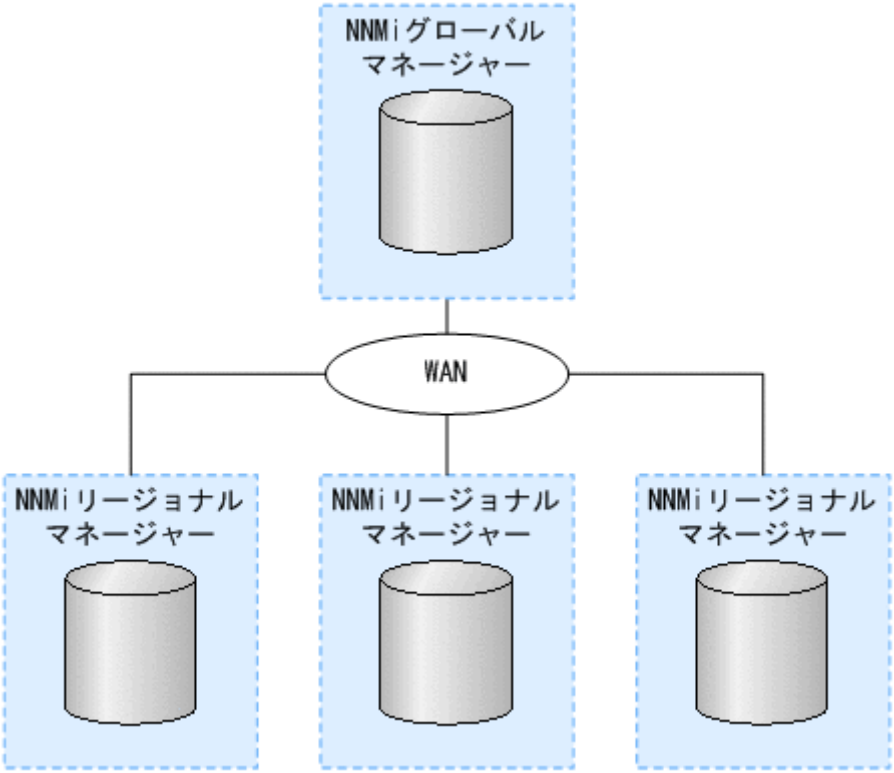
15 グローバルネットワーク管理

この章では，グローバルネットワークを管理する方法について説明します。

15.1　グローバルネットワーク管理の前提条件

グローバルネットワーク管理機能を利用する場合，グローバルネットワーク管理を構成するNNMi管理サーバーは同じバージョン・リビジョンである必要があります（リビジョンまで同じ必要があります）。修正版のバージョンは同じである必要はありません。

15.2　グローバルネットワーク管理の利点



NNMi を地理的位置が異なる複数の NNMi 管理サーバーに導入しているとします。各 NNMi 管理サーバーでは，検出と監視のニーズに合うように，ネットワークの検出および監視を行っています。こうした既存の NNMi 管理サーバーと設定を使用して，特定の NNMi 管理サーバーをグローバルマネージャーとして指定することで，新たな検出を追加したり監視の設定を変更したりせずに，集約したノードオブジェクトデータを表示できます。

NNMi グローバルネットワーク管理機能で，地理的位置が異なるネットワークを管理しながら，複数の NNMi 管理サーバーを連携させることができます。特定の NNMi 管理サーバーをグローバルマネージャーとして指定し，複数のリージョナルマネージャーを集約したノードオブジェクトデータを表示します。

NNMi グローバルネットワーク管理機能には，次の利点があります。

- グローバルマネージャーから見た，企業のネットワークの全体像を表示できます。
- 次のように容易に設定できます。
  - リージョナルマネージャーの管理者はそれぞれ，すべてのノードオブジェクトデータを指定するか，またはグローバルマネージャーレベルで参加する特定のノードグループを指定します。
  - 各グローバルマネージャーの管理者は，情報の提供を許可するリージョナルマネージャーを指定します。
- 各サーバーごとに，インシデントの生成と管理を行うことができます（各サーバーで使用可能なトポロジのコンテキスト内で生成されます）。

詳細については，NNMi ヘルプの「NNMi のグローバルネットワーク管理機能 (NNMi Advanced)」を参照してください。

動的ネットワークアドレス変換（NAT），動的ポートアドレス変換（PAT），または動的ネットワークアドレスおよびポート変換（NAPT）の各グループには，NNMi グローバルネットワーク管理設定全体で一意のテナントに加え，NNMi リージョナルマネージャーが必要です。詳細については，「13. NAT 環境の重複 IP アドレスの管理」および NNMi ヘルプを参照してください。

15.3 グローバルネットワーク管理の適用を検討する

15.3.1 複数サイトのネットワークを継続的に監視する

IT グループは，複数のサイトに配備されているネットワーク機器を週 7 日，24 時間体制で管理している場合，NNMi のグローバルネットワーク管理機能を使用すれば，トポロジとインシデントを集約して表示し，監視できるようになります。

15.3.2 重要なデバイスを選択して監視する

複数の場所に配備された重要デバイスのステータスとインシデントを，1 つの NNMi 管理サーバーで表示できる場合，リージョナルマネージャーに転送フィルタを設定します。このフィルタによって，リージョナルマネージャーからグローバルマネージャーに送信するノードオブジェクトデータを選択できます。例えば，リージョナルマネージャーに対し転送フィルタを設定して，重要デバイスに関する情報だけをグローバルマネージャーに転送するようにできます。

15.3.3 ライセンスを考慮する

グローバルマネージャーとして使用する NNMi 管理サーバーには，NNMi Advanced ライセンスを購入してインストールする必要があります。NNMi 管理サーバーをリージョナルマネージャーとして使用する場合は，NNMi Advanced ライセンスは必要ありません。

グローバルネットワーク管理機能を使用しながら，グローバルマネージャーに必要な新しいライセンスの数を抑えることができます。例えば，IT グループが複数のサイトに配備された重要な装置を監視する必要がある場合は，リージョナルマネージャーに転送フィルタを設定して，グローバルマネージャーに重要な装置に関する情報だけが転送されるようにできます。このようなフィルタ設定を使用することで，既存のグローバルマネージャーのライセンスを最大限に活用し，NNMi への投資をむだなく使用できます。

ライセンスを取得したノードの総数がグローバルマネージャーの NNMi Advanced ライセンスより多くなるように，リージョナルマネージャー用に NNMi ライセンスを増やします。グローバルマネージャーには，すべての領域のすべてのノードの完全なインベントリがありません。グローバルマネージャーをすべてのリージョナルマネージャーと同期させて，ライセンスが不十分だったために前回省略したノードを検索して作成する場合，グローバルマネージャーで十分な NNMi Advanced ライセンスを購入してインストールし，リージョナルマネージャーでインストールしたライセンス総数を上回るようにする必要があります。

十分なライセンスをインストールしたら，次のどちらかの方法で対処します。

- すべてのリージョナルマネージャーで設定されている，すべての再検出間隔の時間が経過して，すべての領域ですべてのノードが再検出されるまで待ちます。リージョナルマネージャーは，すべての領域ですべてのノードを再検出したら，再検出されたノードの情報をグローバルマネージャーに送信します。グローバルマネージャーはこのノード情報を受信し，各領域でノードごとにグローバルノードを作成します。
- 各リージョナルマネージャーで`nnmnoderediscover.ovpl -all`スクリプトを実行します。

メモ

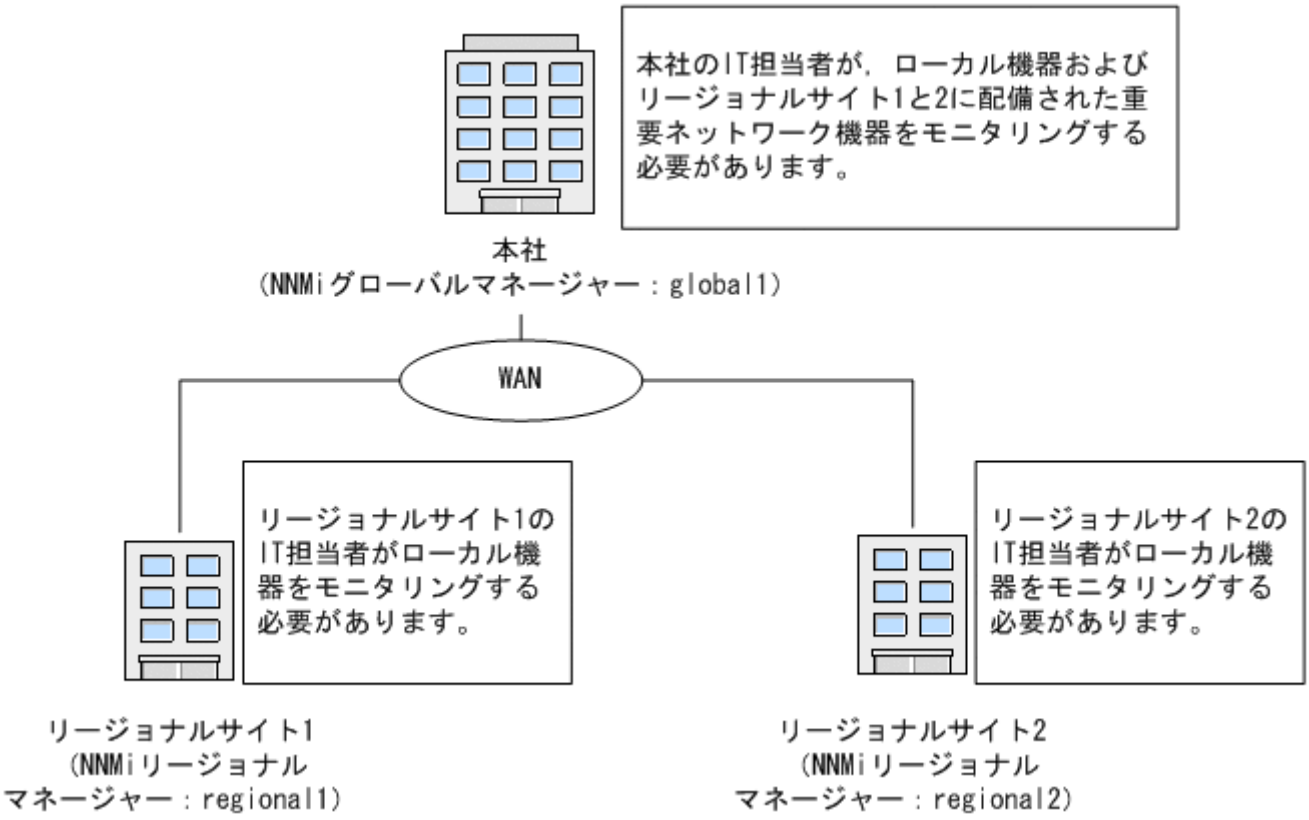
2 番目の方法では，ネットワーク上のトラフィックが増加し，NNMi マネージャーのセット全体から多くの NNMi リソースが消費されることにもなります。このオプションは，最初の NNMi 検出ほどリソースの多くを消費しませんが，最初の検出を実行することに似ています。最適な方法では，ある程度の時間をおくか，現在のリージョナルマネージャーの負荷が減って正常になるのを待ち，領域ごとに間隔をおいてスクリプトを実行してから，次のリージョナルマネージャーの再検出を始めます。

15.4　実践的なグローバルネットワーク管理の例

次の図を参照してください。地理的位置が異なる 2 つの運用サイトがあるとします。本社は，運用サイトとは別の地理的位置にあります。つまり，全部で 3 か所で NNMi 管理サーバーが機能しています。

本社の IT 担当者が，ローカルネットワーク機器およびリージョナルサイト 1 と 2 の両方に配備された重要ネットワーク機器を，ネットワークの観点から監視する必要があります。リージョナルサイト 1 と 2 両方の IT 担当者は，それぞれのサイトに配備されている重要なネットワーク機器を監視する必要があります。

図 15-1　ネットワークの例



15.4.1　要件のレビュー

本社，リージョナルサイト 1，リージョナルサイト 2 の NNMi 管理サーバーが，それぞれのサイトに配備された複数のルーターとスイッチを管理すると想定します。この例では，NNMi 管理サーバーをそれぞれ global1，regional1 および regional2 と呼びます。それぞれの場所に配備された重要なスイッチとルーターの検出と監視を行うように NNMi 管理サーバーを設定したとします。グローバルネットワーク管理機能を使用するために，これらのサイトにある NNMi 管理サーバーでの検出を再設定する必要はありません。

メモ

グローバルネットワーク管理機能の設定中，nnmbackup.ovpl スクリプトを使って 1 つの NNMi 管理サーバーをバックアップし，nnmrestore.ovpl スクリプトを使ってこのバックアップを第 2 の NNMi 管理サーバーに復元し，この両方の NNMi 管理サーバーをリージョナル NNMi 管理サーバーに接続することはしないでください。ある NNMi 管理サーバーから 2 番目の NNMi 管理サーバーにバックアップデータを配置すると，これらの両方のサーバーに同じデータベース UUID が存在することになります。NNMi を第 2 の NNMi 管理サーバーに復元したあと，元の NNMi 管理サーバーから NNMi をアンインストールする必要があります。

本社 IT グループでは，リージョナルサイト 1 と 2 に配備された重要な機器だけの監視を行い，ほかのデバイスの管理はしない予定です。次の表に，監視のニーズをまとめます。

表 15-1　グローバルネットワーク管理のネットワーク要件

サイト	NNMi 管理サーバー	重要なスイッチ	管理するリージョナル機器
本社	global1	15 台の HP ProCurve 2620 Switch	各リージョナルサイトの HP ProCurve 2620 Switch すべて
リージョナルサイト 1	regional1	15 台の HP ProCurve 2620 Switch	該当なし
リージョナルサイト 2	regional2	15 台の HP ProCurve 2620 Switch	該当なし

要約すると，NNMi 管理サーバー global1 が本社を監視し，NNMi 管理サーバー regional1 と regional2 が，各リージョナルサイトを監視しています。リージョナルサイト 1 と 2 に配備された HP ProCurve 2620 Switch のインシデントとデバイス情報を，本社で表示する必要があります。この例では，regional1 と regional2 の両方で，リージョナルサイト 1 に配備された複数の共通スイッチを管理しています。

(1)　リージョナルマネージャーとグローバルマネージャーの接続

グローバルネットワーク管理接続を設定するときに，次の情報を考慮します。

- NNMi では，リージョナルマネージャーと通信する 1 つ以上のグローバルマネージャーを設定できます。例えば，regional1 と通信するために第 2 のグローバルマネージャー，global2 が必要な場合，NNMi では，regional1 と通信する global1 と global2 の両方を設定できます。詳細については，リリースノートを参照してください。
- グローバルネットワーク管理は，1 つの接続レイヤーで動作します。例えば，この章の例では，1 つの接続レイヤー，regional1 と通信する global1 と regional2 と通信する global1 について検討します。NNMi は，複数の接続レベルを設定しないでください。例えば，global1 は regional1 と通信し，かつ regional1 が regional2 と通信するようには設定しないでください。グローバルネットワーク管理機能は，この 3 つのレイヤー設定用に設計されていません。

- 2 つの NNMi 管理サーバーは，相互に両方向に通信する設定にはしないでください。例えば，global1 が regional1 と通信し，かつ regional1 が global1 と通信するようには設定しないでください。

15.4.2　初期準備

(1)　ポート可用性：ファイアウォールの設定

グローバルネットワーク管理機能が正しく機能するためには，global1 から regional1 と regional2 への TCP アクセス用に，特定のウェルノウンポートが開いているかどうかを確認する必要があります。NNMi インストールスクリプトでは，デフォルトとしてポート 20480，20481 を設定します。ただし，インストール中にこの値は変更できます。

> **メモ**
>
> ここで説明した例では，global1 が regional1 と regional2 への TCP アクセスを確立します。ファイアウォールは，一般的に接続を開始するサーバーに基づいて設定されます。global1 が regional1 と regional2 への接続を確立すると，トラフィックは両方向に流れます。

現在の値を確認したりポート設定を変更したりするには，次のファイルを編集します。

- Windows：`%NNM_CONF%¥nnm¥props¥nms-local.properties`
- Linux：`$NNM_CONF/nnm/props/nms-local.properties`

次の表に，アクセス可能にしておく必要があるウェルノウンポートを示します。

表 15-2　アクセス可能にしておく必要があるソケット

セキュリティ	パラメーター	TCP ポート
非 SSL	nmsas.server.port.web.http	20480
	nmsas.server.port.hq	4457
SSL	nmsas.server.port.web.https	20481
	nmsas.server.port.hq.ssl	4459

(2)　証明書の設定

global1 と 2 つのリージョナル NNMi 管理サーバー（regional1 と regional2）間で安全な通信プロトコルによるグローバルネットワーク管理機能を使用する場合は，証明書を設定する必要があります。NNMi のインストール中，NNMi インストールスクリプトでは，ほかのエンティティに対して自身を識別できるよう，NNMi 管理サーバーに自己署名証明書を作成します。使用する NNMi 管理サーバーには，正しい証明書を持つグローバルネットワーク管理機能を設定する必要があります。「10.3.7　グローバルネットワーク管理環境での証明書の使用」に示した手順を実行してください。

バージョン 11-50 にアップグレードされた NNMi 管理サーバー

現在使用している複数の NNMi 管理サーバーのうち，一部が以前のバージョンの NNMi から NNMi 11-50 にアップグレードしたもので，一部が新たにインストールした NNMi 11-50 インスタンスである場合は，GNM を設定する前に追加的な設定タスクを実行する必要があります。

NNMi 11-50 より前のバージョンでは，NNMi は証明書を保存するために Java KeyStore (JKS) リポジトリを提供していました。NNMi 11-50 では，証明書を保存するために Public Key Cryptography Standards (PKCS) #12 リポジトリが導入されています。NNMi 11-50 の新しいインスタンスをシステムにインストールすると，新しい PKCS #12 ファイルベースの証明書管理方法を利用できます。

ただし，古いバージョンの NNMi をバージョン 11-50 にアップグレードした場合，PKCS #12 ファイルベースの証明書管理はすぐには利用できず，NNMi では証明書管理に JKS リポジトリが引き続き使用されます。

このような環境で GNM を設定する場合は，事前に「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従い，アップグレードしたすべての NNMi 管理サーバーで PKCS #12 ファイルベースの証明書管理方法を使用するように設定を行ってください。

(3)　NNMi 管理サーバー規模の考慮事項

この例では，グローバルネットワーク管理設定で既存の NNMi 管理サーバーを使用することを想定しています。

NNMi のインストールが必要となるサーバーのサイズに関する具体的な情報については，リリースノートを参照してください。

(4)　システムクロックの同期化

global1，regional1，および regional2 サーバーをグローバルネットワーク管理設定に接続する前に，これらの NNMi 管理サーバークロックを同期化することが重要です。グローバルネットワーク管理（グローバルマネージャーとリージョナルマネージャー）やシングルサインオン（SSO）に属するネットワーク環境内のすべての NNMi 管理サーバーは，それぞれの内部タイムクロックを世界標準時で同期化する必要があります。例えば，Linux ツールの Network Time Protocol Daemon（NTPD）や使用可能な Windows オペレーティングシステムツールなどの時刻の同期プログラムを使用します。詳細については，NNMi ヘルプの「クロック同期化の問題 (SSO / グローバルネットワーク管理)」または「グローバルネットワーク管理をトラブルシューティングする」と「15.11.2　クロック同期」を参照してください。

メモ

サーバークロック同期の問題など，リージョナルマネージャーとの接続に問題がある場合，NNMi では NNMi コンソールの下部に警告メッセージが表示されます。

(5) グローバルネットワーク管理で自己署名証明書を使用する場合のアプリケーションフェイルオーバー機能の使用法

アプリケーションフェイルオーバー設定で，自己署名証明書を使用したグローバルネットワーク管理機能を使用する場合は，追加の手順を実行する必要があります。

(6) グローバルネットワーク管理での自己署名証明書の使用法

グローバルネットワーク管理機能で自己署名証明書を使用する場合は，追加の手順を実行する必要があります。「10.3.7 グローバルネットワーク管理環境での証明書の使用」を参照してください。

(7) グローバルネットワーク管理での認証機関の使用法

グローバルネットワーク管理機能で認証機関を使用する場合は，追加の手順を実行する必要があります。「10.3.7 グローバルネットワーク管理環境での証明書の使用」を参照してください。

(8) 監視する重要な機器の一覧作成

各リージョナルマネージャーによって管理され，グローバルマネージャーからモニタリングされる機器のリストを作成します。例えば，global1 からモニタリングされる regional1 と regional2 の管理対象機器リストを作成します。この情報を転送フィルターで使用します。詳細については，「15.5 リージョナルマネージャーで転送フィルタを設定する」を参照してください。

regional1 と regional2 から global1 に転送する情報を制限した場合に得られる結果については，慎重に考慮する必要があります。計画を立てるときに，次の点を考慮してください。

- global1 で完全な分析を行って正確なインシデントを生成するには，regional1 と regional2 から得られる完全なトポロジが必要になるため，除外するデバイスが多くなり過ぎないように注意します。
- 重要ではないデバイスを除外すると，global1 のシステムパフォーマンスコストを節約できます。
- 重要ではないデバイスを除外すると，ソリューションの全体的な拡張性が改善され，NNMi で必要となるネットワークトラフィックを削減できます。

(9) グローバルマネージャーとリージョナルマネージャーの管理ドメインの検討

リージョナルマネージャーからグローバルマネージャーに転送する情報を決定するために，グローバルマネージャーとリージョナルマネージャーの管理ドメインを検討します。

この例では，NNMi 管理サーバー global1，regional1，および regional2 は，独自のノードセットを管理しています。この例では，あとで regional1 と regional2 から global1 に，それぞれが管理する機器に関する情報を転送するよう設定します。

次の手順に従って，global1，regional1，および regional2 が現在監視している機器を確認します。機器を確認しておくと，regional1 と regional2 から global1 に転送する重要な機器を選択するときに役立ちます。

この例では，次の手順を実行してこの情報を確認します。

1. ブラウザで global1 の NNMi コンソールを指定する。
2. サインインする。
3. **[インベントリ]** ワークスペースをクリックする。
4. このワークスペースで global1 が現在監視していて検出されたインベントリを確認できる。
5. ブラウザで regional1 の NNMi コンソールを指定する。
6. サインインする。
7. **[インベントリ]** ワークスペースをクリックする。
8. regional1 が監視しているノードを確認し，global1 で監視するデバイスの一覧を作成する。
9. ブラウザで regional2 の NNMi コンソールを指定する。
10. サインインする。
11. **[インベントリ]** ワークスペースをクリックする。
12. regional2 が監視しているノードを確認し，global1 で監視するデバイスの一覧を作成する。

(10) NNMi ヘルプトピックの確認

グローバルネットワーク管理に関するすべてのヘルプトピックを確認するには，次の手順を実行します。

1. NNMi ヘルプで，**[検索]** をクリックする。
2. **[検索]** フィールドに「**グローバルネットワーク管理**」と入力する。
3. **[検索]** をクリックする。

この検索によって，グローバルネットワーク管理に関連する 50 以上のトピックが見つかります。

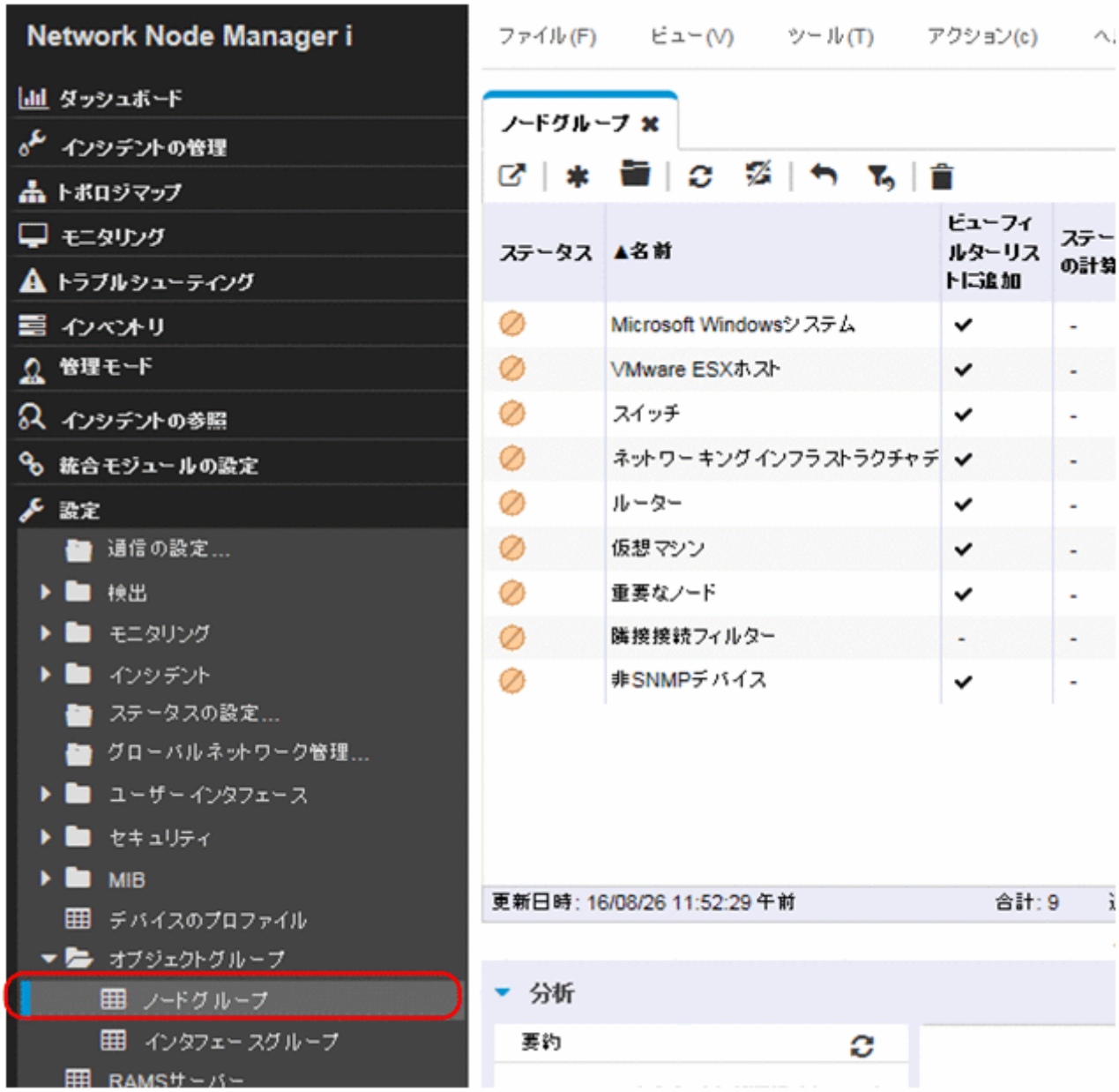
15.5　リージョナルマネージャーで転送フィルタを設定する

この例では，global1 は regional1 と regional2 の両方と通信します。グローバルマネージャー global1 がリージョナルマネージャー regional1 と regional2 から受け取るノードオブジェクトデータを制御するには，regional1 と regional2 の両方で転送フィルタを設定する必要があります。

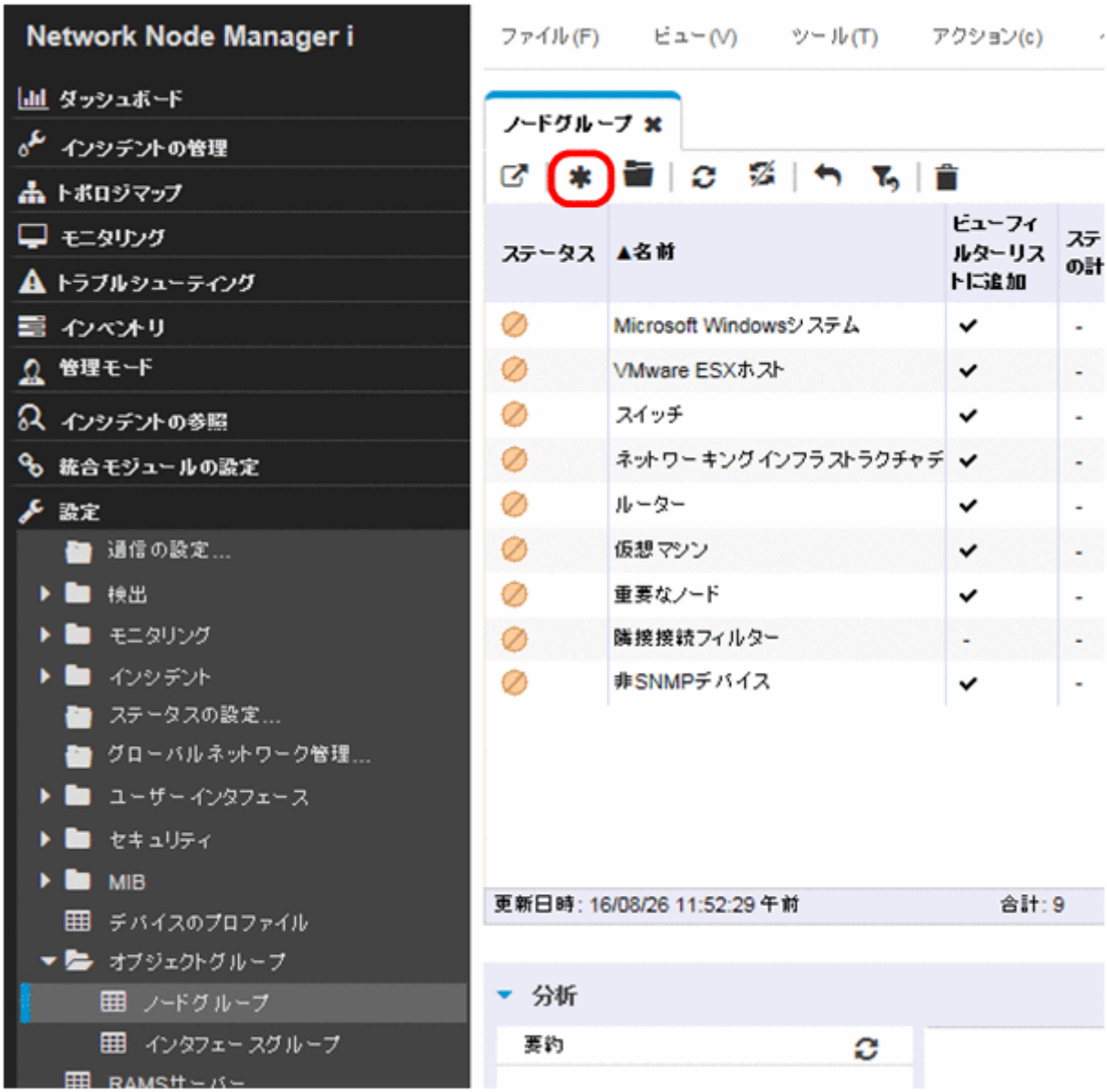
15.5.1　転送されるノードを制限する転送フィルタを設定する

この例では，HP ProCurve 2620 Switch のノード情報だけを regional1 から global1 に転送できるノードグループを作成します。新しいノードグループを作成し，グループに制限を設定するには，次の手順を実行します。

1. NNMi コンソールの regional1 の［設定］＞［オブジェクトグループ］から，［ノードグループ］をクリックする。



2. [新規作成] をクリックする。

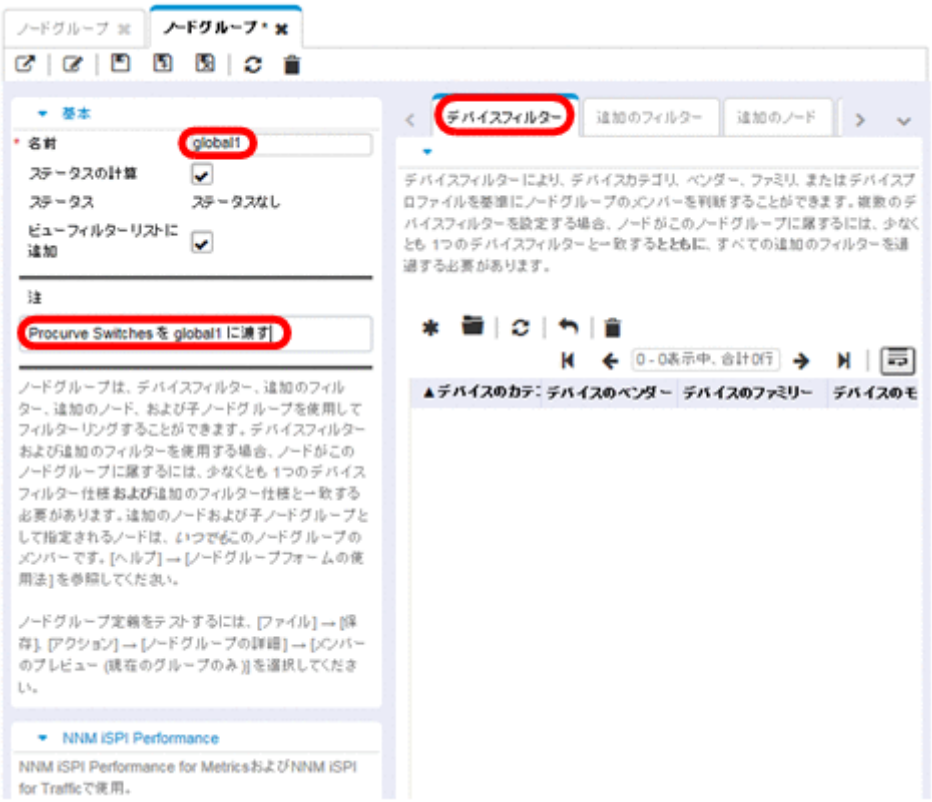


メモ

この例では，ノードフィルタの新規作成し，そのフィルタを使用して regional1 と regional2 の転送フィルタを作成する方法を説明していますが，既存のフィルタを使用して，リージョナル NNMi 管理サーバーからグローバル NNMi 管理サーバーへの転送フィルタを設定することもできます。

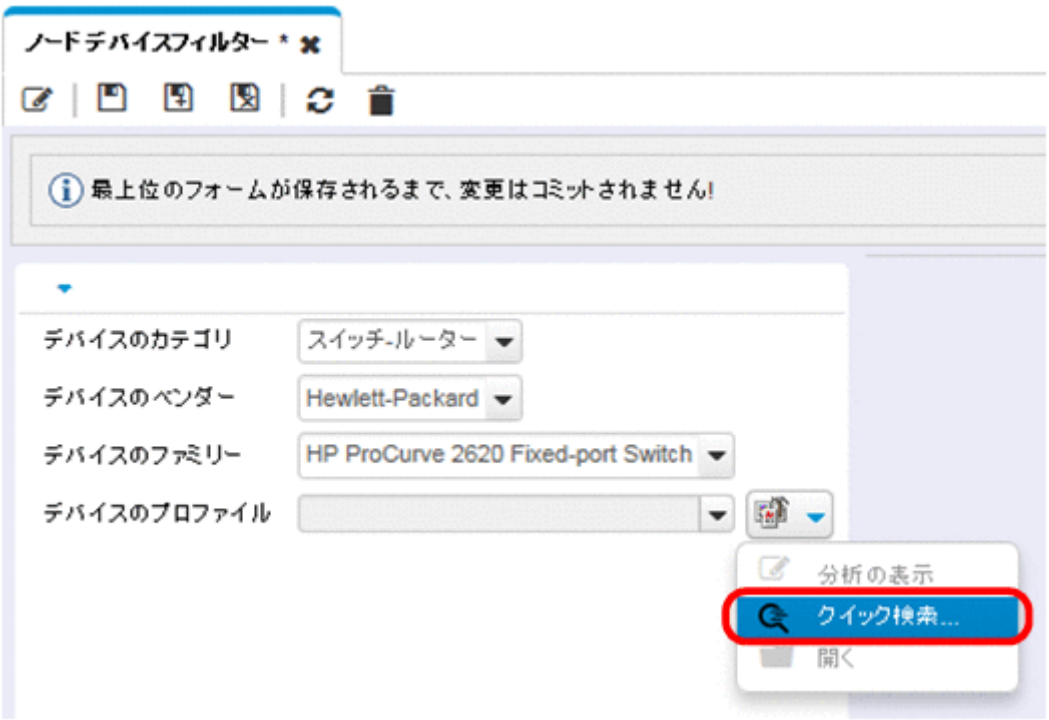
独自のデバイスもフィルタも含まれていないコンテナノードグループを作成して，子ノードグループを指定できます。この方法を使用すると，1 つのコンテナノードグループを使用して，ノードオブジェクトデータをグローバル NNMi 管理サーバーに転送できます。

3. フィルタ名として名前フィールドに`global1` と入力し，［注］フィールドに作成するフィルタの説明を入力する。

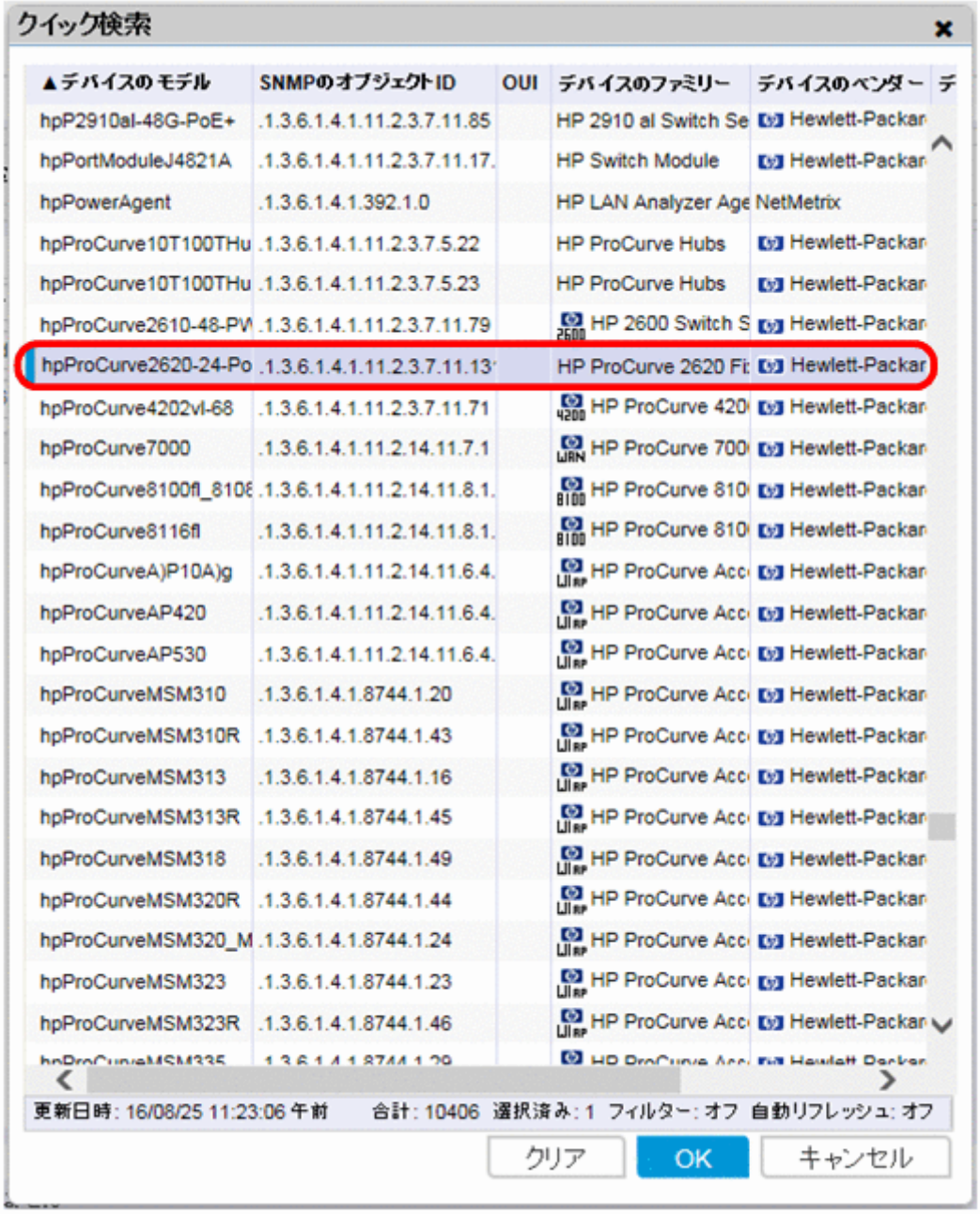


4. [デバイスフィルター] タブで [新規作成] アイコンをクリックして，[ノードデバイスフィルター] フォームを開く。

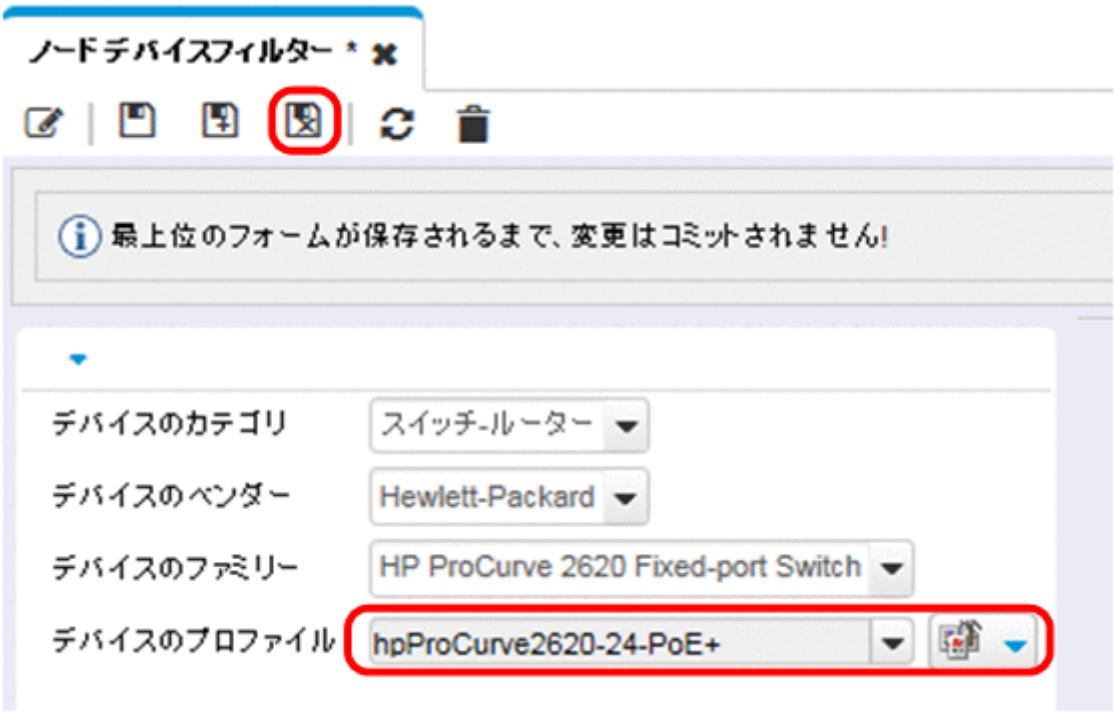
5. プルダウンメニューを使用して，［デバイスのカテゴリ］では［スイッチ-ルーター］，［デバイスのベンダー］では［Hewlett-Packard］，および［デバイスのファミリー］では［HP ProCurve 2620 Fixed-port Switch］を選択する。
6. プルダウンメニューから，［クイック検索］をクリックして，［デバイスのプロファイル］フォームを開く。



7. HP ProCurve 2620 Switch のプロファイルを検索して選択し，［OK］をクリックする。

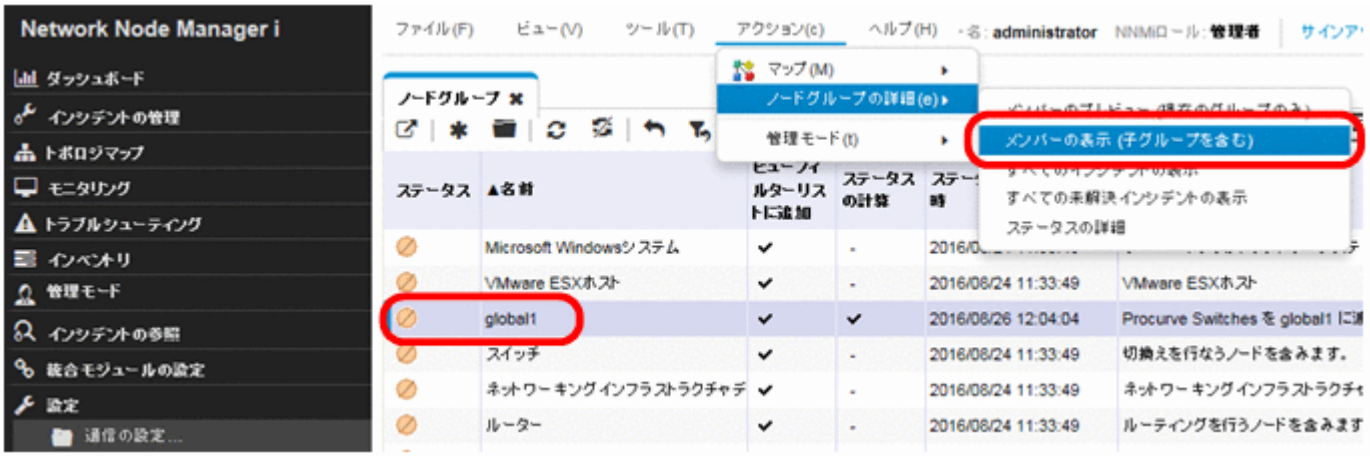


8. 設定フォームごとに，［保存して閉じる］をクリックする。

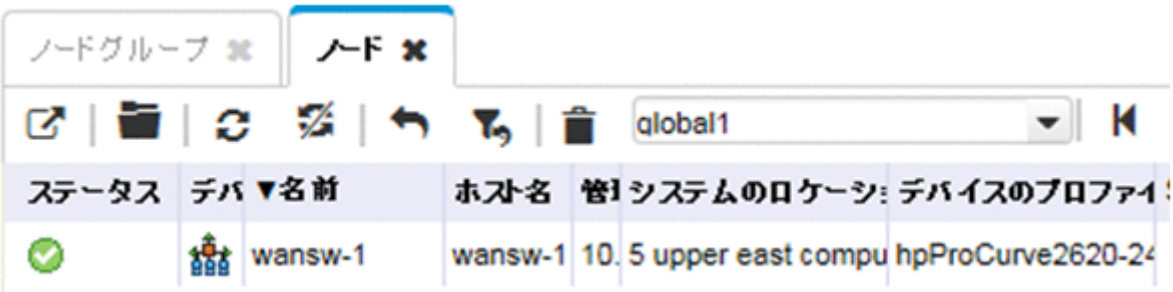


9. このフィルタをテストするため，［global1］を選択する。

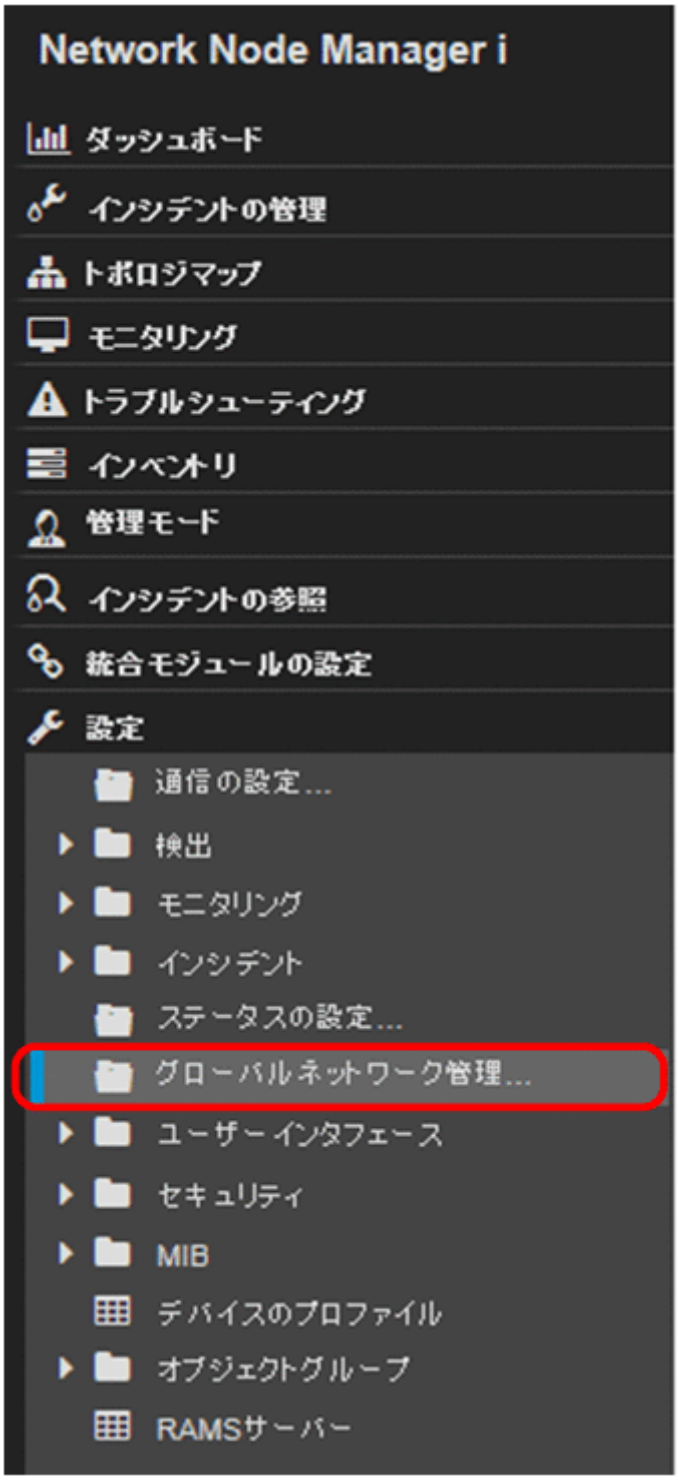
10. ［アクション］＞［ノードグループの詳細］メニューから，［メンバーの表示］をクリックする。



11. NNMi ではすでに HP ProCurve 2620 Switch が 1 つ検出されている。これは，作成したフィルタが，設定した特定のスイッチモデルを検索していることを示している。次のステップでは，今作成したこのノードフィルタを使用して転送フィルタを設定する。



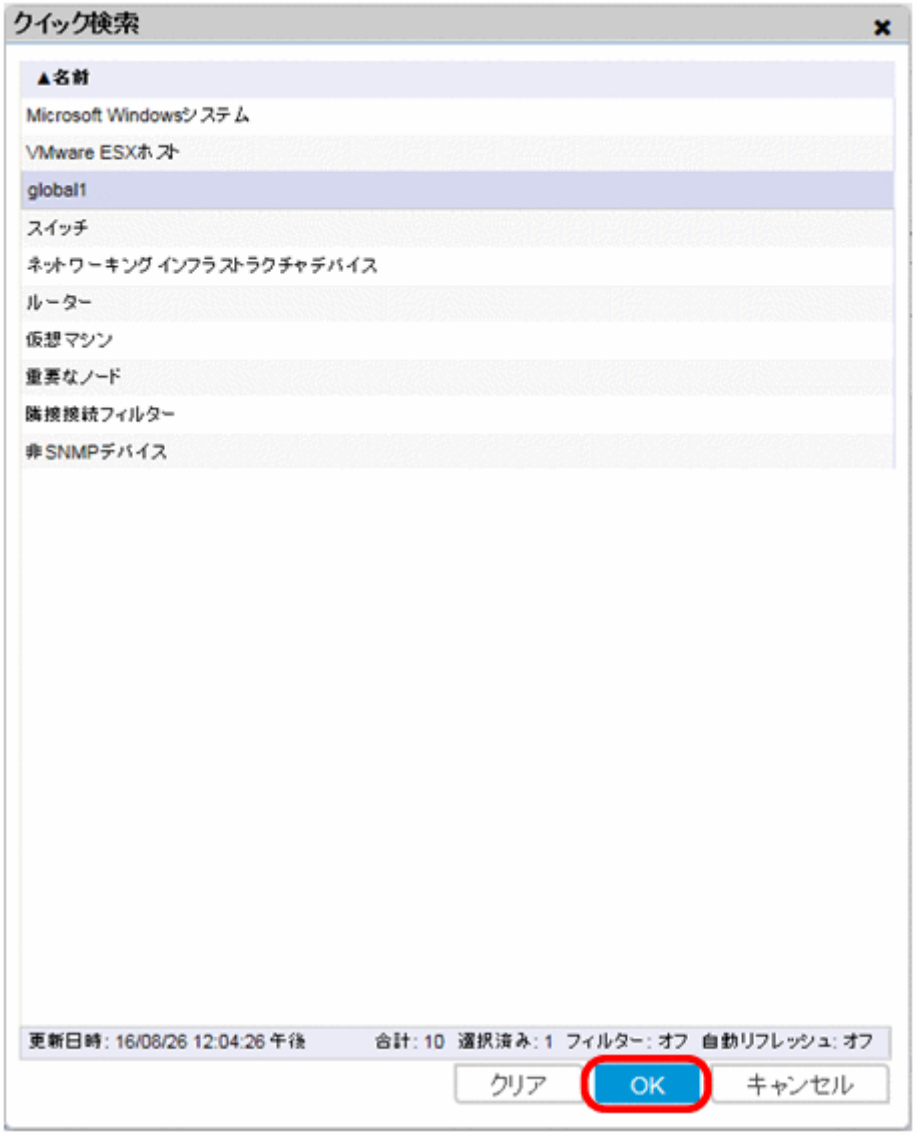
12. NNMi コンソールの regional1 の［設定］ワークスペースから，［グローバルネットワーク管理］をクリックする。



13. ［転送フィルター］タブをクリックする。

14. [クイック検索] をクリックする。

15. [global1] フィルタを選択し，[OK] をクリックする。



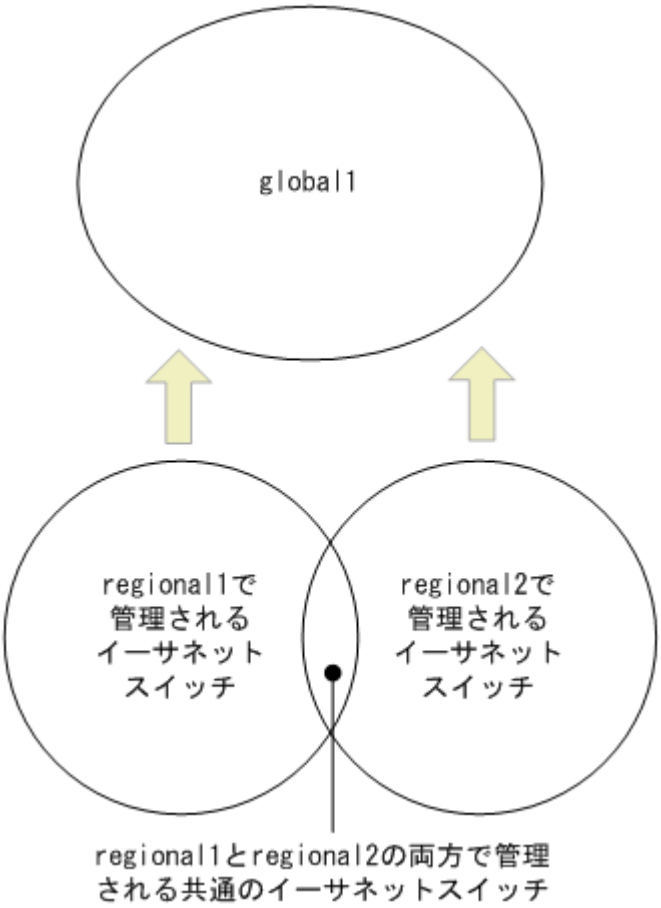
16. [保存して閉じる] をクリックする。

これで，regional1 の転送フィルターの設定作業は完了です。regional2 についても手順 1.から手順 16.を実行し，「15.6　グローバルマネージャーとリージョナルマネージャーを接続する」の説明に従って，global1 を regional1 と regional2 に接続します。

15.6 グローバルマネージャーとリージョナルマネージャーを接続する

この例では，regional1 と regional2 の両方で，共通のスイッチを複数管理します。

この共通のスイッチ情報を regional1 か global1 に転送するには，必要な接続を設定する必要があります。



そのためには，global1 を先に regional1 に接続してから regional2 に接続する必要があります。この接続順によって，global1 は regional1 をこれらの共通スイッチの監視を行う NNMi 管理サーバーであると見なし，regional2 から受け取るこれらの共通スイッチに関する情報を無視します。

メモ

この機能の動作を理解するには，まずは小さな規模で使用してから，それぞれのネットワーク管理ニーズに合わせて拡張することを推奨します。

global1 を先に regional1 に接続し，次に regional2 に接続するには，次の手順を実行します。

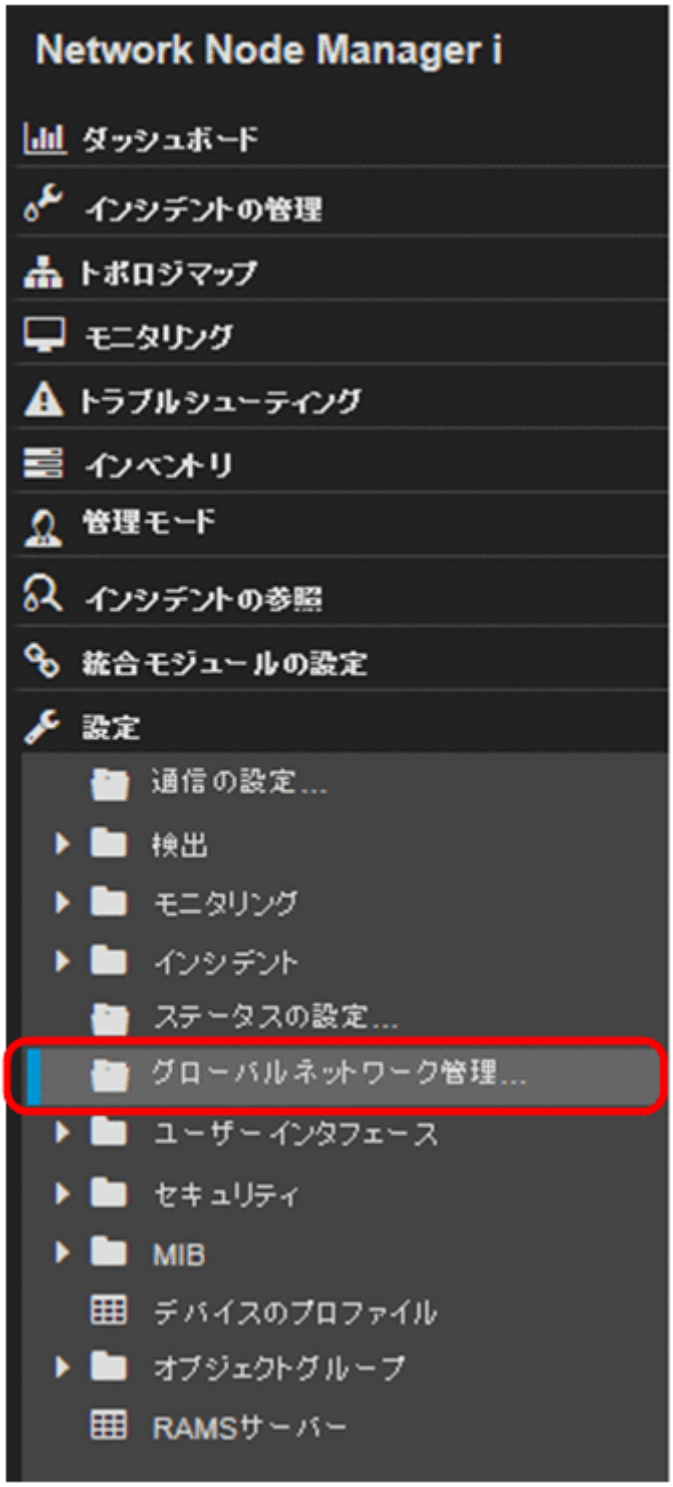
1. 先ず，NNMi 管理サーバーのクロックを global1，regional1，および regional2 と同期してから，グローバルネットワーク管理設定内のこれらのサーバーを接続する。

   詳細については，NNMi ヘルプの「クロック同期化の問題 (SSO / グローバルネットワーク管理)」を参照してください。

> **メモ**
>
> サーバークロック同期の問題など，リージョナルマネージャーとの接続に問題がある場合は，NNMi では警告メッセージが表示されます。

2. global1 から regional1 への接続を設定する。

   a　global1 の NNMi コンソールで，［設定］ワークスペースの［グローバルネットワーク管理］をクリックします。



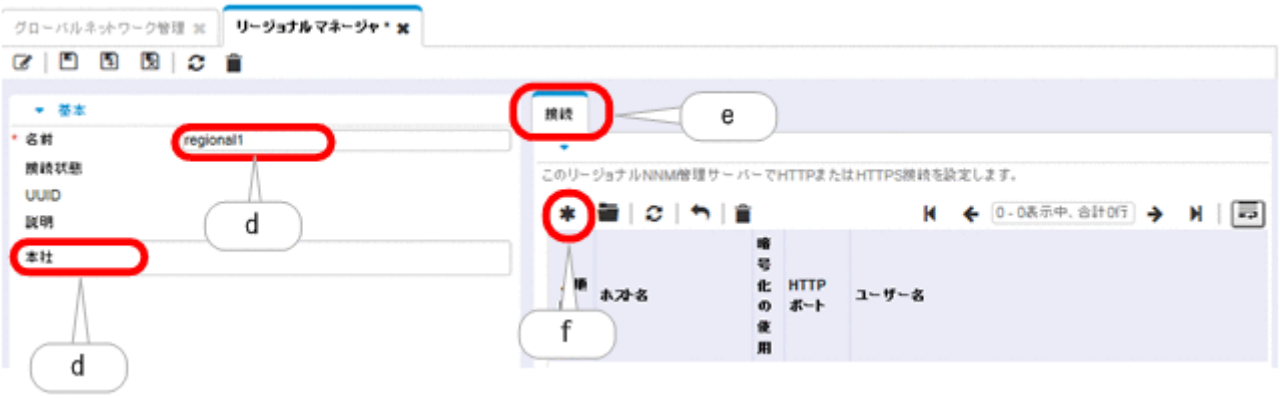
b ［リージョナルマネージャ接続］をクリックします。

c ［新規作成］アイコンをクリックして，リージョナルマネージャーを新規作成します。

d regional1 の名前と説明情報を追加します。

e ［接続］タブをクリックします。

f ［新規作成］アイコンをクリックします。



g regional1 の接続情報を追加します。

> **メモ**
>
> このフォームの実行に関する個別の情報については，NNMi ヘルプの「グローバルマネージャー: リージョナルマネージャーに接続する」を参照してください。

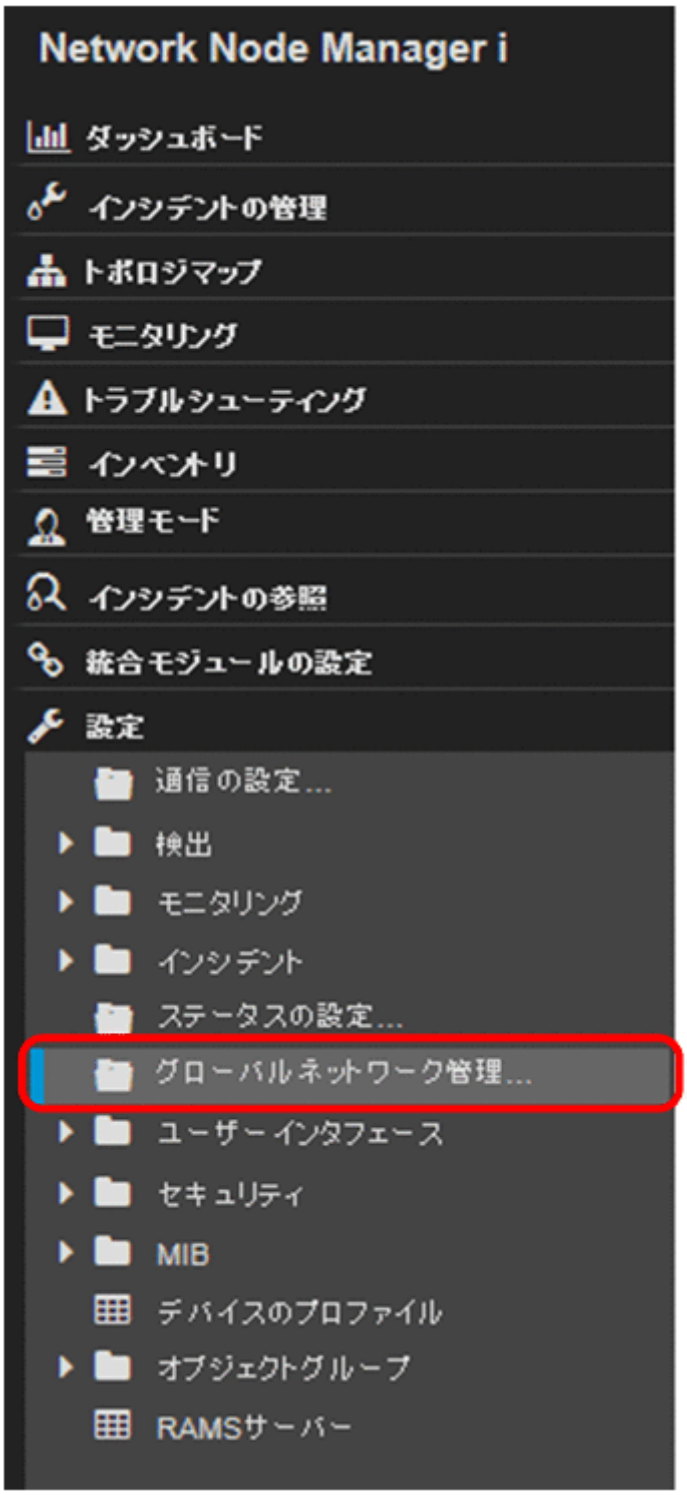
h 各設定フォームで［保存して閉じる］をクリックし，変更を保存します。

3. global1 から regional2 への接続を確立するため，手順 2.の a から h までを実行する。

15.7　global1 から regional1 と regional2 への接続ステータスを確認する

global1 から regional1 および regional2 への接続の状態を確認するには，次の手順を実行します。

1. global1 の NNMi コンソールで，［設定］ワークスペースの［グローバルネットワーク管理］をクリックする。



2. ［リージョナルマネージャ接続］タブをクリックする。

3. regional1 と regional2 の接続ステータスを確認する。

   [接続済み] と表示されたら，正しく機能していることを意味します。

   詳細については，NNMi ヘルプの「リージョナルマネージャーとの接続状態を確認する」を参照してください。

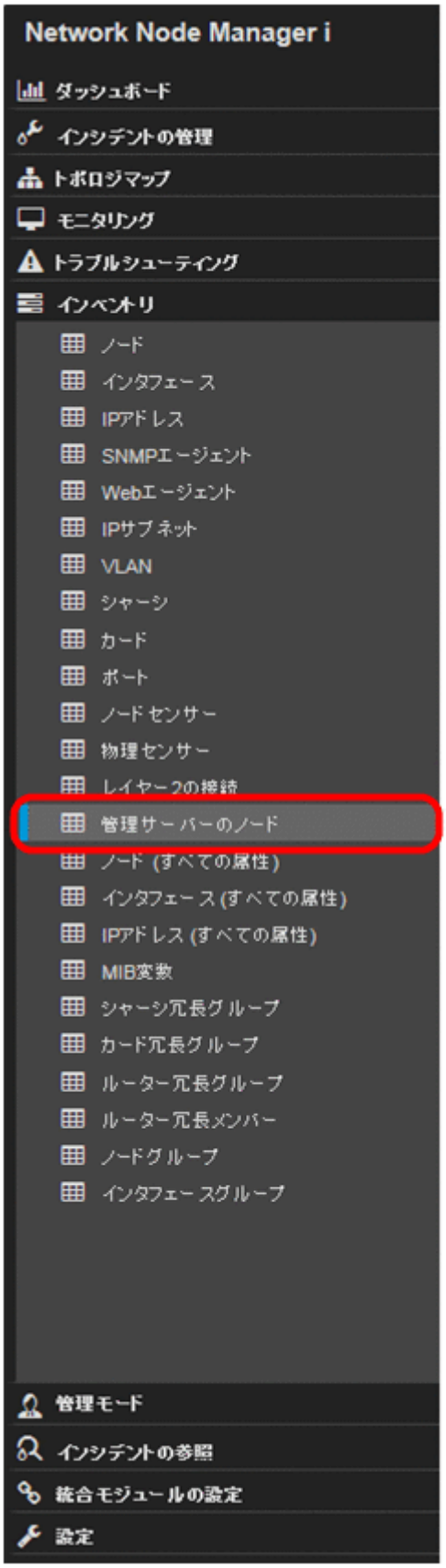
NNMi が検出を完了するまで，次のセクションには進まないでください。詳細については，「3.3.3　検出の進行状況を確認する」を参照してください。

15.8　global1 のインベントリを確認する

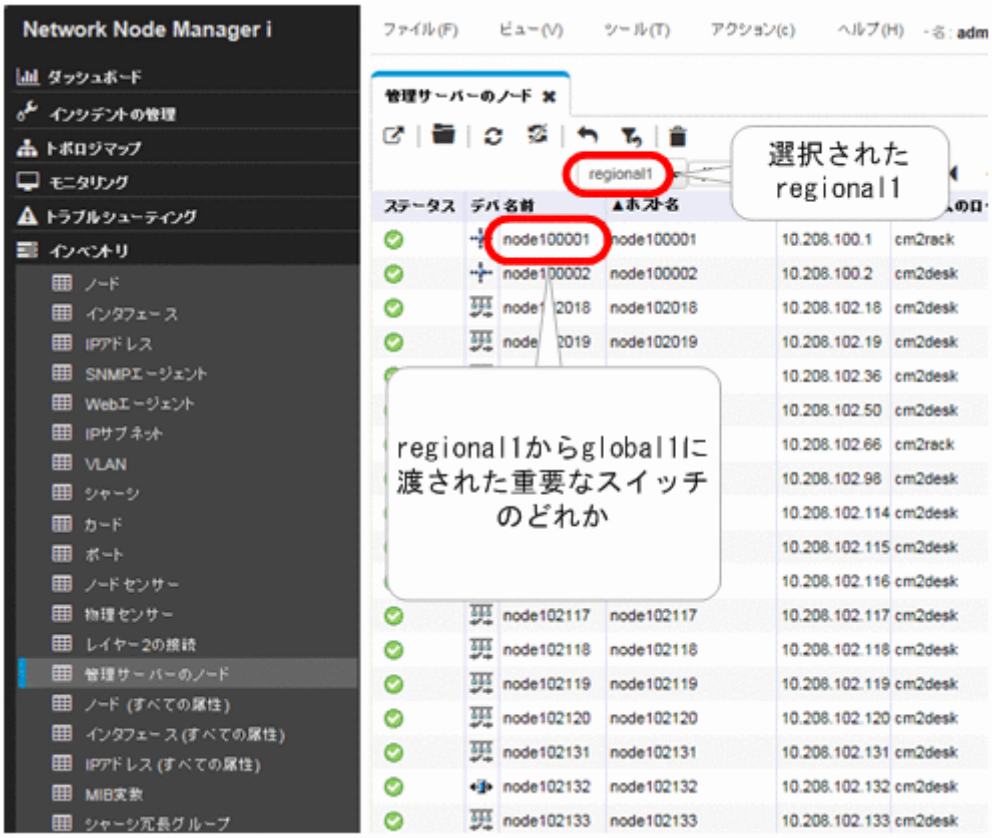
NNMi が検出を完了するまで，このセクションは実行しないでください。詳細については，「3.3.3　検出の進行状況を確認する」を参照してください。

global1 に転送されるノード情報 regional1 を表示するには，次の手順を実行します。

1. ［インベントリ］ワークスペースに配置されている［管理サーバーのノード］フォームに，global1 の NNMi コンソールから移動する。



2. スイッチ node102130 に関する情報が regional1 から global1 に転送されたと仮定する。
   regional1 を選択すると，インベントリは次のように表示されます。



手順 1.から手順 2.を実行して，接続されているほかのリージョナルマネージャーから global1 に転送されたデバイスインベントリも表示します。

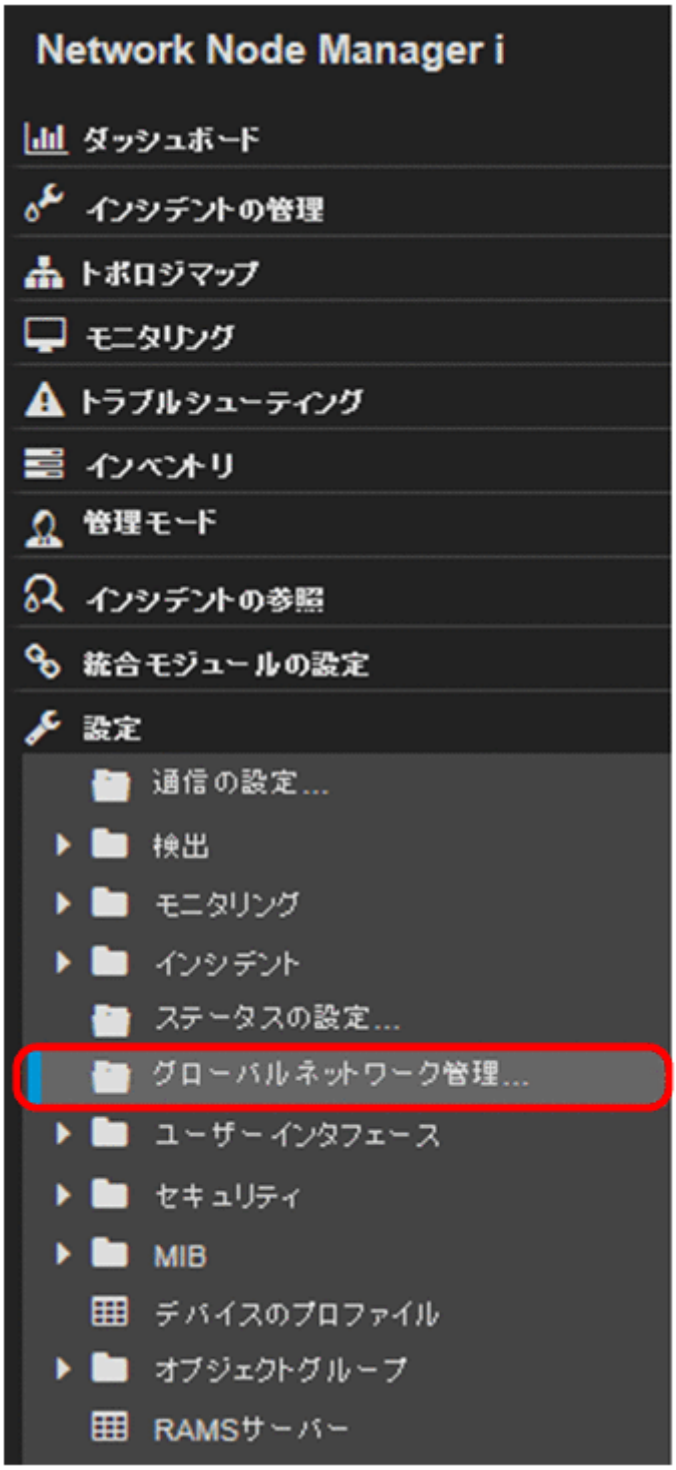
15.9 global1 と regional1 との通信を切断する

（一時的または完全に）グローバルマネージャー（global1 など）をシャットダウンするには，グローバルマネージャーとリージョナルマネージャー間の通信を切断する必要があります。

この例では，global1 では対 regional1 のサブスクリプションがまだアクティブであると想定します。

global1 と regional1 間の通信を切断するには，次の手順を実行します。

1. global1 の NNMi コンソールで，［設定］ワークスペースの［グローバルネットワーク管理］をクリックする。



2. [リージョナルマネージャ接続] をクリックする。

3. 接続状態が［接続されています］であることを確認する。

接続状態が［接続されています］ではない場合，処理を続行する前に，NNMi ヘルプの「グローバルネットワーク管理をトラブルシューティングする」を参照して問題を診断します。

4. regional1 を選択して［開く］アイコンをクリックする。

5. ［接続］をクリックして［regional1.xx.x.xx］を選択してから［削除］アイコンをクリックする。

6. ［保存して閉じる］をクリックする。

7. ［リージョナルマネージャ接続］タブでは，regional1 の［名前］属性に注意する（大文字小文字は区別される）。

   手順 9.で，この［名前］属性が必要になります。

8. ［保存して閉じる］をクリックする。

9. global1 のコマンドラインで次のコマンドを入力する。

```
nnmnodedelete.ovpl -rm regional1 -u NNMiadminUserName -p
NNMiadminPassword
```

   -rm には，手順 7.で確認した名前を指定します。

10. これらのコマンドで，regional1 から転送されたノードレコードを global1 から削除する。

    コマンドでは，regional1 から global1 に転送されたノードに関連するインシデントも閉じます。詳細については，NNMi ヘルプの「リージョナルマネージャーとの接続を解除する」を参照してください。

11. regional1 の設定レコードを削除するには，次を実行する。

    a ［設定］ワークスペースをクリックします。

    b ［グローバルネットワーク管理］フォームを選択します。

    c ［リージョナルマネージャ接続］タブを選択します。

    d regional1 を選択して［削除］アイコンをクリックします。

    e ［保存して閉じる］をクリックして削除を保存します。

15.10 グローバルネットワーク管理の追加情報

15.10.1 検出とデータの同期化

ネットワーク管理者がネットワーク上のデバイスの追加，削除，または変更を行うと，regional1 や regional2 などのリージョナルサーバーはそうした変更を検出して，この章の例での global1 などのグローバルサーバーを更新します。regional1 と regional2 では，これらが管理するノードの管理モードに対して管理者が行う変更についても global1 に通知します。

メモ

整合性を保つため，regional1 と regional2 はデバイスの状態の変化を検出すると，global1 を継続的に更新するので，グローバルサーバーとリージョナルサーバーの両方でノードの状態が同じに保たれます。

regional1 または regional2 が管理するノードに関する情報を global1 が要求するたびに，regional1 または regional2 は要求された情報を global1 に返します。global1 からノードに直接要求することはありません。global1 が検出を実行するとき，デバイスに対する SNMP クエリーは重複しません。

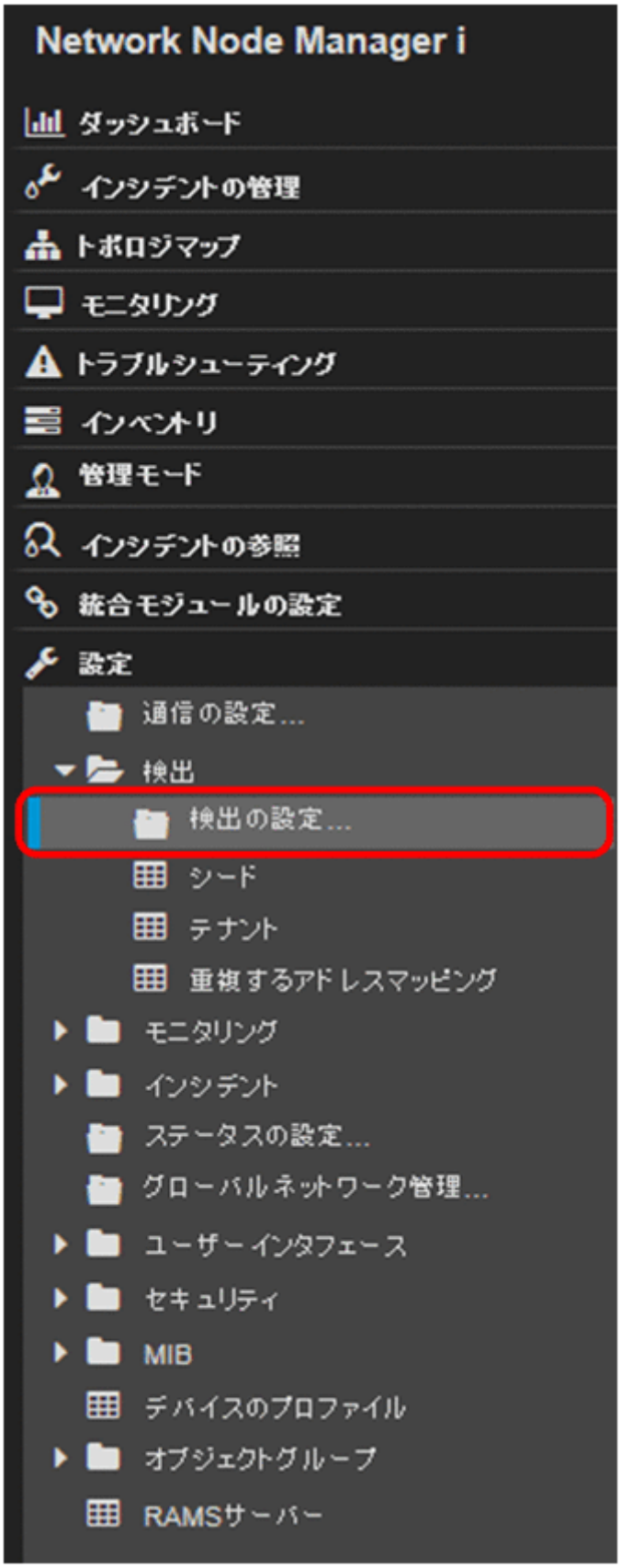
global1 は，regional1 または regional2 が検出を完了するたびに，regional1 と regional2 を同期します。NNMi は FDB（転送データベース）データを使用して，レイヤー 2 接続を計算します。FDB データは非常にダイナミックなもので，特に，1 つのグローバルサーバーに複数のリージョナルサーバーが接続しているような場合には，検出するごとに大きく異なります。

メモ

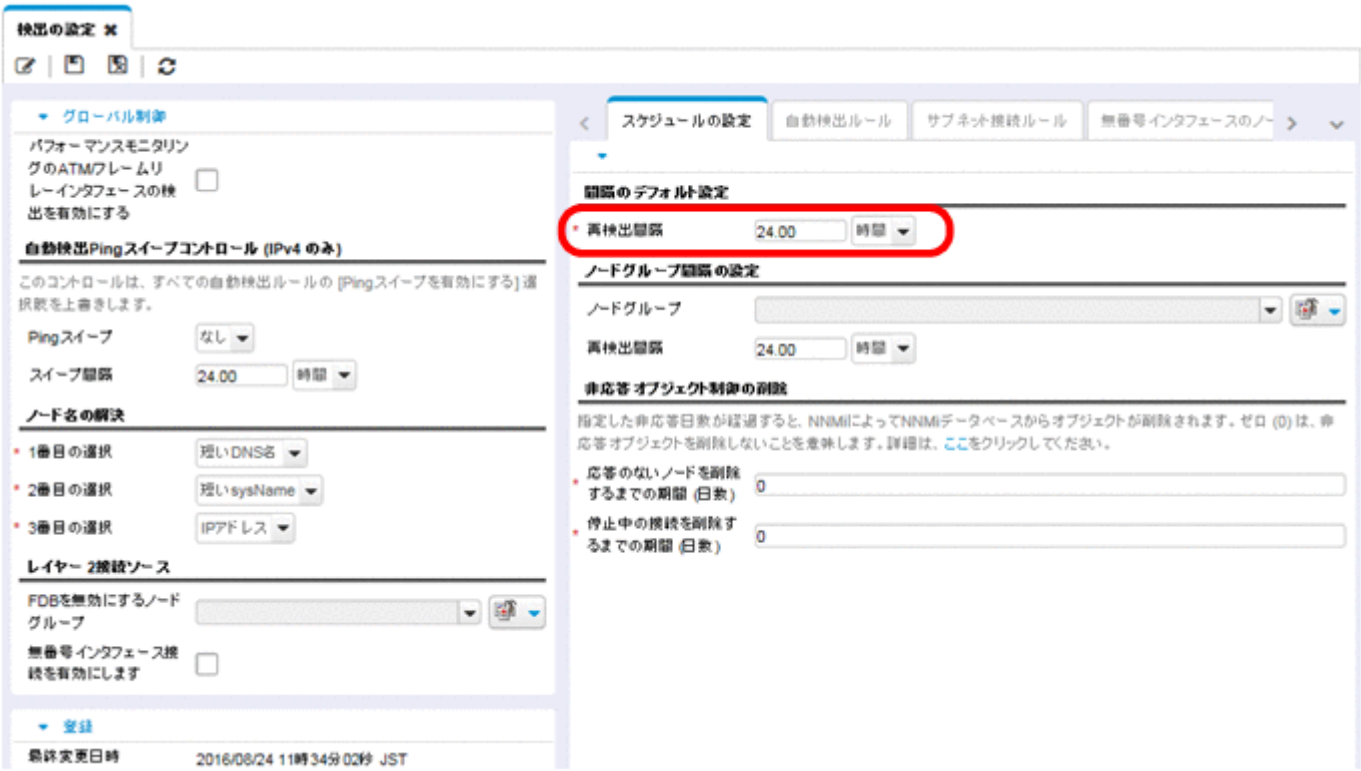
ユーザーが修正した属性やアプリケーションが修正した属性に対する変更は，グローバルサーバーでは同期中に更新されません。

[再検出間隔] は，各リージョナルサーバーで調整でき，global1 とリージョナルマネージャーとの間の検出の精度を変更できます。**[再検出間隔]** が短くなるほど，検出の精度が上がり，NNMi が行うネットワークトラフィックも増えます。**[再検出間隔]** が長くなるほど，検出の精度は下がり，NNMi が行うネットワークトラフィックも減ります。これは，ネットワークが大きくなるほど，ユーザーが行う再検出の頻度が少なくなることを意味します。**[再検出間隔]** を設定するには，次の手順を実行します。

1. regional1 または regional2 の NNMi コンソールから，**[設定]** ワークスペースの **[検出]** > **[検出の設定]** をクリックする。



2. リージョナルサーバーで検出を開始する頻度に従い，［再検出間隔］を調整する。
   グローバルサーバーは，リージョナルサーバーが検出を完了するとすぐに検出を開始します。



3. [保存して閉じる] をクリックする。

15.10.2 リージョナルマネージャーからグローバルマネージャーへのカスタム属性の複製

NNMi では，リージョナルマネージャーでカスタム属性を設定して，それらのカスタム属性をグローバルマネージャーに複製できます。例えば，カスタム属性データをリージョナルマネージャーのノードに追加して，そのデータをグローバルマネージャーに複製したあとで，そのデータを使用してそれらのノードのインシデントを強化できます。

メモ

NNMi では，リージョナルマネージャーからグローバルマネージャーにノードおよびインタフェースのカスタム属性を複製できます。

NNMi コンソールで，グローバルマネージャーの [カスタム属性の複製] タブ（[グローバルネットワーク管理] 設定内）を使用してカスタム属性の複製を設定できます。

メモ

NNMi では，ユーザーによる設定や入力を行わずに無番号インタフェースのカスタム属性が複製されます。詳細については，NNMi ヘルプを参照してください。

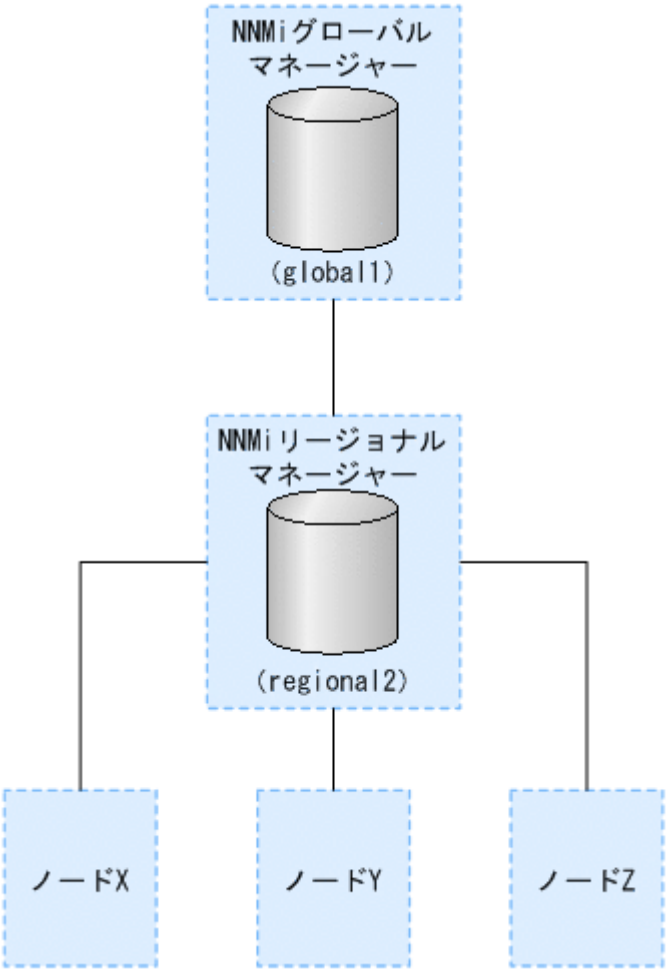
15.10.3　デバイスに対するステータスポーリングまたは設定ポーリング

この例では，次の 2 つを前提としています。

- リージョナル NNMi 管理サーバー regional2 は，Node X を検出および管理する。
- グローバル NNMi 管理サーバー global1 は，リージョナル NNMi 管理サーバー regional2 と接続する。

次の図を参照してください。

図 15-2　ノードのステータスポーリングまたは設定ポーリング



global1 から Node X のステータスをポーリングするには，次を実行します。

1. global1 から，［インベントリ］ワークスペースの［ノード］をクリックする。
2. ノードインベントリから Node X を選択する。
3. ［アクション］＞［ポーリング］＞［ステータスのポーリング］メニュー項目を使用して，Node X のステータスポーリングを要求する。
4. NNMi 管理サーバー global1 は，リージョナル NNMi 管理サーバー regional2 からのステータスポーリングを要求し，結果を画面に表示する。

   ステータスポーリング要求は，global1 と regional2 のどちらから発行しても問題ありません。

ステータスポーリングの結果は同じものが表示されます。

global1 で Node X の最新の検出情報を取得するようにするには，次を実行して global1 から Node X の設定ポーリングを行います。

1. global1 から，［インベントリ］ワークスペースの［ノード］をクリックする。
2. ノードインベントリから Node X を選択する。
3. ［アクション］＞［ポーリング］＞［設定のポーリング］メニュー項目を使用して，Node X の設定ポーリングを要求する。
4. NNMi 管理サーバー global1 は，リージョナル NNMi 管理サーバー regional2 からの設定ポーリングを要求し，結果を画面に表示する。

   設定ポーリング要求は，global1 と regional2 のどちらから発行しても問題はありません。設定ポーリングの結果は同じものが表示されます。

15.10.4 グローバルマネージャーでのデバイスステータスの判定とインシデントの生成

NNMi 管理サーバー global1 は，リージョナルマネージャー regional1 と regional2 からくるステータス変更をリッスンし，ローカルデータベースにあるステータスを更新します。

NNMi 管理サーバー regional1 と regional2 の NNMi StatePoller サービスは，監視するデバイスの状態の値を計算します。global1 は，regional1 と regional2 から状態の値の更新を受け取ります。global1 は，自分が検出するノードにポーリングしますが，regional1 と regional2 によって管理されているノードにはポーリングしません。

regional1 によって管理されているノードの管理モードを変更したあと，global1 上の管理モードも変更されます。ネットワーク管理者が regional1 または regional2 によって管理されるネットワーク機器の追加，削除，変更を行うと，regional1 または regional2 はそれらのネットワークデバイスの変更について global1 を更新します。

global1 は，regional1 と regional2 によって転送されてきたノードオブジェクトデータなど，独自の Causal Engine とトポロジを使用してインシデントを生成します。これは，生成するインシデントが，トポロジに違いがある場合に，regional1 と regional2 のインシデントとは少し異なる場合があることを意味します。

フィルタリングが global1 の接続性に影響する可能性があるため，転送フィルタを regional1 や regional2 に使用することは避けた方がよいでしょう。ここで生じる差異が，global1 と 2 つのリージョナル（regional1 と regional2）との間の根本原因分析での差異になる可能性があります。ほとんどの場合，転送フィルタの使用しないことを選択すると，グローバル NNMi 管理サーバーのトポロジは大きくなります。これは，より正確な根本原因分析の結果を得るのに役立ちます。

追加の設定をしないと，regional1 はトラップを global1 に転送しません。これを行うには，特定のトラップを global1 に転送するように regional1 を設定する必要があります。グローバルマネージャーに過剰な負荷がかからないように，リージョナルマネージャーは量の少ない，重要なトラップを転送するよう設定することをお勧めします。NNMi は，転送されたトラップが TrapStorm インシデントを引き起こすような場合，転送されたトラップを削除します。NNMi コンソールで TrapStorm 管理イベントの詳細を参照してください。

15.11　グローバルネットワーク管理のトラブルシューティングのヒント

15.11.1　NNMi ヘルプのトラブルシューティング情報

グローバルネットワーク管理のトラブルシューティング情報については，NNMi ヘルプの「グローバルネットワーク管理をトラブルシューティングする」を参照してください。

15.11.2　クロック同期

グローバルネットワーク管理（グローバルマネージャーとリージョナルマネージャー）やシングルサインオン（SSO）に属するネットワーク環境内のすべての NNMi 管理サーバーは，それぞれの内部タイムクロックを世界標準時で同期化する必要があります。例えば，Linux ツールの Network Time Protocol Daemon（NTPD）や使用可能な Windows オペレーティングシステムツールなどの時刻の同期プログラムを使用します。

NNMi コンソールの下部に次のメッセージが表示される場合の対応は，次のとおりです。

```
NNMiのセルフモニタリングが問題を検出しました（警戒域）。詳細は，［ヘルプ］＞［システム情報］
＞［ヘルス］を参照してください。
```

グローバルマネージャーの`nnm.log` ファイルに次のメッセージがないか確認します。

```
致命的
[com.hp.ov.nms.topo.spi.server.bridge.BridgeConnectionSelectorImpl] <number of seconds>のク
ロックの違いにより、システム<server_name>には接続されません。リモート時間は、<date/time>で
す。
```

クロックが合っていないため，再同期化が必要です。

このメッセージがログに出力されて数分以内に，NNMi はリージョナルマネージャー接続を切断します。

また，NNMi セルフモニタリングが次の問題を検出します。

[警戒域] リージョナルマネージャー ’<name>’ への接続は停止しています。

15.11.3　グローバルネットワーク管理のシステム情報

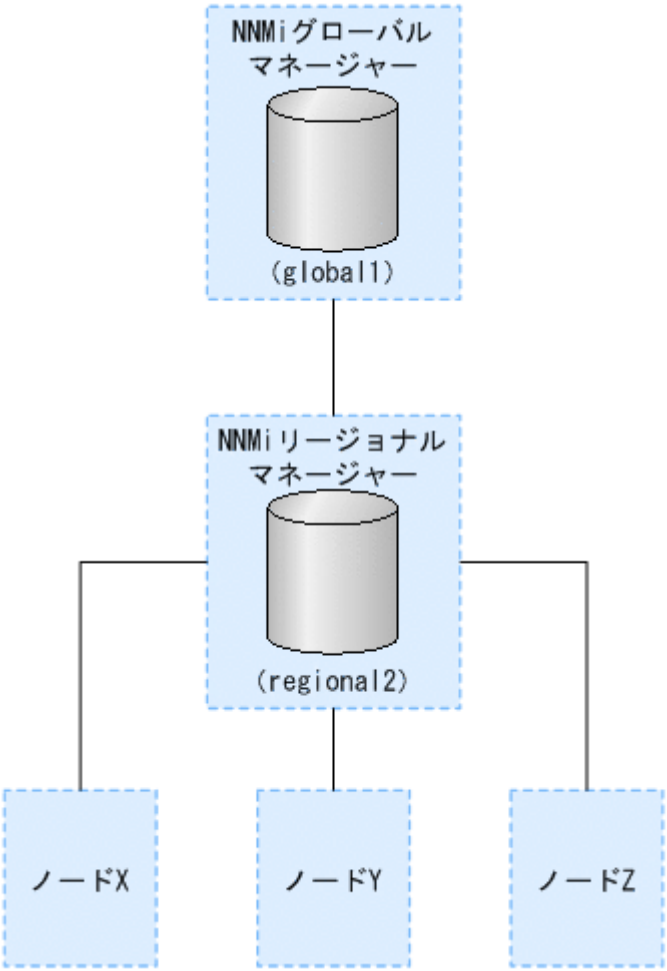
グローバルネットワーク管理接続に関する情報を表示するには，**［ヘルプ］** ＞ **［システム情報］** を選択して **［グローバルネットワーク管理］** タブをクリックします。

15.11.4 グローバルマネージャーとリージョナルマネージャーの検出情報の同期

global1 と regional2 の間で情報に矛盾があることに気が付いた場合は，global1 から nnmnoderediscover.ovpl スクリプトを実行して，global1 と regional2 を同期します。実行の結果，regional2 は新しい検出結果を使用して global1 を更新します。

この例では，次の図に示すネットワークを使用します。

図 15-3　グローバルネットワーク管理



次のコマンドを実行してノード X，Y，および Z と global1 を同期化します。

```
nnmnoderediscover.ovpl -u username -p password -rm regional2
```

詳細については，nnmnoderediscover.ovpl のリファレンスページを参照してください。

次のことに注意してください。

- NNMi は，手動再同期の後，トポロジ，状態，およびステータスを自動的に再同期します。

15.12　グローバルネットワーク管理環境での NNMi のバージョンアップ手順

グローバルネットワーク管理環境で設定されている NNMi 管理サーバーをバージョンアップする場合は「24.3　NNMi 13-00 からのグローバルマネージャーとリージョナルマネージャーのアップグレード」を参照してください。

15.13　グローバルネットワーク管理とアドレス変換プロトコル

動的ネットワークアドレス変換（NAT），動的ポートアドレス変換（PAT），または動的ネットワークアドレスおよびポート変換（NAPT）の各グループには，NNMi グローバルネットワーク管理設定全体で一意のテナントに加え，NNMi リージョナルマネージャーが必要です。「13.　NAT 環境の重複 IP アドレスの管理」を参照してください。NNMi ヘルプも参照してください。

16 NNMi IPv6 管理機能

IPv6 管理機能を使用するには，NNMi Advanced ライセンスを購入してインストールする必要があります。この章での NNMi は，NNMi Advanced ライセンスがインストールされている NNMi を指します。NNMi の IPv6 管理で，インタフェース，ノード，サブネットも含めた IPv6 アドレスの検出と監視が可能になります。シームレスな統合を提供するため，NNMi は IPv4 と IPv6 両方のアドレスを含めるよう IP アドレスモデルを拡張します。NNMi では，可能なかぎりすべての IP アドレスが等しく扱われます。IPv4 アドレスに関連するほとんどの機能は IPv6 アドレスについても使用できます。ただし，幾つか例外があります。NNMi コンソールに表示される IPv6 情報の詳細については，NNMi ヘルプを参照してください。

16.1 NNMi IPv6 管理機能の概要

NNMi IPv6 管理機能には，次の機能があります。

- IPv6 専用デバイスおよびデュアルスタックデバイスの IPv6 インベントリ検出
  - IPv6 アドレス
  - IPv6 サブネット
  - IPv6 アドレス，サブネット，インタフェースおよびノード間の関連づけ
- 次のためのネイティブ IPv6 SNMP 通信
  - ノードの検出
  - インタフェースの監視
  - トラップと通知の受信と転送
- デュアルスタックデバイスでの IPv4 または IPv6 通信（管理アドレス）の自動選択

  NNMi コンソールを使用し，［設定］ワークスペースの［通信の設定］で，SNMP 管理アドレス設定を IPv4 または IPv6 に設定します。
- IPv6 アドレスフォルト監視のためのネイティブ ICMPv6 通信
- IPv6 アドレスまたはホスト名をシードに使用したデバイスの検出
- IPv6 レイヤー 3 隣接検出ヒントを使用した IPv6 デバイスの自動検出
- LLDP（Link Layer Discovery Protocol） IPv6 隣接情報を使用するレイヤー 2 隣接検出ヒントを使用した IPv6 デバイスの自動検出
- IPv4，IPv6 情報の統合表示
  - ノード，インタフェース，アドレス，サブネットおよび関連づけのインベントリビュー
  - IPv4 デバイスと IPv6 デバイス用のレイヤー 2 隣接ビューおよびトポロジマップ
  - IPv4 デバイスと IPv6 デバイス用のレイヤー 3 隣接ビューおよびトポロジマップ
  - インシデント，結果，根本原因分析
- NNMi コンソールアクション：IPv6 アドレスとノードに対する`ping` と`traceroute`
- IPv6 アドレスとアドレス範囲を使用した NNMi 設定
  - 通信の設定
  - 検出の設定
  - 監視の設定
  - ノードとインタフェースグループ
  - インシデントの設定
- IPv6 インベントリとインシデント用の DTK Web サービスサポート

NNMi IPv6 管理機能では，次はサポートしていません。

- 検出のための IPv6 Ping スイープの使用
- IPv6 ネットワークパスビュー（Smart Path）
- IPv6 リンクローカルアドレス障害監視
- 検出シードとしての IPv6 リンクローカルアドレスの使用

16.2 NNMi IPv6 管理機能を使用するための必要条件

管理サーバーの仕様および NNMi のインストールの詳細については，リリースノートを参照してください。

ネイティブ IPv6 通信を使用するには，NNMi 管理サーバーはデュアルスタックシステムであることが必要です。つまり，IPv4 と IPv6 両方を使用して通信するということです。

IPv6 の追加要件は次のとおりです。

- 少なくとも 1 つのネットワークインタフェースで IPv4 を有効化し設定する必要があります。
- IPv6 を有効にして管理する必要のある，IPv6 ネットワークに接続する少なくとも 1 つのネットワークインタフェースで，リンクローカルユニキャストアドレス以外のユニキャストアドレス（例：グローバルユニキャストアドレス，ユニークローカル IPv6 ユニキャストアドレス）を持つ必要があります。
- NNMi 管理サーバーに IPv6 ルートを設定し，IPv6 を使用して NNMi で検出と監視を行うデバイスと NNMi が通信できるようにする必要があります。

メモ

IPv4 専用の NNMi 管理サーバーを使用することもできますが，IPv4/IPv6 デュアルスタックデバイスを NNMi で完全に管理することはできなくなります。例えば，IPv4 専用管理サーバーを使用すると，NNMi は IPv6 専用デバイスの検出，IPv6 シードとヒントを使用した検出，および IPv6 アドレスを持つデバイス上での障害の監視はできません。

NNMi 管理サーバーで使用される DNS サーバーは，ホスト名から IPv6 アドレスおよび IPv6 アドレスからホスト名を名前解決する必要があります。つまり，DNS サーバーはホスト名を 128 ビット IPv6 アドレスにマッピングする必要があります。IPv6 対応 DNS サーバーが使用できない場合でも，NNMi は正しく機能しますが，NNMi では IPv6 アドレスを使用するノードの DNS ホスト名の判定や表示は行いません。

16.3 NNMi IPv6 管理機能を使用するためのライセンス

すでに説明したように，IPv6 管理機能を使用するには NNMi Advanced ライセンスを購入してインストールする必要があります。NNMi Advanced ライセンスの取得とインストールの詳細については，「2. NNMi のインストールとアンインストール」を参照してください。

NNMi 製品には，インスタントオンライセンス用パスワードが含まれています。これは一時的なものですが，有効な NNMi Advanced ライセンスです。できるだけ早く，恒久ライセンスキーを入手してインストールしてください。

16.4　NNMi IPv6 管理機能がサポートする環境

NNMi をサポートするオペレーティングシステム構成の詳細については，リリースノートを参照してください。

16.4.1　NNMi 管理サーバーの種類とサポートする機能

次の表に，IPv4 専用およびデュアルスタック両方の NNMi 管理サーバーの機能を示します。

表 16-1　管理サーバーの機能

機能	IPv4 専用	デュアルスタック
IPv4 通信（SNMP，ICMP）	対応	対応
IPv6 通信（SNMP，ICMPv6）	非対応	対応
デュアルスタック管理ノード	対応	対応
IPv4 シードを使用した検出	対応	対応
IPv6 シードを使用した検出	非対応	対応
IPv4 アドレスおよびサブネットインベントリ	対応	対応
IPv6 アドレスおよびサブネットインベントリ	対応	対応
SNMP を使用したインタフェースステータスとパフォーマンス	対応	対応
ICMP を使用した IPv4 アドレスステータス	対応	対応
ICMPv6 を使用した IPv6 アドレスステータス	非対応	対応
IPv6 専用管理ノード	非対応	対応
IPv4 専用管理ノード	対応	対応

16.4.2　IPv6 をサポートしている SNMP MIB

NNMi では，IPv6 用の次の SNMP MIB がサポートされています。

- RFC 4293（現在の IETF 標準）
- RFC 2465（元の IETF 提案）
- Cisco IP-MIB

16.5　NNMi のインストールと IPv6 管理機能の有効化

NNMi のインストール中に，インストールスクリプトが IPv6 機能をアクティブにします。ただし，必要に応じて`nms-jboss.properties` ファイルを編集し，これらの IPv6 機能を手動で非アクティブにできます。

非アクティブにしたあとで，IPv6 機能を再度アクティブにできます。詳細については，「16.6　IPv6 管理機能を無効にする」および「16.7　IPv6 管理機能を再度有効にする」を参照してください。

16.6 IPv6 管理機能を無効にする

次のどちらかの方法を使用して，管理上 IPv6 機能を無効化できます。

1. nms-jboss.properties ファイルの IPv6 マスタースイッチをオフにし，NNMi を再起動する。
2. NNMi Advanced ライセンスを期限切れにするか，または基本 NNMi ライセンスに置き換える。

1 の方法で管理上 IPv6 機能を無効化する手順は，次のとおりです。

1. nms-jboss.properties ファイルを開く。
   次の場所を探してください。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties

   NNMi では，各プロパティの完全な記述を用意しており，nms-jboss.properties ファイルのコメントとして示しています。
2. NNMi の IPv6 通信を非アクティブ化するには，次の手順を実行する。
   a # Enable Java IPv6 Communication で始まるテキストを探します。
   b 次の行を見つけます。

   ```
   java.net.preferIPv4Stack=false
   ```

   c この行を次のように編集します。

   ```
   java.net.preferIPv4Stack=true
   ```

   行がコメント化されていないことを確認します。
3. NNMi で IPv6 管理全体を非アクティブ化するには，次の手順を実行する。
   a # Enable NNMi IPv6 Management で始まるテキストを探します。
   b 次の行を見つけます。

   ```
   com.hp.nnm.enableIPv6Mgmt=true
   ```

   c この行を次のように編集します。

   ```
   com.hp.nnm.enableIPv6Mgmt=false
   ```

   行がコメント化されていないことを確認します。
   d nms-jboss.properties ファイルを保存して閉じます。
4. 次のコマンドを実行して，NNMi を再起動する。

   ```
   ovstop
   ovstart
   ```

重要

高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，`ovstop` および`ovstart` コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

5. 次のコマンドを使用して，NNMi プロセスを確認する。

```
ovstatus -v ovjboss
```

次のセクションでは，IPv6 を無効化したあとの NNMi の動作とインベントリのクリーンアップについて説明します。

16.6.1　IPv6 管理機能を無効にしたあとの IPv6 監視

IPv6 管理または IPv6 通信が完全に無効になると，StatePoller サービスは ICMPv6 による IPv6 アドレスの監視をすぐに停止します。NNMi は，これらのアドレスの IP アドレス状態を［未ポーリング］に設定します。アドレスを選択し，このアドレスに対して［アクション］＞［設定の詳細］＞［モニタリングの設定］を使用すると，関連する［モニタリングの設定］ルールで［IP アドレス障害のポーリングを有効にする］が有効になっている場合でも，NNMi は「`ICMPポーリングの管理アドレス : false`」と表示します。

16.6.2　IPv6 管理機能を無効にしたあとの IPv6 インベントリ

一度 NNMi が完全に IPv6 インベントリを検出すると，次の場合には，NNMi にそのインベントリを自動的に消去させることができます。

- マスター IPv6 スイッチをオンにしたあとで，オフにして NNMi を再起動した。

  NNMi は IPv6 インベントリをすぐに削除しません。NNMi は SNMP ノードの IPv6 インベントリを次の検出サイクルで削除します。ただし，管理アドレスが IPv6 アドレスであったノードの場合，管理アドレスが IPv6 アドレスのまま残ります。また，NNMi は SNMP IPv6 でないノードを削除しません。IPv6 データが残ったノードは，NNMi インベントリから手動で削除する必要があります。

16.6.3　IPv6 インベントリクリーンアップ時の既知の問題点

IPv6 インベントリが残る場合があります。例えば，NNMi が SNMP を使用して，ある IPv6 ノードを正常に管理し，次の検出の前にそのノードにアクセスできなくなったような場合です。既存の検出システムの設計上，検出プロセスは SNMP を使用した通信ができなくなったノードを更新できません。このように

して残ったノードを削除するには，通信の問題を解決してから，NNMi コンソールの［アクション］＞［ポーリング］＞［設定のポーリング］コマンドを使用してそれらのノードの設定情報を取得する必要があります。ネイティブ IPv6 ノードの場合，NNMi コンソールから直接ノードを削除します。

16.7　IPv6 管理機能を再度有効にする

IPv6 専用デバイスの検出や IPv6 アドレスステータスの監視など，IPv6 通信を必要とする機能では，NNMi 管理サーバーに IPv6 グローバルユニキャストアドレスが設定され機能を果たすことが必要です。

次に示す手順で非アクティブにしたあとで，IPv6 機能を再度アクティブにする方法を説明します。

1. nms-jboss.properties ファイルを編集する。

   次の場所を探してください。

   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties

   NNMi では，各プロパティの完全な記述を用意しており，nms-jboss.properties ファイルのコメントとして示しています。

2. # Enable NNMi IPv6 Management で始まるテキストを探す。
3. NNMi で IPv6 通信を有効化するには，次のプロパティをコメント解除する。

   ```
   java.net.preferIPv4Stack=false
   ```

   プロパティをコメント解除するには，行の先頭から#!文字を削除します。

4. # Enable NNMi IPv6 Management で始まるテキストを探す。
5. NNMi で IPv6 通信全体を有効化するには，次のプロパティをコメント解除する。

   ```
   com.hp.nnm.enableIPv6Mgmt=true
   ```

6. nms-jboss.properties ファイルを保存して閉じる。
7. 次のコマンドを実行して，NNMi を再起動する。

   ```
   ovstop
   ovstart
   ```

   重要

   高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，ovstop およびovstart コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

8. 次のコマンドを使用して，NNMi プロセスを確認する。

   ```
   ovstatus -v ovjboss
   ```

   起動に成功すると，次のように表示されます。

```
object manager name: ovjboss
state: RUNNING
PID: <Process ID #>
last message: Initialization complete.
exit status: -
additional info:
SERVICE STATUS
CommunicationModelService サービスが起動されました
CommunicationParametersStatsService サービスが起動されました
CustomPoller サービスが起動されました
IslandSpotterService サービスが起動されました
ManagedNodeLicenseManager サービスが起動されました
MonitoringSettingsService サービスが起動されました
NamedPoll サービスが起動されました
msApa サービスが起動されました
NmsCustomCorrelation サービスが起動されました
NmsDisco サービスが起動されました
NmsEvents サービスが起動されました
NmsEventsConfiguration サービスが起動されました
NmsExtensionNotificationService サービスが起動されました
NnmTrapService サービスが起動されました
PerformanceSpiAdapterTopologyChangeService サービスが起動されました
PerformanceSpiConsumptionManager サービスが起動されました
RbaManager サービスが起動されました
RediscoverQueue サービスが起動されました
SpmdjbossStart サービスが起動されました
StagedIcmp サービスが起動されました
StagedSnmp サービスが起動されました
StatePoller サービスが起動されました
TrapConfigurationService サービスが起動されました
TrustManager サービスが起動されました
```

9. IPv6 を再度アクティブにすると，NNMi ビューには，新たに検出されたノードの IPv6 インベントリが表示される。

   次の検出サイクルの間に，NNMi ビューにはその前の検出ノードに関連する IPv6 インベントリが表示されます。

10. 必要に応じて，デュアルスタック管理ノードの SNMP 管理アドレス設定を指定する。

    デュアルスタック管理ノードは，IPv4 または IPv6 のどちらかを使用して通信できるノードです。これを行うには，次の手順を実行します。

    a NNMi コンソールで，**[設定]** ワークスペースにある **[通信の設定]** をクリックします。

    b **[管理アドレスの選択]** セクションを見つけます。**[IP バージョン設定]** フィールドで，**[IPv4]**，**[IPv6]**，または **[任意]** を選択します。

    c 変更を保存します。

    d 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

重要

高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，`ovstop` および`ovstart` コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

スピードアップを図るには，デュアルスタックノードとわかっているノードを選択し，NNMi コンソールで［**アクション**］＞［**設定のポーリング**］コマンドを使用します。`nnmnoderediscover.ovpl` スクリプトを使用して，NNMi 検出キューにノードを追加することもできます。詳細については，`nnmnoderediscover.ovpl` のリファレンスページを参照してください。

NNMi 管理サーバーで IPv6 通信を有効化すると，NNMi は ICMPv6 を使用して IPv6 アドレスフォルトがないかノードの監視を開始します。

17 NNMi がサポートするデータの保護

この章では，ハードウェア障害の場合の NNMi データを保護するため，NNMi がサポートしている方法について説明します。

17.1　NNMi がサポートするデータ保護の仕組み

NNMi では，ハードウェア障害の場合に NNMi データを保護するため，次の 2 つの方法をサポートしています。

- アプリケーションフェイルオーバー構成

  NNMi のアプリケーションフェイルオーバーでは，NNMi データベースのトランザクションログのコピーが同一設定システムで維持され，ディザスタリカバリが提供されます。詳細については，「18.　アプリケーションフェイルオーバー構成の NNMi を設定する」を参照してください。

- 高可用性（HA）クラスタでの動作

  HA クラスタで NNMi を実行すると，NNMi データベースと設定ファイルが共有ディスクに保持され，NNMi 管理サーバーのほぼ 100 パーセントの可用性が提供されます。詳細については，「19.　高可用性クラスタに NNMi を設定する」を参照してください。

これらの方法では，現在の NNMi 管理サーバーで障害が発生すると，第 2 システムが自動的に NNMi 管理サーバーになります。

17.2　NNMi がサポートするデータ保護の仕組みの比較

NNMi がサポートするデータ保護の仕組みの比較を，次の表に示します。

表 17-1　NNMi がサポートするデータ保護の仕組みの比較

比較項目	NNMi のアプリケーションフェイルオーバー	HA クラスタで動作する NNMi
必要なソフトウェア製品	NNMi	• NNMi • 個別に購入する HA 製品
フェイルオーバーに掛かる時間	トランザクションログを処理する時間（通常の状態では 10 分～60 分）	通常の状態では 5 分～30 分
フェイルオーバーの透過性	部分的。 NNMi 管理サーバーの IP アドレスは，スタンバイサーバーの物理アドレスに変わります。ユーザーは新しい IP アドレスで，NNMi コンソールに接続してください。	完全。 すべての接続は HA クラスタの仮想 IP アドレスが使用され，これはフェイルオーバー時にも変わりません。
アクティブサーバーとスタンバイサーバーの相対的な近接性	LAN または WAN	LAN または WAN（一部の HA 製品だけ）
グローバルネットワーク管理とのインタラクション	アプリケーションフェイルオーバー アプリケーションフェイルオーバー用にグローバルマネージャー，リージョナルマネージャーを設定できません。 HA ・HA 用に各グローバルマネージャー，リージョナルマネージャーを設定できます。 ・それぞれの設定には，2 つの物理システムが必要です。 ・グローバルマネージャーまたはリージョナルマネージャーがフェイルオーバーすると，NNMi は，グローバルマネージャーとリージョナルマネージャー間の接続を再確立します。	
NNMi のメンテナンス	修正版の適用またはバージョンアップをする前に，NNMi のアプリケーションフェイルオーバークラスタを停止する必要があります。	HA を設定解除しないで，NNMi に修正版の適用またはバージョンアップができます。

18 アプリケーションフェイルオーバー構成の NNMi を設定する

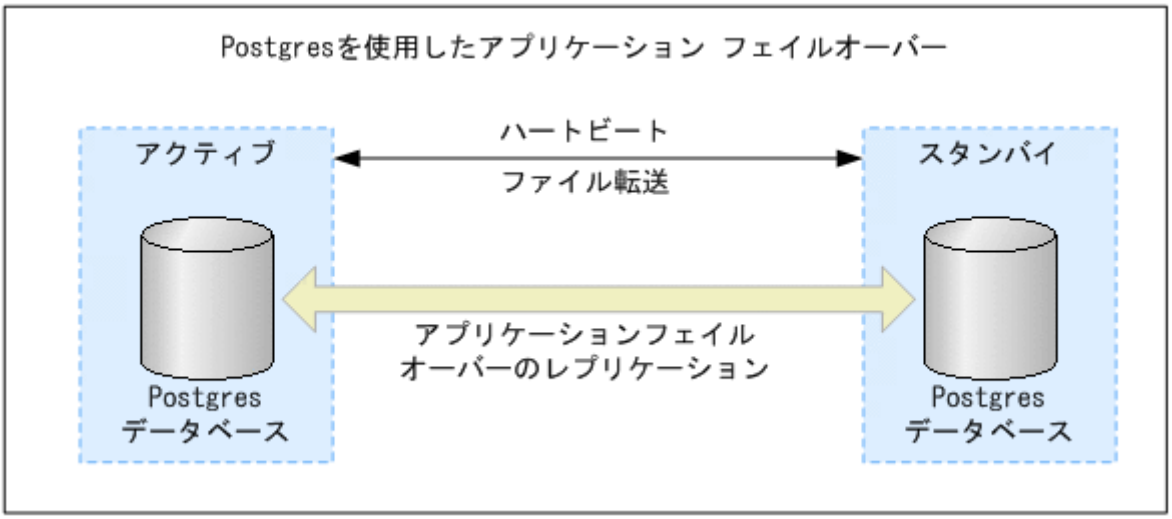
重要なネットワーク機器の障害発生を知らせ，その障害の根本原因を示す NNMi は，多くの IT プロフェッショナルから信頼を寄せられています。NNMi 管理サーバーに障害が発生した場合でも，引き続き NNMi がネットワーク機器の障害発生を知らせてくれる必要があります。このニーズを満たすのが NNMi のアプリケーションフェイルオーバーで，NNMi プロセスのアプリケーションコントロールをアクティブな NNMi 管理サーバーからスタンバイ NNMi 管理サーバーに引き渡すことで，NNMi の機能は中断なく提供されます。

18.1 アプリケーションフェイルオーバーの概要

アプリケーションフェイルオーバーは，クラスタソフトや共有ディスクなしで NNMi 管理サーバーを多重化する機能です。

2 台の NNMi 管理サーバーをアクティブサーバーおよびスタンバイサーバーとして構成し，NNMi が稼働するアクティブサーバーに障害が発生したときに，スタンバイサーバーに NNMi を引き継ぐことでネットワーク監視を継続できます。

このアプリケーションフェイルオーバー機能は，NNMi 独自のクラスタマネージャー（nnmcluster プロセス）の制御によって実現していて，クラスタソフトと連携する HA 構成とは異なった特徴があります。なお，マニュアルやヘルプでは，アプリケーションフェイルオーバーの構成を NNMi クラスタ（または単にクラスタ）と表記している場合がありますので適宜読み替えてください。



アプリケーションフェイルオーバー機能は，NNMi データベースを使用して NNMi をインストールすることで利用できるようになります。システムにアプリケーションフェイルオーバー機能を設定すると，NNMi は NNMi 管理サーバーの障害を検出した場合に，スタンバイサーバーに NNMi の機能を引き渡します。

NNMi のアプリケーションフェイルオーバー設定では，次の用語と定義を使用しています。

- アクティブ：ネットワーク監視を実行中のサーバー。
- スタンバイ：フェイルオーバーのイベントを待機している NNMi クラスタ内のサーバー。このサーバーはネットワーク監視を実行していません。
- Cluster Member：クラスタに接続するために JGroups 技術を使用しているシステムで実行中の Java プロセス。1 つのシステムに複数のメンバーを登録できます。
- Postgres：トポロジ，インシデント，設定情報などの情報を保存するために NNMi が使用するデータベース。
- Cluster Manager：アプリケーションフェイルオーバー機能でサーバーの監視と管理に使用される nnmcluster プロセスおよびツール。

18.2　アプリケーションフェイルオーバーの基本セットアップ

アプリケーションフェイルオーバー機能を導入するには，NNMi を 2 つのサーバーにインストールします。ここでは，この 2 つの NNMi 管理サーバーをアクティブサーバーとスタンバイサーバーとして説明します。通常の運用では，アクティブサーバーだけがネットワーク監視を実行します。

アクティブおよびスタンバイ NNMi 管理サーバーは，各 NNMi 管理サーバーのハートビートを監視するクラスタの一部です。アクティブサーバーに障害が発生し，そのハートビートが消失すると，スタンバイサーバーがアクティブサーバーになります。

アプリケーションフェイルオーバー機能は，次のどちらかの方法で設定できます。

- 手動によるアプリケーションフェイルオーバーの設定
- NNMi クラスタセットアップウィザードを使用したアプリケーションフェイルオーバーの設定

18.2.1　アプリケーションフェイルオーバーを設定するための前提条件

アプリケーションフェイルオーバーが正しく機能するには，NNMi 管理サーバーが次の要件を満たしている必要があります。

- NNMi を単独で使用する構成だけをサポートしています。
  ほかの JP1 などの関連製品と連携して使用する構成はサポートしていませんので，この場合は，クラスタソフトによる HA 構成を使用してください。
- 両方の NNMi 管理サーバーで，アクティブサーバーのホスト名と IP アドレス，スタンバイサーバーのホスト名と IP アドレスが名前解決できる必要があります。
- 両方の NNMi 管理サーバーの持つすべての IPv4 アドレスが，ネットワーク上で重複していない必要があります。
- 両方の NNMi 管理サーバーが同じ種類のオペレーティングシステムを実行している必要があります。例えば，アクティブサーバーが Microsoft(R) Windows Server(R) 2012 R2 Datacenter を実行している場合，スタンバイサーバーも Microsoft(R) Windows Server(R) 2012 R2 Datacenter を実行している必要があります。
- 両方の NNMi 管理サーバーは同じバージョン（修正版のバージョンを含む）の NNMi を実行している必要があります。例えば，アクティブサーバーで NNMi 12-00 を実行している場合，スタンバイサーバーでも同一の NNMi 12-00 がインストールされている必要があります。
- 両方の NNMi 管理サーバーの system ユーザーのパスワードが同一である必要があります。
- アプリケーションフェイルオーバーを設定する前に NNMi への HTTP アクセスを完全に無効にしないでください。詳細については，「21.18　リモートアクセスには暗号化を必須とするように NNMi を設定する」を参照してください。アプリケーションフェイルオーバークラスターの設定が正常に完了したあと，HTTP およびその他の非暗号化アクセスを無効にできます。

- （Windows の場合）両方の NNMi 管理サーバーは NNMi のインストール先が同一で，%NnmDataDir%および%NnmInstallDir%のシステム変数を同一の値に設定している必要があります。
- 両方の NNMi 管理サーバーのライセンス属性（管理ノード数，NNMi か NNMi Advanced か）が同一である必要があります。例えば，ノードカウントおよびライセンス取得済みの機能が同一である必要があります。

> **重要**
>
> スタンバイサーバーにも同一のライセンスが必要です。

- NNMi が初回検出の高度なステージに入るまで，アプリケーションフェイルオーバーを有効にしないでください。詳細については，「6.4　検出の評価」を参照してください。
- アプリケーションフェイルオーバーが正しく機能するには，アクティブサーバーとスタンバイサーバーは相互のネットワークアクセスに制限のないことが必要です。ファイルをロックしたり，ネットワークのアクセスを制限したりするソフトウェアが原因で，NNMi の通信の問題が発生する場合があります。こうしたアプリケーションで，NNMi が使用するファイルとポートを無視するように設定します。
- アクティブサーバーとスタンバイサーバーは，クラスタ通信に使用する NIC の持つすべての IPv4 アドレスで，アプリケーションフェールオーバーを構成する相手の NNMi 管理サーバーと通信ができる必要があります。相手と通信できない IPv4 アドレスが存在する場合，アプリケーションフェールオーバーの構築に失敗したり，構築済みのアプリケーションフェールオーバー環境の動作が不正になったりします。クラスタ通信に使用する NIC の設定方法については，「18.3.3　アプリケーションフェイルオーバー通信の設定」を参照してください。
- アクティブサーバーとスタンバイサーバーの間に，ファイアウォールを設置することは推奨しません。ファイアウォールを設置する場合は，両サーバーがすべてのポートで通信できるように設定してください。
- NNMi 管理サーバー内でファイアウォールを実行する場合，自サーバー内のプロセス同士の通信および相手サーバーとの通信を，すべてのポートで許可するように設定してください。アプリケーションフェイルオーバーは動的に任意のポートで通信します。
  - プロセス単位で通信許可を設定するファイアウォールの場合（例：Windows Firewall）は，クラスタマネージャー（nnmcluster.exe）の通信を許可してください。
  - ポート単位で通信許可を設定するファイアウォールの場合，次の通信を許可してください。

    IP アドレス：自サーバーと相手サーバーに割り当てられたすべての IP アドレス

    ポート：すべてのポート
- アクティブサーバーとスタンバイサーバーの NNMi データベースは同じパスワードが設定されている必要があります。

  NNMi データベースのパスワードを変更した場合は，アプリケーションフェイルオーバーの設定を行う前に，すべてのサーバーで同じパスワードを設定してください。

この条件を満たしたら，「18.3　アプリケーションフェイルオーバー構成の NNMi を設定する」に示した手順を実行してください。詳細については，「付録 E　NNMi が使用するポートの一覧」を参照してください。

18.2.2　アプリケーションフェイルオーバーの注意事項

アプリケーションフェイルオーバーについての注意事項を説明します。

- アプリケーションフェイルオーバー構成では，サーバー停止時には NNMi がフェイルオーバーしますが，（nnmcluster 以外の）NNMi のプロセスが停止してもフェイルオーバーはしません。詳しくは「18.4.2　アプリケーションフェイルオーバーのシナリオ」を参照してください。

  NNMi プロセスが停止したときにフェイルオーバーさせたい場合は，HA クラスタソフトによる HA 構成を使用してください。
- スタンバイサーバーが停止している場合（切り替え先がない状態）にそれを通知する機能は提供していません。
- フェイルオーバー時に何らかの処理を実行するためのユーザー指定コマンドを実行する機能は提供していません。
- NNMi が稼働するサーバーの IP アドレスがフェイルオーバー時に変わります。IP アドレスは引き継ぎません。このため，次の点に注意してください。
  - SNMP トラップの送信先は，両方の NNMi 管理サーバーに設定してください。
  - Web ブラウザに両方の NNMi 管理サーバーのブックマークを登録しておき，アクティブサーバー側に接続してください。
- バックアップを定期的に行い，万一データが壊れた場合に備えてください。アプリケーションフェイルオーバー機能でスタンバイサーバーに複製されるデータは，バックアップの代替としては使用できません。
- アプリケーションフェイルオーバー構成では，データベース部分は通常構成に比べて 3 倍のディスク容量が必要です。データベースの構成について「18.4.1　アプリケーションフェイルオーバーの動作」を参照してください。

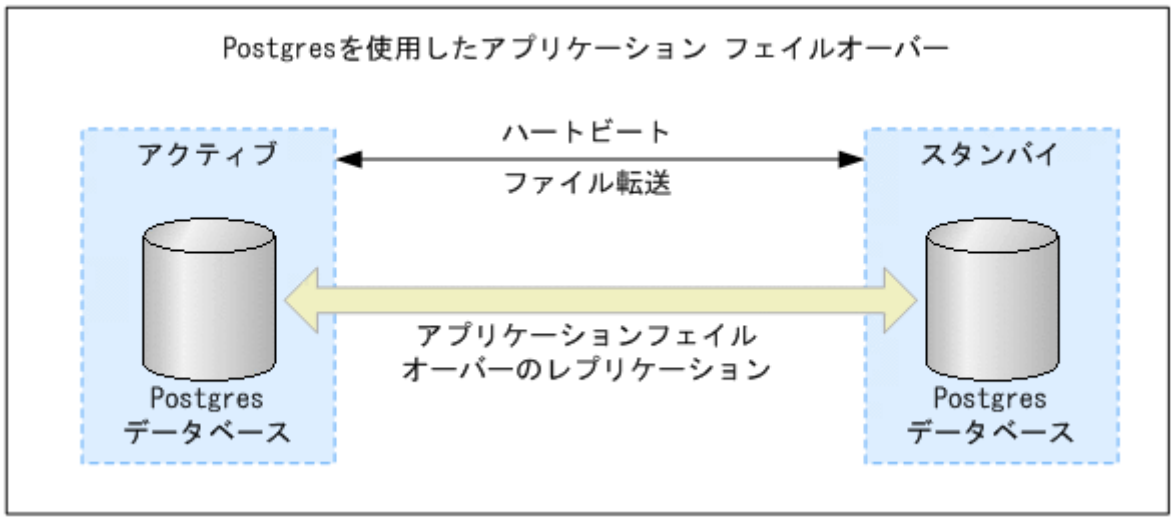
18.3　アプリケーションフェイルオーバー構成の NNMi を設定する

アプリケーションフェイルオーバー構成の NNMi の設定方法について説明します。

18.3.1　手動によるアプリケーションフェイルオーバーの設定

アプリケーションフェイルオーバー構成の NNMi を設定するには，次の手順を実行します。

1. アクティブサーバー（サーバー X）とスタンバイサーバー（サーバー Y）に NNMi をインストールする。



2. 「2.3　NNMi のライセンスを取得する」に記載されているように，各サーバーに恒久ライセンスを導入する。
3. 各サーバーでovstop コマンドを実行して NNMi をシャットダウンする。
4. nms-cluster.properties ファイルに含まれる指示を参考にして，サーバー X（アクティブ）およびサーバー Y（スタンバイ）のアプリケーションフェイルオーバー機能を設定する。

   次の手順を実行します。次の手順では，ファイルのテキストブロックの行のコメント（先頭の#!）を解除し，テキストを変更することを編集と呼びます。

   a　次のファイルを編集します。

   - Windows：%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
   - Linux：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

   b　NNMi クラスタに一意の名前を宣言します。アクティブサーバーとスタンバイサーバーが同じ名前を使用するように設定します。名前は英数字で指定してください。大小文字は区別されます。

   このパラメーターを指定することで，アプリケーションフェイルオーバー機能が有効化されます。

   ```
   com.hp.ov.nms.cluster.name=MyCluster
   ```

   c　nms-cluster.properties ファイルのcom.hp.ov.nms.cluster.member.hostnames パラメーターに，クラスタのすべてのノードのホスト名を追加します。

```
com.hp.ov.nms.cluster.member.hostnames = fqdn_for_active,fqdn_for_standby
```

重要

両方のノードのどちらかが複数の IPv4 アドレスを持つ場合は，com.hp.ov.nms.cluster.member.hostnames パラメーターに，各ノードの NNMi の通信に使用する IPv4 アドレスを記載してください。

5. NNMi の証明書（nnm-key.p12 ファイルおよびnnm-trust.p12 ファイル，または認証機関を使用する）を設定する。

   選択した方法に基づき，「10.3.5　アプリケーションフェイルオーバー環境での証明書の使用」に示されている指示を実行します。

重要

アプリケーションフェイルオーバー機能を設定するときには，両方のノードのnnm-trust.p12 ファイルをマージして，1 つのnnm-trust.p12 ファイルを作成する必要があります。選択した方法の指示を参照してください。

6. 次のファイルをサーバー X からサーバー Y にコピーする。

   コピーする前に，アプリケーションフェイルオーバー構成を解除するときのために元のファイルをバックアップしてください。

   - Windows：%NnmDataDir%shared¥nnm¥conf¥nnmcluster¥cluster.keystore
   - Linux：$NnmDataDir/shared/nnm/conf/nnmcluster/cluster.keystore

7. 両ノード間のクラスタ通信に使用する NIC を設定する。

   詳細については，「18.3.3　アプリケーションフェイルオーバー通信の設定」を参照してください。

重要

両方のノードのどちらかが複数の NIC を持つ場合は，必ず設定を実施してください。

8. サーバー X とサーバー Y の両方で次のコマンドを実行する。

```
nnmcluster
```

   各サーバーに，次のように表示されます。

```
================== 現在のクラスタ状態 =====================
状態ID: 0000000001000000005
日付/時間: 15 3 2011 - 09:37:58 (GMT+0900)
クラスタ名: ThisCluster (キー CRC:626,187,650)
自動フェールオーバー: Enabled
NNMデータベースの種類: 組み込み
NNMで設定済みのACTIVEノード: NO_ACTIVE
NNMの現在のACTIVEノード: NO_ACTIVE
クラスタメンバー:

 ローカル?  ノード タイプ 状態 OvStatus ホスト名/アドレス
```

```
-------  -------- ----- -------- ----------------------------
* REMOTE ADMIN     N/A   N/A      serverX.xxx.yyy.yourcompany.com/16.78.61.68:7800
(SELF)   ADMIN     N/A   N/A      serverY.xxx.yyy.yourcompany.com/16.78.61.71:7800
==========================================================
```

重要

コマンドを終了する場合は Enter キーを押下後，「quit」と入力してください。

画面には，サーバー X とサーバー Y の両方がリストされます。両方のノードの情報が表示されない場合，それらのノードはお互いに通信していません。手順を進める前に，次のことを確認して，修正してください。

- 次に示す両方のサーバーのクラスタ名「com.hp.ov.nms.cluster.name」に，同じクラスタ名を設定してください。
  - ・Windows：%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
  - ・Linux：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties
- キー CRC が，サーバー X とサーバー Y で異なっているかどうか。

  サーバー X とサーバー Y の両方で，次のファイルの内容を確認してください。
  - ・Windows：%NnmDataDir%shared¥nnm¥conf¥nnmcluster¥cluster.keystore
  - ・Linux：$NnmDataDir/shared/nnm/conf/nnmcluster/cluster.keystore

  異なっている場合は，手順 6.を実施してください。
- サーバー X またはサーバー Y のファイアウォールによって，ノードの通信が妨げられているかどうか。

  ノードが通信できる状態に設定してください。
- nnm-trust.p12 ファイルがマージされているかどうか。

  このエラーが表示されるのは，nnmcluster コマンドを実行した後です。
- com.hp.ov.nms.cluster.interface に指定した NIC から取得できる IP アドレスと com.hp.ov.nms.cluster.member.hostnames に指定したホスト名から解決できる IP アドレスを一致させてください。

  com.hp.ov.nms.cluster.interface は次のファイル内に指定します。
  - ・Windows：%NnmDataDir%Conf¥nnm¥props¥nms-cluster-local.properties
  - ・Linux：$NnmDataDir/conf/nnm/props/nms-cluster-local.properties

  com.hp.ov.nms.cluster.member.hostnames は次のファイル内に指定します。
  - ・Windows：%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
  - ・Linux：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties
- 相手サーバーの待機 IP アドレスと，相手サーバーとして指定した IP アドレスが一致しているかどうか。

  相手サーバーがクラスタ通信のために待機している IP アドレスと，クラスタ通信の相手サーバーとして指定した IP アドレスが一致しているかどうかを確認してください。

クラスタ通信のために使用するポートはデフォルト 7800 です。

待機している IP アドレスはnetstat コマンドで確認できます。

クラスタ通信の相手サーバーとして指定した IP アドレスは，nms-cluster.properties ファイルの com.hp.ov.nms.cluster.member.hostnames パラメーターで確認できます。

com.hp.ov.nms.cluster.member.hostnames にホスト名を指定している場合は，ホスト名から解決できる IP アドレスを確認してください。

- サーバー X とサーバー Y で，異なるオペレーティングシステムが実行されているかどうか。

  例えば，サーバー X で Linux オペレーティングシステムが実行され，サーバー Y で Windows オペレーティングシステムが実行されている場合などです。同じオペレーティングシステムが実行されている環境で設定してください。

- サーバー X とサーバー Y が，異なるバージョンの NNMi を実行しているかどうか。

  例えば，サーバー X が NNMi 11-00 を実行しており，サーバー Y が NNMi 11-00 の修正版を実行している場合などです。同じバージョンの NNMi をインストールした環境で設定してください。

9. サーバー X で，NNMi クラスタマネージャーを開始する。

```
nnmcluster -daemon
```

nnmcluster -daemon コマンドを NNMi 管理サーバー X で実行すると，NNMi クラスタマネージャーが次の起動ルーチンを実行します。

- NNMi 管理サーバー X をクラスタに接続します。
- ほかの NNMi 管理サーバーが存在しないことを検知します。
- NNMi 管理サーバー X はアクティブ状態に変わります。
- NNMi 管理サーバー X（アクティブサーバー）の NNMi サービスを開始します。
- データベースのバックアップを作成します。

詳細については，nnmcluster のリファレンスページを参照してください。

10. サーバー X がクラスタの最初のアクティブサーバーになるまで数分待ったあと，サーバー X でnnmcluster -display コマンドを実行する。

「ACTIVE_NNM_STARTING」または「ACTIVE_SomeOtherState」と表示されていることを確認してください。サーバー X がアクティブサーバーであることを確認するまで手順 11.に進まないでください。

11. サーバー Y で NNMi クラスタマネージャーを開始する。

```
nnmcluster -daemon
```

nnmcluster -daemon コマンドを NNMi 管理サーバー Y で実行すると，NNMi クラスタマネージャーが次の起動ルーチンを実行します。

- NNMi 管理サーバー Y をクラスタに接続します。
- NNMi 管理サーバー X が存在し，アクティブな状態であることが検出されます。画面に「STANDBY_INITIALIZING」と表示されます。

- NNMi 管理サーバー Y のデータベースバックアップが NNMi 管理サーバー X のバックアップと比較されます。一致しない場合は，新しいデータベースバックアップが NNMi 管理サーバー X（アクティブ）から NNMi 管理サーバー Y（スタンバイ）に送信されます。画面に「STANDBY_RECV_DBZIP」と表示されます。
- NNMi 管理サーバー Y は，スタンバイ状態に該当するバックアップに最低限必要となる，トランザクションログの最小限のセットを受信します。画面に「STANDBY_RECV_TXLOGS」と表示されます。
- NNMi 管理サーバー Y は待機状態になり，新しいトランザクションログとハートビート信号を NNMi 管理サーバー X から受信し続けます。画面に「STANDBY_READY」と表示されます。

詳細については，nnmcluster のリファレンスページを参照してください。

12. フェイルオーバーが発生した場合，サーバー X の NNMi コンソールは機能しなくなる。サーバー X の NNMi コンソールセッションを閉じて，サーバー Y（新たにアクティブになったサーバー）にサインインする。

    NNMi ユーザーに，サーバー X（アクティブサーバー）とサーバー Y（スタンバイサーバー）への 2 つのブックマークを登録するように指示します。フェイルオーバーが発生すると，ユーザーはサーバー Y（スタンバイ NNMi 管理サーバー）に接続できます。

13. サーバー X とサーバー Y の両方にトラップを送信するように，NNMi の監視対象機器の設定を変更する。

    サーバー X（アクティブ）が実行している間，サーバー X は転送されたトラップを処理し，サーバー Y（スタンバイ）はそのトラップを無視します。

18.3.2 NNMi クラスタセットアップウィザードを使用したアプリケーションフェイルオーバーの設定

NNMi クラスタセットアップウィザードは，アプリケーションフェイルオーバーで使用する NNMi 内のクラスタの設定プロセスを自動化します。ウィザードでは，次の操作ができます。

- クラスタノードの指定および検証を行う
- クラスタのプロパティおよびポートを定義する
- 両方のノードのnnm-key.p12 ファイルおよびnnm-trust.p12 ファイルの内容をマージして，1 つのnnm-key.p12 ファイルおよびnnm-trust.p12 ファイルにする

1. サポートされる Web ブラウザに次の URL を入力して，クラスタセットアップウィザードを起動する。

```
http://<NNMiserver>:<port>/cluster
```

   - <NNMiserver>は，NNMi ホストの値です。
   - <port>は，NNMi ポートの値です。

2. システムの［ユーザー名］と［パスワード］を入力して［ログイン］ボタンをクリックし，NNMi にログインする。

3. [ローカルホスト名] と [リモートクラスタノード] の値を入力してクラスタノードを定義し，[次へ] をクリックする。

> **重要**
>
> 両方のノードのどちらかが複数の IPv4 アドレスを持つ場合は，[ローカルホスト名] と [リモートクラスタノード] に，各ノードの NNMi の通信に使う IPv4 アドレスを記載してください。

4. [通信結果] ページで，通信の検証結果を確認する。

   エラーが発生した場合は [前へ] をクリックして問題を修正します。エラーが発生しなかった場合は [次へ] をクリックします。

   緑のステータスメッセージは，リモートクラスタノードに正常に接続されたことを示します。

5. [クラスタプロパティを定義] ページで，[クラスタ名] を入力して [バックアップ周期(時間)] を定義する。

   [クラスタ名] は，英数字で指定してください。次に自動フェイルオーバーを有効にするかどうかを指定します。[次へ] をクリックします。

6. [クラスタポートを定義] ページで，[開始クラスタポート] と [ファイル転送ポート] の値を入力する。

   NNMi クラスタでは，[開始クラスタポート] で始まる 4 個の連続したポートが使用されます。

7. [次へ] をクリックする。

8. [要約] ページで，入力した情報の概要を確認する。

   戻って設定情報を変更する場合は [前へ] をクリックします。変更しない場合は [コミット] をクリックしてクラスタ設定を保存します。

   最後の概要には，設定が成功したかどうかが示されます。

9. 両方のノードで`ovstop` を実行して，両方のノードの NNMi を直ちに停止する。

10. 両ノード間のクラスタ通信に使用する NIC を設定する。

    詳細については，「18.3.3　アプリケーションフェイルオーバー通信の設定」を参照してください。

> **重要**
>
> 両方のノードのどちらかが複数の NIC を持つ場合は，必ず設定を実施してください。

11. 両方のノードで`nnmcluster` コマンドを実行して，2 つのノードをクラスタ構成にできることを確認する。

    ノードをクラスタ構成にできない場合は，「18.3　アプリケーションフェイルオーバー構成の NNMi を設定する」を参照してください。

12. `nnmcluster -daemon` コマンドを使用して，アクティブにするノード上の NNMi を起動する。

    NNMi が ACTIVE をレポートするまで待機します。詳細は「18.3　アプリケーションフェイルオーバー構成の NNMi を設定する」を参照してください。

13. `nnmcluster -daemon` コマンドを使用して，スタンバイノードを起動する。

18.3.3 アプリケーションフェイルオーバー通信の設定

インストール時に，NNMi はシステム上のすべてのネットワークインタフェースカード（NIC）に対してクエリーを実行し，クラスタ通信に使用する NIC を特定します。システムに複数の NIC が存在する場合，次の手順を実行して，nnmcluster 操作に使用する NIC を選択できます。

> **重要**
>
> 両方のノードのどちらかが複数の NIC を持つ場合は，必ず設定を実施してください。

1. nnmcluster -interfaces を実行して，使用可能なすべてのインタフェースをリスト表示する。
   詳細については，nnmcluster のリファレンスページを参照してください。
2. 次のファイルを編集する。
   - Windows：%NnmDataDir%Conf¥nnm¥props¥nms-cluster-local.properties
   - Linux：$NnmDataDir/conf/nnm/props/nms-cluster-local.properties
3. 次のような内容のテキストが含まれる行を見つける。

```
com.hp.ov.nms.cluster.interface=<値>
```

4. 必要に応じて値を変更する。
   インタフェースの値は，有効なインタフェースである必要があります。インタフェースの値が無効の場合は，クラスタが開始できない場合があります。
   設定する値は手順 1.のnnmcluster -interfaces で出力された eth3 などの値です。
   Windows の場合は，eth3 などの値に続いてシステムのインタフェースの説明が表示されます。
   ipconfig /all コマンドなどによって，インタフェースの説明を確認することで，使用するインタフェースと eth3 などの値を対応させてください。
   Linux の場合は，インタフェースの名前が表示されます。ifconfig コマンドなどによって，使用するインタフェースの名前を確認してください。

> **重要**
>
> アクティブサーバーとスタンバイサーバーは，クラスタ通信に使用する NIC の持つすべての IPv4 アドレスで，アプリケーションフェールオーバーを構成する相手の NNMi 管理サーバーと通信ができる必要があります。相手と通信できない IPv4 アドレスが存在する場合，アプリケーションフェールオーバーの構築に失敗したり，構築済みのアプリケーションフェールオーバー環境の動作が不正になったりします。

5. nms-cluster-local.properties ファイルを保存する。
   com.hp.ov.nms.cluster.interface パラメーターを使用すると，NNMi の管理者はnnmcluster の通信に使用する通信インタフェースを選択できるようになります。com.hp.ov.nms.cluster.interface に指定した NIC から取得できる IP アドレスとcom.hp.ov.nms.cluster.member.hostnames に指定したホスト名から解決できる IP アドレスを一致させるように設定してください。複数の IP アドレスが同一のホ

スト名に名前解決される環境では，com.hp.ov.nms.cluster.member.hostnames パラメーターにホスト名ではなく，アプリケーションフェイルオーバーの通信に使用する IP アドレスを設定してください。com.hp.ov.nms.cluster.member.hostnames パラメーターは，次のファイルで設定します。

- Windows：%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
- Linux：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

18.4 アプリケーションフェイルオーバー機能の使用

両方の NNMi 管理サーバーでクラスタマネージャーが実行しているため（アクティブサーバーとスタンバイサーバー），クラスタマネージャーを使用してクラスタのステータスを表示できます。クラスタマネージャーには 3 つのモードがあります。

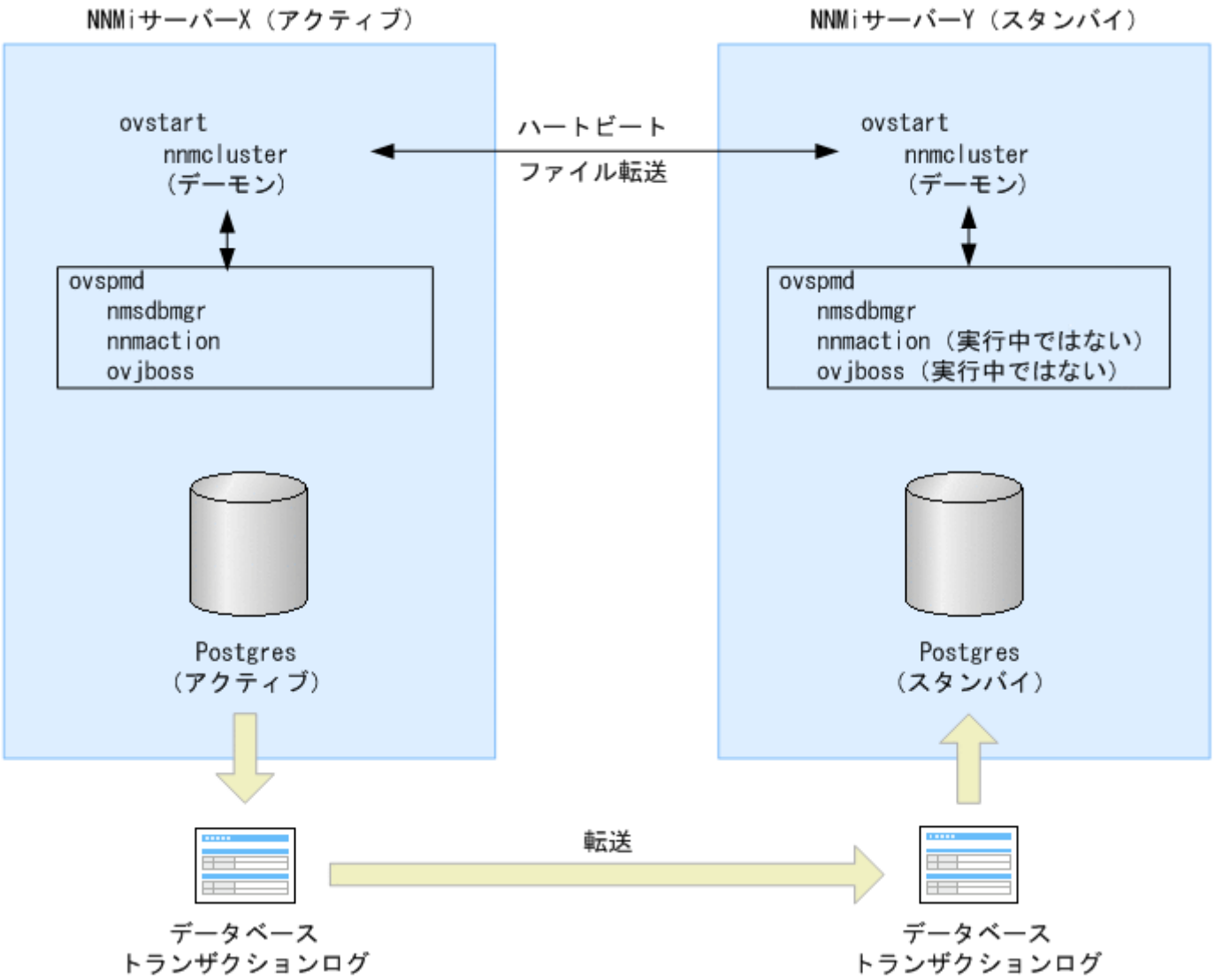
- デーモンモード：クラスタマネージャーのプロセスはバックグラウンドで実行し，`ovstop` および`ovstart` コマンドを使用して NNMi サービスを開始および停止します。
- インタラクティブモード：クラスタマネージャーは，NNMi 管理者がクラスタの属性を表示および変更できるインタラクティブセッションを実行します。例えば，NNMi 管理者はこのセッションを使用して，アプリケーションフェイルオーバー機能を有効または無効にしたり，デーモンプロセスをシャットダウンしたりできます。
- コマンドラインモード：NNMi 管理者は，コマンドプロンプトでクラスタの属性を表示および変更します。

詳細については，`nnmcluster` のリファレンスページを参照してください。

18.4.1 アプリケーションフェイルオーバーの動作

次の図は，NNMi データベースを使用した 2 つの NNMi 管理サーバーのアプリケーションフェイルオーバー設定を示します。この章の以降のセクションについて，この図を参照してください。

図 18-1　アプリケーションフェイルオーバーの設定（NNMi データベース）



クラスタからスタンバイサーバーを削除し，そのサーバーをスタンドアロンサーバーとして動作させて，次にそのサーバーを再度クラスタに戻すと，データベースのエラーになる場合があります。この場合，コマンドラインから次のコマンドを実行します。

```
nnmcluster dbsync
```

NNMi 11-00 には，アプリケーションフェイルオーバー内にストリーミングレプリケーション機能が含まれており，スタンバイサーバーとアクティブサーバーが同期した状態のまま，データベーストランザクションがアクティブサーバーからスタンバイサーバーに送信されます。これによって，（以前のバージョンの NNMi のように）フェイルオーバーでデータベーストランザクションログをスタンバイサーバーにインポートする必要がなくなり，スタンバイサーバーがアクティブサーバーを引き継ぐのに要する時間が大幅に短縮されます。この機能には，データベースバックアップファイルが必要な場合だけノード間で送信されるという利点もあり，データベーストランザクションファイルの通常の転送で，大きなデータベースバックアップファイルを送信する頻度が少なくなります。

アクティブサーバーとスタンバイサーバーの両方を開始すると，スタンバイサーバーはアクティブサーバーを検知してアクティブサーバーにデータベースのバックアップをリクエストしますが，ネットワーク監視は開始しません。このデータベースのバックアップは 1 つの ZIP ファイルとして保存されます。すでにスタンバイサーバーに以前のクラスタ接続から得た ZIP ファイルがあり，そのファイルがすでにアクティブサーバーと同期されていることを確認した場合は，ファイルは再送されません。

アクティブサーバーとスタンバイサーバーの両方が実行している間，アクティブサーバーは定期的にデータベースのトランザクションログをスタンバイサーバーに送信します。nms-cluster.properties ファイルのcom.hp.ov.nms.cluster.timeout.archive パラメーターの値を変更すると，このデータの転送頻度を変更できます。これらのトランザクションログはスタンバイサーバーに蓄積されるため，スタンバイからアクティブになったときにすぐに利用できます。

標準のデータの転送頻度は，次のとおりです。

- 6 時間ごとに，データベースのフルバックアップを転送します。
- 15 分ごとに，トランザクションログ（データベースの更新情報）を転送します。なお，データベースが大量に更新された場合は，より短い間隔で転送する場合があります。
  データが転送されるまでの間に更新した内容は引き継がれません。

スタンバイサーバーがアクティブサーバーからデータベースの完全バックアップを受信すると，その情報を NNMi データベースに取り込みます。また，recovery.conf ファイルを作成して受信したすべてのトランザクションログを取り込んでからでないと，ほかのサービスがデータベースを使用できません。そのことを NNMi データベースに知らせます。何らかの理由でアクティブサーバーが利用できなくなると，スタンバイサーバーは NNMi サービスを開始するovstart コマンドを実行してアクティブになります。スタンバイサーバーは，残りの NNMi サービスを開始する前に，トランザクションログをインポートします。

データベースのファイルは，次のディレクトリ下に格納されます。

- Windows：%NnmDataDir%shared¥nnm¥databases¥
- Linux：$NnmDataDir/shared/nnm/databases/

アプリケーションフェイルオーバー構成では，上のディレクトリ下に三つのディレクトリ（Postgres，Postgres_standby，Postgres_OLD）が作成されます。それぞれの用途は次のとおりです。

- Postgres：稼働中またはスタンバイ用に受信したデータベース本体のデータを格納
- Postgres_standby：アクティブサーバーからスタンバイサーバーへ転送・受信したデータを格納
- Postgres.OLD：スタンバイサーバーがデータ受信時の旧Postgres データを退避するために使用

アクティブサーバーに障害が発生すると，スタンバイサーバーは，ディスカバリとポーリングアクティビティを開始します。このようにシステムを切り替えることによって，障害が発生したシステムの診断と修理を行う間，NNMi はネットワークを監視およびポーリングし続けます。

重要

- NNMi ではアプリケーションフェイルオーバー構成でのフェイルオーバー後に再同期が行われるためステータスおよびインシデントの更新が遅延する可能性があります。
- この再同期中に次のメッセージが表示されても問題はありません。

Causal Engine のキューサイズが大きいため，ステータスおよびインシデントの更新が遅延しています。これは，アプリケーションフェイルオーバー構成でのフェイルオーバー，バックアップの復元，または手動による再同期のあとに再同期が行われることが原因で発生する可能性があります。

18.4.2 アプリケーションフェイルオーバーのシナリオ

アクティブ NNMi 管理サーバーがハートビートを送信しなくなり，フェイルオーバーが発生してしまう原因には幾つかあります。

ここでは障害発生時の NNMi 管理サーバーのフェイルオーバーについて，障害発生時の状況を場合分けしたシナリオを使用して説明します。

表 18-1 想定障害とシナリオの対応

想定する障害		障害の発生個所	
		アクティブ側	スタンバイ側
サーバー	サーバーダウン※1	シナリオ 1	シナリオ 6
	OS の停止	シナリオ 2	シナリオ 6
プロセス	nnmcluster の停止	シナリオ 3	シナリオ 6
	nnmcluster 以外の停止	シナリオ 5	該当なし※2
ネットワーク	相手と通信不可	シナリオ 4	シナリオ 4

注※1 サーバーダウンはハード障害や OS 障害などで停止した場合を想定しています。

注※2 スタンバイサーバーの NNMi は停止しているため，該当する場合はありません。

(1) フェイルオーバーが発生する場合

次のシナリオ 1～3 の場合，自動フェイルオーバーが有効になっていれば NNMi がスタンバイサーバーへフェイルオーバーし，NNMi のネットワーク監視が継続されます。

- シナリオ 1：アクティブ NNMi 管理サーバーに障害が発生した。

  アクティブサーバーがハード障害や OS 障害によって OS のシャットダウン処理が行われないで停止した場合です。スタンバイサーバーは相手が停止したことを検知し，アクティブになって自動的に NNMi

を起動し，ネットワーク監視は継続されます。元のアクティブサーバーは，起動するとスタンバイとして動作します。

- シナリオ 2：システム管理者がアクティブな NNMi 管理サーバーをシャットダウンまたはリブートした。

  アクティブサーバーが OS のシャットダウン処理を行って停止した場合です。スタンバイサーバーは相手が停止したことを検知し，アクティブになって自動的に NNMi を起動し，ネットワーク監視は継続されます。元のアクティブサーバーは，起動するとスタンバイとして動作します。

  ただし，NNMi 管理サーバーが Linux オペレーティングシステムの場合は，OS の停止時に終了スクリプトが実行されると，`ovstop` コマンドが自動的に実行されるため，アプリケーションフェイルオーバーが無効になり，フェイルオーバーが発生しません。

- シナリオ 3：NNMi 管理者がクラスタをシャットダウンした。

  クラスタマネージャー（nnmcluster プロセス）が，管理者の操作または何らかの要因で停止した場合です。スタンバイサーバーは相手が停止したことを検知し，アクティブになって自動的に NNMi を起動し，ネットワーク監視は継続されます。

重要

アクティブサーバーの nnmcluster だけが何らかの要因で停止して，ほかの NNMi のプロセスが残ったまま動作している状態になった場合，シナリオ 3 の障害と同様の状態になり，元のアクティブサーバーと新たなアクティブサーバーの 2 台で NNMi が動作する状況になる場合があります。この場合は，元のアクティブサーバーの OS を再起動して回復してください。

(2) フェイルオーバーが発生しない場合

「(1) フェイルオーバーが発生する場合」で挙げたシナリオに該当しない現象が起こった場合，フェイルオーバーは発生しません。例えば，次のような場合があります。

- シナリオ 4：アクティブ NNMi 管理サーバーとスタンバイ NNMi 管理サーバーの間のネットワーク接続に障害が発生した。

  両サーバーの通信ができなくなった場合です。クラスタマネージャー（nnmcluster プロセス）の間のハートビート通信ができないため， 次の状態に陥ります。

  - アクティブサーバーは相手が停止したと検知し，そのまま動作します。
  - スタンバイサーバーも相手が停止したと検知し，アクティブとなって NNMi を起動します。

  シナリオ 4 では，両方の NNMi 管理サーバーがアクティブな状態で稼働します。ネットワークデバイスが復旧すると，2 つの NNMi 管理サーバーは自動的にネゴシエーションしてアクティブサーバーとして稼働するサーバーを決定し，片方のサーバーがスタンバイとなり NNMi を停止します。

  なお，両方のサーバーがアクティブになる状態は，クラスタソフトでの HA 構成の場合はスプリットブレインと呼ばれる問題が発生します。しかし，アプリケーションフェイルオーバーの場合は仕組みが異なるため，通信障害が回復すると次のように問題なく回復します。

  - 通信が回復すると片方がスタンバイサーバーとなり通常の構成に回復します。

- アプリケーションフェイルオーバーのデータベースは，共有ディスクを使用しないで，スタンバイ側がアクティブ側へデータベースの転送を要求してデータベースを同期する方式です。このため，両方のサーバーで NNMi が動作しても整合性に問題は発生しません。

- シナリオ 5：NNMi のプロセスが停止した。

  何らかの要因でクラスタマネージャー（nnmcluster プロセス）以外の NNMi のプロセスが停止しても，フェイルオーバーは発生しません。

  クラスタマネージャーのハートビート通信によって相互にサーバーの動作監視をしていますが，自サーバー内の NNMi のプロセスの監視は行っていないため，このような動作となります。

  NNMi プロセスが停止した時にフェイルオーバーさせたい場合は，クラスタソフトによる HA 構成を使用してください。

- シナリオ 6：スタンバイサーバーで障害が発生した。

  スタンバイサーバー側で，シナリオ 1～3 の障害（サーバーダウン，OS の停止または nnmcluster プロセスの停止）が発生した場合です。この場合，スタンバイサーバーがクラスタ構成のメンバーからは外れますが，アクティブサーバーの NNMi は動作し続け，NNMi によるネットワーク監視は継続できます。

重要

スタンバイサーバーがない状態（片系運用）になったことは通知されません。

18.4.3 アプリケーションフェイルオーバー構成の NNMi 管理サーバーで使用する ovstart および ovstop コマンド

アプリケーションフェイルオーバーが設定された NNMi 管理サーバーで`ovstop` コマンドおよび`ovstart` コマンドを使用した場合，実際には NNMi は次のコマンドを実行します。これらは NNMi の起動や停止の完了を待たないですぐに終了します。

- `ovstart: nnmcluster -daemon`
- `ovstop: nnmcluster -disable -shutdown`

メモ

- `ovstop` コマンドを実行すると，NNMi はスタンバイサーバーにフェイルオーバーしません。`ovstop` コマンドは，メンテナンスによる一時的な停止をサポートするように設計されています。フェイルオーバーを手動で行うには，`ovstop` コマンドに`-failover` オプションを使用します。詳細については，`ovstop` のリファレンスページを参照してください。
- `ovstop` コマンドを実行すると`nnmcluster` の`-disable` オプションが指定されているため，自動フェイルオーバーが無効化されますので注意してください。フェイルオーバーの有

効無効を確認するにはnnmcluster -display で「自動フェールオーバー」の項を確認します。フェイルオーバーを有効化するにはnnmcluster -enable を実行します。

ovstop コマンドに使用する次のオプションは，アプリケーションフェイルオーバークラスタに構成されたNNMi 管理サーバーで使用します。

- ovstop -failover：ローカルのデーモンモードのクラスタプロセスを停止し，スタンバイ NNMi 管理サーバーに強制的にフェイルオーバーします。以前にフェイルオーバーモードが無効にされている場合は，このコマンドで有効になります。このコマンドはnnmcluster -enable -shutdown と同等です。
- ovstop -nofailover：フェイルオーバーモードを無効にし，ローカルのデーモンモードのクラスタプロセスを停止します。フェイルオーバーは行われません。このコマンドはnnmcluster -disable -shutdown と同等です。
- ovstop -cluster：アクティブサーバーとスタンバイサーバーを停止し，これらをクラスタから削除します。このコマンドはnnmcluster -halt と同等です。

重要

Linux オペレーティングシステムを実行している NNMi 管理サーバーで OS の停止時に NNMi の終了スクリプトが実行されると，ovstop コマンドが自動的に実行され，アプリケーションフェイルオーバーが無効になります。メンテナンス中にアプリケーションフェイルオーバーを制御するには，OS の停止コマンドを実行する前に，nnmcluster -acquire コマンドとnnmcluster -relinquish コマンドを使用してアクティブサーバーとスタンバイサーバーを目的の動作に設定します。詳細については，nnmcluster のリファレンスページを参照してください。

18.4.4 アプリケーションフェイルオーバーのインシデント

nnmcluster プロセスまたはnnmcluster コマンドを使用するユーザーが，ノードをアクティブとして開始すると，NNMi ではそのたびに次のどちらかのインシデントが生成されます。

- *NnmClusterStartup*：NNMi クラスタは，アクティブサーバーがない状態で開始されました。したがって，このサーバーはアクティブ状態で起動されました。このインシデントの重大度は「正常域」です。
- *NnmClusterFailover*：NNMi クラスタでアクティブサーバーの障害が検出されました。そのため，スタンバイサーバーがアクティブサーバーになり，そのノードで NNMi サービスが開始されました。このインシデントの重大度は「重要警戒域」です。

18.5　フェイルオーバーの問題解決後の設定

アクティブノードで障害が発生し，スタンバイノードがアクティブノードとして機能している場合，以前のアクティブノードで問題を解決したあとで，元の設定に戻すことができます。

次の手順を実行します。

1. 以前のアクティブノードで問題を解決する。
2. 目的のアクティブノードで次のコマンドを実行し，元の設定に戻す。

```
nnmcluster -acquire
```

詳細については，nnmcluster のリファレンスページを参照してください。

18.6 アプリケーションフェイルオーバーを無効にする

アプリケーションフェイルオーバーを設定し，数日間使用したあとに，完全に無効化するとします。次の情報は，アプリケーションフェイルオーバーを完全に無効にする方法を説明しています。アプリケーションフェイルオーバークラスタに構成された，アクティブおよびスタンバイ NNMi 管理サーバーでのアクションを含め，次の指示に従ってください。

1. アクティブ NNMi 管理サーバーで`nnmcluster -enable` コマンドを実行する。
2. アクティブ NNMi 管理サーバーで`nnmcluster -shutdown` コマンドを実行する。
3. 既存のスタンバイ NNMi 管理サーバーが新しくアクティブ NNMi 管理サーバーになるまで数分待つ。
4. 新しいアクティブ（以前のスタンバイ）NNMi 管理サーバーで`nnmcluster -display` コマンドを実行する。
5. 表示された結果で，`ACTIVE_NNM_RUNNING` ステータスを検索する。

   `ACTIVE_NNM_RUNNING` ステータスを確認できるまで，手順 4.を繰り返します。
6. 新しいアクティブ（以前のスタンバイ）NNMi 管理サーバーで`nnmcluster -shutdown` コマンドを実行する。
7. 新しいアクティブ（以前のスタンバイ）で`nnmcluster -display` コマンドを実行する。

   コマンド実行結果でノードタイプ列が DAEMON の行がなくなるまで繰り返し実行します。
8. クラスタに構成されている両方の NNMi 管理サーバーで，次のファイルを編集する。
   - Windows：`%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties`
   - Linux：`$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties`
9. 両方の NNMi 管理サーバーの`com.hp.ov.nms.cluster.name` オプションをコメントにし（行の先頭に`#!` を付ける），各ファイルを保存する。
10. 両方の NNMi 管理サーバーの次のファイルを編集する。
    - Windows：`%NnmDataDir%shared¥nnm¥databases¥Postgres¥postgresql.conf`
    - Linux：`$NnmDataDir/shared/nnm/databases/Postgres/postgresql.conf`

    `postgresql.conf` を編集する場合は，改行コードが LF（0x0A）だけのファイルを編集できるエディタを使用してください（Windows の場合は，メモ帳は使用しないで，ワードパットを使用する。Linux の場合は vi を使用する）。
11. 各ファイルで，次の行を削除する。

    次の例は，Windows の NNMi 管理サーバーの表示例です。サーバーによって，表示がやや異なります。

```
# The following lines were added by the NNM cluster.
archive_command = 'nnmcluster.exe -archive  -logCONFIG "%p" "file:/C:/ProgramData/Hitachi
/Cm2NNMi/shared/nnm/databases/Postgres_standby/TxWALs_send/%f"'
archive_timeout = 900
max_wal_senders = 4
archive_mode = 'on'
wal_level = 'hot_standby'
```

```
hot_standby = 'on'
wal_keep_segments = 500
listen_addresses = 'localhost,XX.XX.XX.XX'
```

必ず変更を保存してください。

12. Windows NNMi 管理サーバーの場合，［サービス(ローカル)］コンソールに移動し，各サーバーで次の手順を実行する。
    a NNM Cluster Manager の［スタートアップの種類］を［無効］に設定します。
    b NNM Process Manager の［スタートアップの種類］を［自動］に設定します。
13. 次のトリガーファイルを作成する。

    このファイルは，Postgres にスタンバイモードでの実行を中止し，完全に実行するように指示します。

    - Windows：`%NnmDataDir%tmp¥postgresTriggerFile`
    - Linux：`$NnmDataDir/tmp/postgresTriggerFile`
14. 両方の NNMi 管理サーバーで`ovstart` コマンドを実行する。
15. 両方の NNMi 管理サーバーが正常に開始したら，スタンバイおよびアクティブ NNMi 管理サーバーから次のディレクトリを削除する。
    - Windows：`%NnmDataDir%shared¥nnm¥databases¥Postgres_standby`
    - Linux：`$NnmDataDir/shared/nnm/databases/Postgres_standby`

    メモ

    このディレクトリはデフォルトのディレクトリで，`nms-cluster.properties` ファイルにある`com.hp.ov.nms.cluster.archivedir` パラメーターの値です。この手順では，この値が変更されていないことを前提としています。`nms-cluster.properties` ファイルの `com.hp.ov.nms.cluster.archivedir` パラメーターの値を変更した場合は，変更後の新しい値に相当するディレクトリを削除します。
16. スタンバイおよびアクティブ NNMi 管理サーバーから次のディレクトリを削除する。
    - Windows：`%NnmDataDir%shared¥nnm¥databases¥Postgres.OLD`
    - Linux：`$NnmDataDir/shared/nnm/databases/Postgres.OLD`

18.7　管理タスクとアプリケーションフェイルオーバー

次は，NNMi 管理サーバーへのパッチ適用や再起動などの管理タスクを行うときに，アプリケーションフェイルオーバーを効果的に管理する方法を説明します。

18.7.1　NNMi のバージョンアップ（修正版の適用を含む）

アプリケーションフェイルオーバー構成の NNMi 管理サーバーをバージョンアップする場合は「24.4　アプリケーションフェイルオーバー構成の NNMi 13-10 へのアップグレード」を参照してください。

18.7.2　NNMi の起動と停止および再起動

(1)　NNMi の起動と停止

アプリケーションフェイルオーバー構成の NNMi は，`ovstart` コマンドで起動，または`ovstop` コマンドで停止を行う際に，`nnmcluster` コマンドに置き換えられて実行します。置き換え後のコマンドは，「18.4.3 アプリケーションフェイルオーバー構成の NNMi 管理サーバーで使用する ovstart および ovstop コマンド」を参照してください。

アプリケーションフェイルオーバー構成の NNMi は，起動時にサーバーがアクティブになるかスタンバイになるかが自動的に調整され，先に nnmcluster を起動したサーバーがアクティブになります。起動時の動作については，「18.4　アプリケーションフェイルオーバー機能の使用」を参照してください。

起動時の処理が完了すると次の通常運用時の状態になります。サーバーの状態は，`nnmcluster` コマンドをオプションなしで実行（インタラクティブモード）または`nnmcluster -display` コマンドを実行して State 列の表示を参照して確認します。

＜通常運用時の状態＞

アクティブサーバー：ACTIVE_NNM_RUNNING

スタンバイサーバー：STANDBY_READY

NNMi の起動時はサーバーが上記の通常運用時の状態になることを確認してください。主なサーバーの状態には次の種類があります。

状態（State の表示）	役割	説明
ACTIVE_NNM_STARTING	アクティブ	NNMi の起動処理中です
ACTIVE_DB_BACKUP	アクティブ	NNMi の DB のバックアップ処理中です
ACTIVE_NNM_RUNNING	アクティブ	NNMi が稼働している状態です
STANDBY_RECV_DBZIP	スタンバイ	アクティブ側の NNMi から DB を転送中です

状態（State の表示）	役割	説明
STANDBY_READY	スタンバイ	スタンバイとして準備完了となった NNMi の状態です

アプリケーションフェイルオーバーの運用操作，例えばアクティブとスタンバイの切り替えなどを行う場合は，通常運用時の状態になっていることを確認してから行ってください。この状態になる前にフェイルオーバーをした場合，サーバー間が正しく同期できずに NNMi の起動が失敗して ACTIVE_NNM_FAILED の状態になる場合があります。この場合は両サーバーを停止してから起動を行ってください。データベースの問題で起動できない場合は，データベースをリセットし，バックアップデータをリストアしてから再起動してください。

(2) NNMi の再起動

スタンバイ NNMi 管理サーバーは，いつでも再起動でき，再起動に関する特別な指示はありません。スタンバイおよびアクティブの両方の NNMi 管理サーバーを再起動する場合，アクティブ NNMi 管理サーバーを先に再起動します。

アクティブまたはスタンバイ NNMi 管理サーバーを再起動するには，次の手順を実行します。

1. NNMi 管理サーバーで`nnmcluster -disable` コマンドを実行し，アプリケーションフェイルオーバー機能を無効にする。
2. 次のコマンドを実行して，NNMi 管理サーバーを再起動する。

```
ovstop
ovstart
```

3. NNMi 管理サーバーで`nnmcluster -enable` コマンドを実行し，アプリケーションフェイルオーバー機能を有効にする。

> **重要**
>
> NNMi の`TrapReceiver` プロセス，およびそのフェイルオーバーとの関連に関する重要情報については，「21.16　NNMi NmsTrapReceiver プロセス」を参照してください。

通信障害後のアプリケーションフェイルオーバーの制御

2 つのクラスタノード間の通信障害が解決したあとは，その通信障害が発生するまでに最も長時間動作していた（つまり以前にアクティブだった）NNMi 管理サーバーが，アクティブサーバーに指定されます。

(3) NNMi のフェイルオーバー

アプリケーションフェイルオーバー構成のシステムでサーバーの障害が発生した場合は，「18.4.2　アプリケーションフェイルオーバーのシナリオ」に示すように状況に応じて自動的にフェイルオーバーし，アクティブサーバーで NNMi が起動されます。

手動でアクティブサーバーを切り替えたい場合は，次の手順を実行します。

1. アクティブサーバーでovstop -failoverを実行する。

   NNMiが停止してからクラスタマネージャー（nnmcluster）が停止し，スタンバイサーバーが新たなアクティブサーバーとなってNNMiが起動します。

2. 手順1.を実行した状態では，操作を行った元のアクティブサーバーはクラスタ構成のメンバーから外れている。スタンバイサーバーとしてクラスタに参加するにはovstartを実行する。

   ovstartは，nnmcluster -daemonに置き換えられて実行します。

18.7.3 NNMiのバックアップとリストア

(1) NNMiのバックアップ

アプリケーションフェイルオーバー構成のNNMiは，通常のシステムと同様の手順でバックアップを実行できます。ただし，-forceオプション（強制的にバックアップに適した状態にする）は使用できませんので，事前にバックアップに適した状態にしてからバックアップをします。

アクティブサーバー側で，次の手順を実行してください。

1. バックアップに適した次の状態にする。

   nnmbackup.ovplを使用する場合

   オンラインバックアップの場合：NNMiサービスを起動状態にする

   オフラインバックアップの場合：NNMiサービスを停止状態にする

   nnmbackupembdb.ovplを使用する場合

   NNMiサービスを起動状態にする

2. バックアップを実行する。

   nnmbackup.ovplまたはnnmbackupembdb.ovplコマンドを実行します。

重要

- バックアップは，アクティブサーバー側（オフラインバックアップでNNMiを停止する場合は，直前まで稼働していたサーバー）で行ってください。
- あるサーバーセット上のNNMiアプリケーションフェイルオーバー環境で取得したバックアップを異なるサーバーセット上のNNMiアプリケーションフェイルオーバー環境で復元するには，NNMiアクティブおよびスタンバイシステム両方のバックアップを取得してください。

(2) NNMiのリストア

アクティブおよびスタンバイNNMi管理サーバーがアプリケーションフェイルオーバー構成の場合に，以前のバックアップからNNMiデータベースをリストアするには，次の手順を実行します。

1. アクティブ NNMi 管理サーバーでnnmcluster -halt コマンドを実行する。

   アクティブサーバーとスタンバイサーバーの両方の NNMi が停止します。nnmcluster コマンドをオプションなしで実行（インタラクティブモード）またはnnmcluster -display コマンドを実行し，NNMi サービスが停止したことを確認します。

2. アクティブおよびスタンバイ NNMi 管理サーバーの次のディレクトリを削除または移動する。

   Windows

   ```
   %NnmDataDir%shared¥nnm¥databases¥Postgres_standby
   %NnmDataDir%shared¥nnm¥databases¥Postgres.OLD
   ```

   Linux

   ```
   $NnmDataDir/shared/nnm/databases/Postgres_standby
   $NnmDataDir/shared/nnm/databases/Postgres.OLD
   ```

3. アクティブ NNMi 管理サーバーでデータベースをリストアする。

   a. アプリケーションフェイルオーバー構成の設定を一時的に解除します。

      次のファイルのクラスタ名「com.hp.ov.nms.cluster.name」をコメントに（行の先頭に #! を付ける）してください。

      - Windows：%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
      - Linux：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

   b. 通常どおり，データベースをリストアします。nnmrestore.ovpl またはnnmrestoreembdb.ovpl を-force オプションを指定して実行します。-force オプションを指定してリストアに必要なサービスが起動したあと，リストアが実行されます。これらのコマンドについては，「20.3　NNMi データをリストアする」を参照してください。

      nnmbackup.ovpl でバックアップしたデータをリストアした場合は，手順 a の変更がリストアしたファイルで上書きされているおそれがあるため，もう一度手順 a を実施してください。

   c. アクティブ NNMi 管理サーバーでovstop コマンドを実行します。手順 b でリストア処理のために起動したサービスが停止します。

   d. アプリケーションフェイルオーバー構成を再設定します。

      次のファイルでクラスタ名「com.hp.ov.nms.cluster.name」のコメントを解除（手順 a で付けた#! を削除）してください。

      - Windows：%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
      - Linux：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

4. アクティブ NNMi 管理サーバーでovstart コマンドを実行する。

5. アクティブ NNMi 管理サーバーが新しいバックアップ（アクティブとスタンバイが同期を取るための ZIP ファイル）を生成するまで待つ。

   この手順が完了したことを確認するには，nnmcluster -display コマンドを実行し，ACTIVE_NNM_RUNNING メッセージを検索します。

6. スタンバイ NNMi 管理サーバーでovstart コマンドを実行する。

スタンバイ NNMi 管理サーバーは新しいバックアップ（手順 5.で作成された ZIP ファイル）をコピーして抽出します。この手順が完了したことを確認するには，nnmcluster -display コマンドを実行し，STANDBY_READY メッセージを検索します。

(3)　異なるサーバーセット上の NNMi フェイルオーバー環境にリストア

異なるサーバーセット上の NNMi フェイルオーバー環境を復元するには，NNMi アクティブおよびスタンバイシステム両方のバックアップを取得し，必要なサーバー上でそれらを復元するとともに，所定のプロパティファイルでホスト名を変更する必要があります。

NNMi フェイルオーバー環境を復元するには，次の手順を実行します。

1. ソースフェイルオーバー環境内のアクティブシステムとスタンバイシステムのすべての NNMi データの完全なオフラインバックアップを取得します。詳細については，「20.2　NNMi データをバックアップする」を参照してください。
2. バックアップファイルを，それぞれの送り先であるアクティブシステムとスタンバイシステムにコピーします。
3. バックアップデータの場合と同じバージョンおよびパッチレベルの NNMi をインストールします。
4. アクティブシステムとスタンバイシステムの両方で NNMi データを復元します。
   アクティブシステムとスタンバイシステムの両方で「(2)　NNMi のリストア」の手順 3.を実施してください。
5. アクティブおよびスタンバイ NNMi 管理サーバーの両方で，次の手順を実行します。
   a. アクティブおよびスタンバイ NNMi サーバーの両方のホスト名を確認します。
   b. 次のファイルを開きます。
      - Windows：%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
      - Linux：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties
   c. アクティブノードおよびスタンバイノードのホスト名をcom.hp.ov.nms.cluster.member.hostnames パラメーターに追加します。

```
com.hp.ov.nms.cluster.member.hostnames = fqdn_for_active, fqdn_for_standby
```

6. セキュア通信用の SSL 証明書を使用するように NNMi フェイルオーバー環境を設定します。詳細については，「10.　NNMi での証明書の使用」を参照してください。

18.7.4　NNMi の設定の変更

(1)　設定の変更

アプリケーションフェイルオーバー構成で，NNMi の設定を変更する場合について説明します。

(a) NNMi の設定ファイル

NNMi の設定ファイルを変更する場合は，アクティブサーバーとスタンバイサーバーの両方で設定変更を行い，同じ内容になるようにしてください。

> **重要**
>
> NNMi の再起動を伴う設定変更は，両方のサーバーを停止して設定変更を行ってください。

なお，NNMi の運用を停止しないで変更したい場合は，次の手順で設定変更を行ってください。各手順は nnmcluster -display を実行して，処理が完了していることを確認しながら，次の手順に進んでください。

1. サーバー A でovstop -failover を実行する。
   サーバー A が停止し，サーバー B がアクティブになります。
2. サーバー A で設定変更を行う。
3. サーバー A でovstart を実行し，スタンバイとして起動する。
4. サーバー B でovstop -failover を実行する。
   サーバー B が停止し，サーバー A がアクティブになります。
5. サーバー B で設定変更を行う。
6. サーバー B でovstart を実行し，スタンバイとして起動する。

(b) NNMi のデータベース

NNMi のデータベースはアクティブサーバーとスタンバイサーバーが自動的に同期を行っていますので，システム管理者の操作は不要です。

詳しい動作については，「18.4 アプリケーションフェイルオーバー機能の使用」を参照してください。

(2) データベースのリセット

「4.8 NNMi 設定およびデータベースのリセット」で説明されているデータベースのリセットを行う場合，一時的にアプリケーションフェイルオーバー構成を解除する必要があります。次の手順を行ってください。

1. （任意）現在の NNMi 設定を保存しておきたい場合は，アクティブサーバーで，次の手順を実行する。
   - nnmconfigexport.ovpl コマンドを使用して，NNMi 設定を XML ファイルに出力します。
   - nnmtrimincidents.ovpl コマンドを使用して，NNMi インシデントをアーカイブします。
2. アクティブサーバーで，nnmcluster -halt コマンドを実行する。
   アクティブサーバーとスタンバイサーバーの両方の NNMi が停止します。
   nnmcluster コマンドをオプションなしで実行（インタラクティブモード）またはnnmcluster -display コマンドを実行し，NNMi サービスが停止されたことを確認します。

3. アクティブサーバーで，NNMi のデータベースをリセットする。

a　アプリケーションフェイルオーバー構成の設定を一時的に解除します。

次のファイルのクラスタ名com.hp.ov.nms.cluster.name をコメントに（行の先頭に #! を付ける）してください。

- Windows：%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
- Linux：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

b　（任意）データベースのデータが削除される前に，必要に応じて次のコマンドで既存のデータベースをバックアップします。

```
nnmbackup.ovpl -type offline -target <backup_directory>
```

c　NNMi データベースを削除して再作成します。

```
nnmresetembdb.ovpl -nostart
```

d　アクティブ NNMi 管理サーバーでovstop コマンドを実行します。手順 c で起動したサービスが停止します。

e　アプリケーションフェイルオーバー構成を再設定します。

次のファイルでクラスタ名com.hp.ov.nms.cluster.nameのコメントを解除（手順 a で付けた#!を削除）してください。

- Windows：%NnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
- Linux：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

4. アクティブおよびスタンバイサーバーの次のディレクトリを削除または移動する。

Windows

```
%NnmDataDir%shared¥nnm¥databases¥Postgres_standby
%NnmDataDir%shared¥nnm¥databases¥Postgres.OLD
```

Linux

```
$NnmDataDir/shared/nnm/databases/Postgres_standby
$NnmDataDir/shared/nnm/databases/Postgres.OLD
```

5. アクティブ NNMi 管理サーバーでovstart コマンドを実行する。

この手順が完了したことを確認するには，nnmcluster -display コマンドを実行し，ACTIVE_NNM_RUNNING メッセージを検索します。

6. スタンバイ NNMi 管理サーバーでovstart コマンドを実行する。

この手順が完了したことを確認するには，nnmcluster -display コマンドを実行し，STANDBY_READY メッセージを検索します。

これで NNMi のデータベースはデフォルト設定だけになりました。

アクティブサーバーで NNMi の設定を行ってください。なお，手順 1.で保存した NNMi 設定をインポートするにはnnmconfigimport.ovpl コマンドを使用します。

18.7.5　NNMi データベースパスワードの変更

1.「18.6　アプリケーションフェイルオーバーを無効にする」を実施し，一時的にアプリケーションフェイルオーバー構成を無効にする。

2. それぞれの NNMi 管理サーバーで，パスワードを変更する。

   手順の詳細については，`nnmchangeembdbpw.ovpl` のリファレンスページを参照してください。

3.「18.3.2　NNMi クラスタセットアップウィザードを使用したアプリケーションフェイルオーバーの設定」を実施し，再度アプリケーションフェイルオーバー構成を有効にする。

18.8 ネットワークレイテンシ/帯域に関する考慮

NNMi アプリケーションフェイルオーバーは，クラスタのノード間で継続的なハートビート信号を交換することによって機能します。これには，NNMi データベース，データベーストランザクションログ，そのほかの NNMi 設定ファイルなどのデータファイルの交換に使用されるネットワークチャネルが使用されます。WAN（広域ネットワーク）に NNMi アプリケーションフェイルオーバーを導入する場合，パフォーマンスが高く，レイテンシが低い接続を使用することをお勧めします。

NNMi データベースは必ず圧縮されていますが，非常に容量が大きくなり，1GB 以上に増大することがあります。また，NNMi は，ビルトインバックアップインターバル（設定パラメーター，デフォルトは 6 時間）の間に膨大な数のトランザクションログを生成します。各トランザクションログのサイズは数メガバイトから最大 16MB になることもあります。(これらのファイルは圧縮されています)。次は，テスト環境から収集されたデータの例です。

```
Number of nodes managed: 15,000
Number of interfaces: 100,000
Time to complete spiral discovery of all expected nodes: 12 hours
Size of database: 850MB (compressed)
During initial discovery: ~10 transaction logs per minute (peak of ~15/min)
-----------------------------------------
10 TxLogs/minute X 12 hours = 7200 TxLogs @ ~10MB = ~72GB
```

これでは，ネットワークで送信するにはデータ量が多過ぎます。2 つのノード間のネットワークが NNMi アプリケーションフェイルオーバーの帯域幅の要求に応じられない場合，スタンバイサーバーへのデータベースファイルの送信に遅延が発生してしまいます。このため，アクティブサーバーに障害が発生した場合，潜在的なデータ喪失の可能性が高くなります。

同様に，2 つのノード間のネットワークのレイテンシが高いか信頼性が低い場合，ノード間で偽のハートビート喪失となります。例えば，ハートビート信号が直ちに応答しない場合に，スタンバイサーバーは，アクティブサーバーに障害が発生したと判断します。ハートビート喪失の検出に関与する要素には幾つかあります。NNMi は，ネットワークがアプリケーションフェイルオーバーのデータ転送の要求に応答できる限り，偽のフェイルオーバー通知を回避します。

18.8.1 アプリケーションフェイルオーバーと NNMi データベース

アプリケーションフェイルオーバーにデータベースを使用するように NNMi を設定すると，NNMi は次のように動作します。

1. アクティブサーバーがデータベースのバックアップを実行し，1 つの ZIP ファイルにデータを保存する。
2. ネットワークを通して，動作 1.の ZIP ファイルをスタンバイサーバーに送信する。
3. スタンバイサーバーは ZIP ファイルを展開し，データベースを設定して最初の起動でトランザクションログをインポートする。

4. アクティブサーバーのデータベースは，データベースアクティビティによって，トランザクションログを生成する。
5. アプリケーションフェイルオーバーでは，トランザクションログがネットワークを通してスタンバイサーバーに送信され，ディスクに蓄積される。
6. スタンバイサーバーがアクティブになると，NNMi が起動されて，データベースがネットワークを通してすべてのトランザクションログをインポートする。

   これに掛かる時間は，ファイル数，およびそのファイルに保存されている情報の複雑さによって決まります。
7. スタンバイサーバーにすべてのトランザクションログがインポートされると，データベースが使用可能になり，スタンバイサーバーは残りの NNMi プロセスを開始する。
8. 元のスタンバイサーバーがアクティブになり，動作 1.がやり直しされる。

(1) アプリケーションフェイルオーバー環境でのネットワークトラフィック

アプリケーションフェイルオーバー環境では，NNMi はアクティブサーバーからスタンバイサーバーにネットワークを介して次の項目を転送します。

- データベースアクティビティ（1 つの ZIP ファイルでのデータベースバックアップ）
- トランザクションログ
- それぞれのアプリケーションフェイルオーバーノードが，他方のノードが動作していることを確認するための定期的なハートビート
- ファイルがアクティブサーバーのものと同期していることをスタンバイサーバーが確認できるようにするファイル比較リスト
- パラメーターの変更（フェイルオーバーやそのほかの有効／無効），およびクラスタでのノードの追加や除外などのイベント

データベースアクティビティとトランザクションログで，アプリケーションフェイルオーバーで使用されるネットワークトラフィックのほとんどが生成されます。ここでは，この 2 つの項目について説明します。

データベースアクティビティ

NNMi はすべてのデータベースアクティビティのトランザクションログを生成します。

データベースアクティビティには，NNMi のすべてが含まれます。アクティビティには，次のデータベースアクティビティが含まれますが，そのほかにも含まれるものがあります。

- 新しいノードの検出
- ノード，インタフェース，VLAN，そのほかの管理対象オブジェクトに関する属性の検出
- 状態ポーリングとステータス変更
- インシデント，イベント，根本原因分析
- NNMi コンソールでのオペレータのアクション

データベースアクティビティを制御することはできません。例えば，ネットワークが停止すると，NNMi は多くのインシデントとイベントを生成します。このインシデントとイベントで，ネットワーク上のデバイスの状態ポーリングが開始され，NNMi でデバイスのステータスが更新されます。停止が復旧されると，ノード開始インシデントによってステータスがさらに変化します。このすべてのアクティビティによって，NNMi データベースのエントリが更新されます。

NNMi データベース自体はデータベースアクティビティによって拡大しますが，時間の経過とともに拡大は穏やかになり，環境でのサイズは安定します。

データベーストランザクションログ

NNMi データベースは，空の 16MB のファイルを作成してからデータベーストランザクション情報をそのファイルに書き込むことで動作します。NNMi は，15 分が経過した時点か，16MB のデータがファイルに書き込まれた時点のどれかの早い時点でこのファイルを閉じて，アプリケーションフェイルオーバーで使用できるようにします。つまり，完全にアイドル状態のデータベースで，15 分ごとに 1 つのトランザクションログファイルが生成されますが，このファイルは本質的に空です。アプリケーションフェイルオーバーでは，すべてのトランザクションログが圧縮され，空の 16MB のファイルは 1MB 未満に圧縮されます。満杯の 16MB のファイルは約 8MB に圧縮されます。データベースアクティビティが多い期間は，それぞれのファイルがすぐに満杯になるため，アプリケーションフェイルオーバーによって短時間により多くのトランザクションログが生成されます。

(2) アプリケーションフェイルオーバーのトラフィックテスト

次のテストモードでは，1 分ごとにおよそ 2 個のトランザクションログファイルが生成され，1 つのファイルの平均ファイルサイズは 7MB になります。これは，それぞれのフェイルオーバーイベントで追加される 5,000 個のノードの検出に関連するデータベースアクティビティによるものです。このテストケースのデータベースは，最終的に約 1.1GB で安定し（バックアップの ZIP ファイルのサイズで測定），ノードは 31,000 個，インタフェースは 960,000 個になります。

テストモード

最初の 4 時間でテスト担当者が 5,000 個のノードを NNMi にシードして，検出が安定するまで待機しました。4 時間後，テスト担当者がフェイルオーバーを誘発し，スタンバイサーバーがアクティブになり，以前のアクティブサーバーがスタンバイになりました。テスト担当者はフェイルオーバー直後に約 5,000 個のノードをさらに追加し，また 4 時間待機して NNMi の検出プロセスを安定させてから，別のフェイルオーバーを誘発し，以前のアクティブサーバーに戻りました。

テスト担当者は，フェイルオーバー間の時間を，4 時間，6 時間，2 時間というよう変更して，このサイクルを数回繰り返しました。テスト担当者は，それぞれのフェイルオーバーイベント後に，次の項目を測定します。

- ノードが初めてアクティブになったときに作成されるデータベース
- バックアップの ZIP ファイルのサイズ
- トランザクションログ
- ファイル総数，およびディスク容量の使用量
- フェイルオーバーを誘発する直前の NNMi データベースのノードとインタフェースの数

- フェイルオーバーが完了するまでの時間

  アクティブサーバーでovstop コマンドを最初に実行してから，スタンバイサーバーが完全にアクティブになって NNMi が動作するまでの時間。

結果

結果は次の表のとおりです。

表 18-2　アプリケーションフェイルオーバーのテスト結果

時間	DB.zip サイズ（単位：MB）	トランザクションのログの数	トランザクションログのサイズ（単位：GB）	ノード数	インタフェース数	フェイルオーバーの時間（単位：分）
4	6.5	50	0.3	5,000	15,000	5
8	34	500	2.5	12,000	222,000	10
12	243	500	2.5	17,000	370,000	25
16	400	500	3.5	21,500	477,000	23
20	498	500	3.5	25,500	588,000	32
26	618	1,100	7.5	30,600	776,000	30
28	840	400	2.2	30,600	791,000	31
30	887	500	2.5	30,700	800,000	16

所見

NNMi がアクティブサーバーからスタンバイサーバーにファイルを転送する場合，転送は 4 時間ごとに平均で約 5GB，連続スループットは約 350KB／秒，または 2.8MB／秒になっています。

メモ

- このデータには，ハートビート，ファイル整合性チェック，そのほかのアプリケーションフェイルオーバー通信など，アプリケーションフェイルオーバートラフィックは含まれていません。また，パケットヘッダーなどのネットワーク I／O のオーバーヘッドも除外されています。このデータには，ネットワークで移動する各ファイルの内容の実ネットワークペイロードだけが含まれます。
- NNMi のアプリケーションフェイルオーバー環境で生成されるトラフィックは非常に膨大です。アプリケーションフェイルオーバーでは，5 分ごとにアクティブサーバーで新しいトランザクションログが識別され，スタンバイサーバーに送信されます。ネットワークの速度によって，スタンバイサーバーではすべての新しいファイルが短時間で受信され，この 5 分間隔の残りの間，ネットワークはアイドル状態となることが多くなります。

アクティブサーバーとスタンバイサーバーがロールを切り替えるたびに，すなわち，スタンバイサーバーがアクティブになり，アクティブサーバーがスタンバイになるたびに，新しいアクティブサーバーは完全なデータベースバックアップを生成し，ネットワークを介して新しいスタンバイサーバーに送信します。このデータベースバックアップも定期的に発生し，デフォルトで 24 時間ごとにバックアップされます。NNMi は，新しいバックアップを生成するたびに，このバックアップをスタンバイサーバー

に送信します。この新しいバックアップがスタンバイサーバーで使用可能になると，その 24 時間に NNMi が生成したすべてのトランザクションログがデータベースに反映されて，フェイルオーバー時にインポートする必要がなくなるため，フェイルオーバー時間が短縮されます。

このことによって，NNMi データベースを使用してアプリケーションフェイルオーバーで NNMi を使用するとき，フェイルオーバー後にネットワークがどのようなパフォーマンスになるかを理解できます。

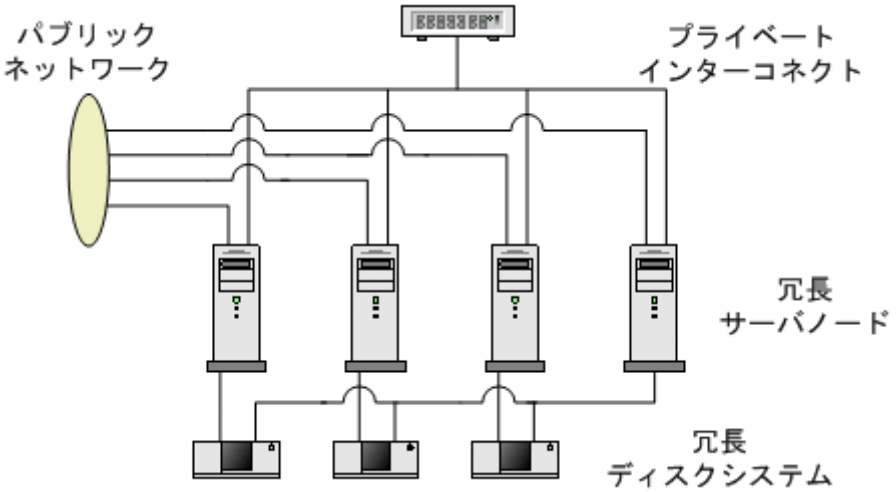
19 高可用性クラスタに NNMi を設定する

高可用性（HA）とは，構成された動作中のハードウェアおよびソフトウェアの一部に障害が発生しても中断されないサービスを提供するシステムです。HA クラスタは，フェイルオーバー発生時の機能とデータの継続性を保証するために，協調して動作するハードウェアとソフトウェアのグループ化を定義します。この章では，HA 環境で実行する NNMi を設定するためのテンプレートについて説明します。この章では，HA 製品の詳細な設定手順については説明しません。NNMi に用意されている HA 設定コマンドは，サポートされる HA 製品用のコマンドに関するものです。NNMi HA コマンドを使用して，NNMi 用に HA を適切に設定します。

19.1　HA の概念

クラスタアーキテクチャには，クラスタ内の複数のノードのプロセスとリソース用の単一のグローバルに徹底した管理ビューが備わっています。次の図に，クラスタアーキテクチャの例を示します。

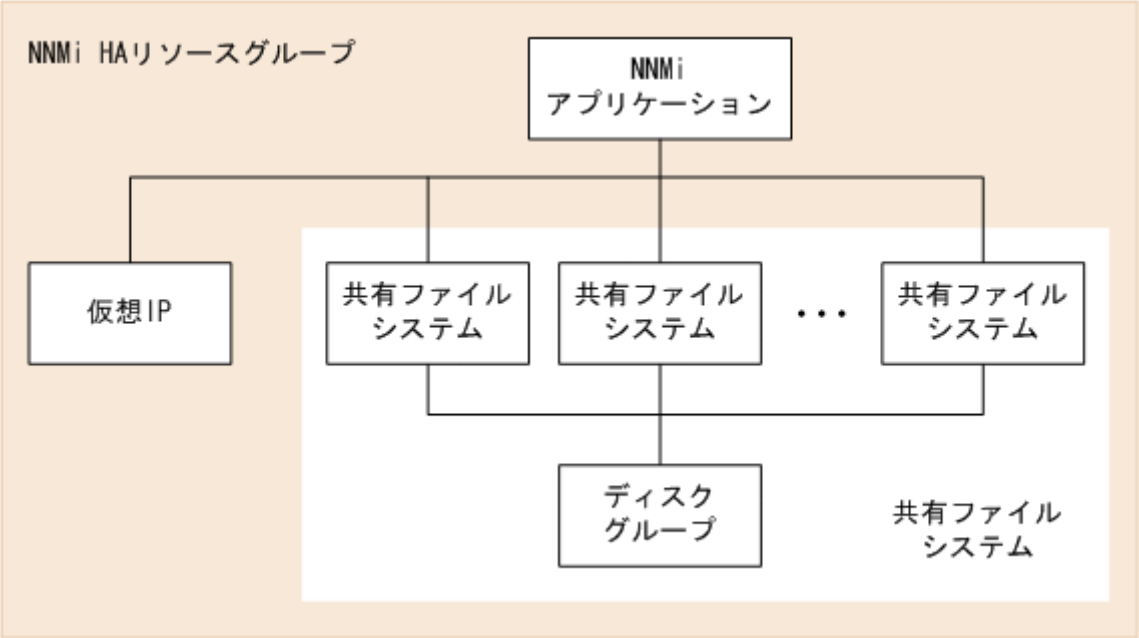
図 19-1　高可用性クラスタのアーキテクチャ



クラスタ内の各ノードは，1 つ以上のパブリックネットワークと 1 つのプライベートインターコネクト（クラスタノード間のデータ伝送用の通信チャネル）に接続されます。

Veritas Cluster Server，Symantec Cluster Server，Windows Server Failover Cluster などの最新のクラスタ環境では，アプリケーションはリソースの複合体として表現され，単純な操作でアプリケーションをクラスタ環境で実行できます。リソースは，クラスタ環境で動作するアプリケーションを表す，**HA リソースグループ**に構成されます。次の図に，HA リソースグループの例を示します。

図 19-2　典型的な HA リソースグループのレイアウト



このマニュアルでは，各種のクラスタ環境内のリソースの集合を指すために，HA リソースグループという用語を使います。各 HA 製品では，HA リソースグループに対して，異なる名前が使われています。次の表に，このマニュアルの HA リソースグループに相当する，サポート対象の HA 製品で使われている用語を示します。

表 19-1　サポート対象の HA 製品で HA リソースグループに相当する名前

HA 製品	略語	HA リソースグループに相当する名前
Veritas Cluster Server	VCS	サービスグループ
Symantec Cluster Server	SCS	サービスグループ
Windows Server Failover Cluster	WSFC※	リソースグループ
HA モニタ	HA モニタ	サーバー
CLUSTERPRO X	CLUSTERPRO	フェイルオーバグループ

注※

WSFC は，MSFC（Microsoft Failover Cluster）と表記する場合もありますが，このマニュアルでは WSFC と表記します。

表 19-1 の HA 製品は，すべての OS で使用できるわけではありません。

対応するクラスタソフト，そのバージョンについての詳細は，弊社ホームページからご確認ください。

19.1.1　HA 用語集

次の表に，一般的な HA 用語の定義を示します。

表 19-2　一般的な HA 用語

用語	説明
HA リソースグループ	クラスタ環境内（HA 製品下）で動作する各種リソースの集合です。
ボリュームグループ	大規模ストレージエリアを形成するよう設定された 1 つ以上のディスクドライブです。
論理ボリューム	ボリュームグループ内で，個別のファイルシステムまたはデバイススワップ空間として使われる任意のサイズの領域です。
プライマリクラスタノード	ソフトウェア製品が最初にインストールされるシステムであり，かつ，HA が最初に設定されるシステムです。初期セットアップでは，共有ディスクはプライマリクラスタノードにマウントされます。プライマリクラスタノードは，通常，最初のアクティブなクラスタノードになりますが，HA の設定完了後には，プライマリとしての役割を解除できます。HA 設定を変更すると，ほかのノードをプライマリクラスタノードにできます。
セカンダリクラスタノード	プライマリクラスタノードでの HA 設定の完了後に，HA 設定に追加される任意のシステムです。
アクティブなクラスタノード	現在 HA リソースグループを実行中のシステムです。

用語	説明
パッシブなクラスタノード	HA 用に設定されているが，現在 HA リソースグループを実行していないシステムです。アクティブなクラスタノードで障害が発生すると，HA リソースグループはパッシブなクラスタノードにフェイルオーバーし，そのノードがアクティブなクラスタノードになります。

19.1.2 NNMi HA クラスタのシナリオ

NNMi HA 設定では，NNMi は各システムにインストールされ，HA リソースグループの一部になります。NNMi データベースは独立したディスクにインストールされ，各システムで動作中の NNMi プログラムからアクセスされます（任意の時点で共有ディスクにアクセスできるのは，アクティブなクラスタノードである 1 つのシステムだけです）。

メモ

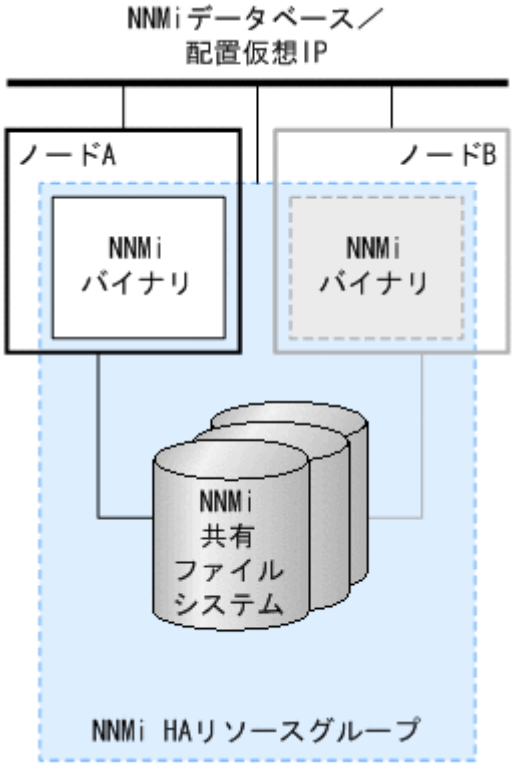
NNMi データベースのバックアップスクリプトとリストアスクリプトを実行できるのは，アクティブなクラスタノードだけです。

NNMi だけのシナリオ

次の図に，NNMi HA クラスタのシナリオを示します。

ノード A とノード B は，どちらも，すべてのソフトウェアがインストールされた NNMi 管理サーバーであり，そのシステムで実行する NNMi プログラムが含まれています。アクティブなクラスタノードが，共有ディスクのランタイムデータにアクセスします。ほかの製品は，HA リソースグループの仮想 IP アドレスを使って，NNMi に接続します。

図 19-3　NNMi HA クラスタ用の基本的なシナリオ



このシナリオの実装方法については，「19.4.2　HA 用に NNMi を設定する」を参照してください。

19.1.3　man ページ

NNMi には，NNMi 高可用性設定に役立つ次の man ページがあります。

- nnm-ha
- nnmhaconfigure.ovpl
- nnmhaunconfigure.ovpl
- nnmhadisk.ovpl
- nnmhaclusterinfo.ovpl
- nnmhastartrg.ovpl
- nnmhastoprg.ovpl

Windows オペレーティングシステムでは，これらの man ページはテキストファイルで提供されます。

19.2 HA 用 NNMi を設定するための前提条件の検証

NNMi を動作させる HA クラスタは，次の要件を満たしている必要があります。

システム構成全般について

- 複数の HA 製品をインストールした構成では NNMi は使用できません。NNMi が HA 製品の種類を正しく認識できず正常に動作しない場合があります。
  - Windows：すべてのクラスタノードで，NNMi のインストール先（%NnmDataDir% と%NnmInstallDir%）を一致させてください。
- 両方の NNMi 管理サーバーは同じバージョン（修正版のバージョンを含む）の NNMi を実行している必要があります。例えば，アクティブサーバーで NNMi 11-00-01 を実行している場合，スタンバイサーバーでも同一の NNMi 11-00-01 がインストールされている必要があります。
- Windows（WSFC）の場合：NNMi は，クラスタシステムで`cluster.exe` コマンドを使用しますが，デフォルトではインストールされません。

  ［サーバー マネージャー］ ＞ **［役割と機能の追加］** より，**［機能］** ＞ **［リモートサーバー管理ツール］** ＞ **［機能管理ツール］** ＞ **［フェールオーバー クラスタリング ツール］** ＞ **［フェールオーバー クラスター コマンド インターフェイス］** をインストールしてください。
- 必要なディスク容量は，リリースノートの「4. メモリ所要量およびディスク占有量」の表に記載されている次の項目を参照してください。
  - ローカルディスクのディスク容量

    「アプリケーションインストール用のディスク容量」の値と「実行時のデータベースとデータ用のディスク容量」の値
  - 共有ディスクのディスク容量

    「実行時のデータベースとデータ用のディスク容量」の値

リソースグループについて

- NNMi は，設定するリソースグループがない状態からセットアップする必要があります。NNMi を既存のリソースグループに追加することはできません。
- 仮想 IP アドレスおよび共有ディスクの使用がサポートされ，NNMi から使用できる構成にしてください。

共有ディスクについて

- NNMi の共有データは次の場所に格納されます。ディレクトリ名に空白を含めることはできません。ディレクトリ名「NNM」は固定です。
  - Windows：<ドライブ文字>:¥NNM （例 Y:¥NNM）または<ドライブ文字>:¥<任意のディレクトリ>¥NNM （例 Y:¥JP1¥NNM）
  - Linux：<マウントポイント>/NNM （例 /shdsk1/NNM）
- 共有ディスクは，Fibre（FC-SAN），SCSI，iSCSI で接続されたストレージを使用してください。NFS 接続や CIFS 接続の NAS などを使う構成は NNMi ではサポートしていません。

- Windows：共有ディスクには，ドライブ文字を割り当てたディスクを使用してください。 [ディスクの管理]でのマウント設定や`mountvol`コマンドによってマウントしたディスクは使用しないでください。マニュアル上にマウントと書かれている個所は Linux を対象とした説明です。
- Windows：Microsoft Cluster Service を使用している Windows Server のクラスタリングでは，ダイナミックディスクはサポートされていません。
- NNMi の前提とする OS やクラスタソフトが，NNMi が共有ディスクと同様にアクセスできるようにディスク（ファイル）領域を提供するのであれば，そのようなディスク領域の使用もサポートします。

  具体的には，次が条件となります。

  - NNMi から，ディスクの「マウントするディレクトリ」にアクセスできる。
  - アクティブ側の NNMi からは上記ディレクトリに制限なくアクセスでき，スタンバイ側の NNMi からはアクセスできない。
  - フェイルオーバーの前後で，ディスクの「マウントするディレクトリ」内のデータに差異がない。

仮想 IP アドレスについて

- 仮想 IP アドレスと仮想ホスト名は，DNS などのネームサービスまたは hosts ファイルに対して，ホスト名から IP アドレスおよび逆に IP アドレスからホスト名が変換できるように設定してください。
- DNS などのネームサービスを使う場合も，hosts ファイルに仮想 IP アドレスと仮想ホスト名が名前解決できるように設定してください。これは通信障害が発生してフェイルオーバーする場合に，名前解決ができないでフェイルオーバー処理が失敗することを防止するためです。
- IPv6 の論理 IP アドレスをリソースとして設定する場合，「19.4　HA を設定する」の手順のあとに手動で追加してください。設定手順については，クラスタソフトのマニュアルなどを参照してください。

  IPv6 が使用できるクラスタソフトのバージョン，IPv6 を使用する場合の構成，IPv6/IPv4 の混在可否などは，クラスタソフトの仕様に依存します。

仮想ホスト名について

クラスタ環境構築時，仮想ホスト名は IPv4 のアドレスで名前解決されるようにしてください。

19.3 HA 設定の注意事項

HA 設定の注意事項を次に示します。

19.3.1 関連製品を使用する場合の注意

NNMi の関連製品，例えば，JP1/SNMP System Observer（JP1/SSO）や JP1/Integrated Management 3 - Event Gateway for Network Node Manager i（JP1/IM-EG）などを使用する場合は，次のように設定してください。

- 最初に NNMi をセットアップし，その後に関連製品をセットアップしてください。なお，HA 構成を解除する場合は，セットアップ時と逆順にしてください。（NNMi を最後に解除してください。）
- NNMi と，関連製品は，同一のリソースグループに登録します。
  このとき，クラスタソフトに設定するリソースの依存関係は次のとおりです。
  - 関連製品は，NNMi を前提とする依存関係を設定します。
  - NNMi は，共有ディスクおよび仮想 IP アドレスを前提とする依存関係を設定します。Windows の場合，追加でネットワーク名リソースについても依存関係を設定します。
  - JP1/Integrated Management 2（JP1/IM2）または JP1/Integrated Management 3（JP1/IM3）のカスタム UI 連携を使用する場合，JP1/IM2 または JP1/IM3 と NNMi は同一のリソースグループに登録する必要があります。ただし，NNMi との依存関係の設定は任意です。

関連製品の設定方法は，それぞれのマニュアル，リリースノート，取扱説明書を参照してください。

19.3.2 設定作業や運用操作の注意

NNMi の HA 構成を設定や操作する場合は，次の状態で操作してください。

- 操作する OS ユーザーには，クラスタソフトの全操作が可能な権限を付与してください。クラスタソフトに対して NNMi のリソース作成やリソースグループの起動停止などの操作を行うため，これらの操作権限が必要です。
- クラスタソフトが動作している状態で操作をしてください。NNMi の HA 構成用の各種コマンドは，クラスタソフトに対し，設定や構成確認などの処理を行います。クラスタソフトが停止している場合はエラーが発生します。
- このマニュアルおよびリリースノートに記載されている手順によって NNMi サービスを再起動する場合，特に断りのないかぎり，HA クラスタ環境ではメンテナンスモードに設定してから実行してください。
- ドキュメントなどで特に断りのないかぎり，コマンド実行やローカルファイルの編集は NNMi のリソースグループがオンラインの状態で実施してください。

また，実施後 3 分以内にフェイルオーバーしないようにしてください。

リソースグループがオフラインの状態でコマンド実行やローカルファイルの編集を実施した場合や，実施後 3 分以内にフェイルオーバーした場合は，古い設定で上書きされるおそれがあります。

- NNMi リソースは，障害が発生した場合にフェイルオーバーすることを想定しています。

  そのため，リソースグループで障害が発生した場合は，障害が発生した系で再起動しないで，フェイルオーバーするように設定してください。

  設定方法についてはクラスタソフトのヘルプなどを参照してください。

- 障害によってフェイルオーバーが発生した環境で，フェイルバックでフェイルオーバー元に戻す場合は，発生した障害を回復し，残存する NNMi のプロセスをすべて停止したあとにフェイルバックしてください。

- HA クラスタ環境で NNMi を使用する場合，プロセスダウンやディスク障害に伴い，ごくまれに NNMi のデータベースが破損することがあります。このため，定期的にバックアップを取得してください。

19.3.3　そのほかの注意

- 環境によっては，NNMi サービスの起動に 10 分以上掛かる場合があります。
- （Windows（WSFC）の場合）フェイルオーバークラスタ管理コンソールに表示される `<resource_group>`-APP の状態には，「オンライン待ち」，「オフライン待ち」が表示されません。`<resource_group>-APP` が待ち状態であるかどうかは，フェイルオーバークラスタ管理コンソールの次の状態が「保留中」となっていることを確認してください。
  - ［<クラスタ名>］ ＞ ［サービスとアプリケーション］ ＞ ［<resource_group>］ の ［<resource_group>の概要］ の状態
- （Windows（WSFC）の場合）資料採取ツールを実行したときに`cluster.exe log /g` を実行して `cluster.log` を作成してください。

19.4　HA を設定する

ここでは，NNMi 用の新規 HA 設定の設定手順を説明します。

19.4.1　HA 用の NNMi 証明書を設定する

NNMi のインストールプロセスでは，NNMi コンソールと NNMi データベースの間でセキュア通信が行われるよう，自己署名証明書を設定します。NNMi HA を正しく設定するプロセスでは，プライマリクラスタノードとセカンダリクラスタノードの間で自己署名証明書を共有します。HA 下で実行される NNMi でデフォルトの証明書を使用するために，追加の手順を実行する必要はありません。

NNMi の通信で別の自己署名証明書，または認証機関（CA）署名の証明書を使用する場合は，追加の手順を実行する必要があります。新しい証明書を入手してから，「10.3.6　高可用性環境での証明書の使用」に従って手順を実行します。この手順は，HA 用 NNMi を設定する前，または後に実行できます。

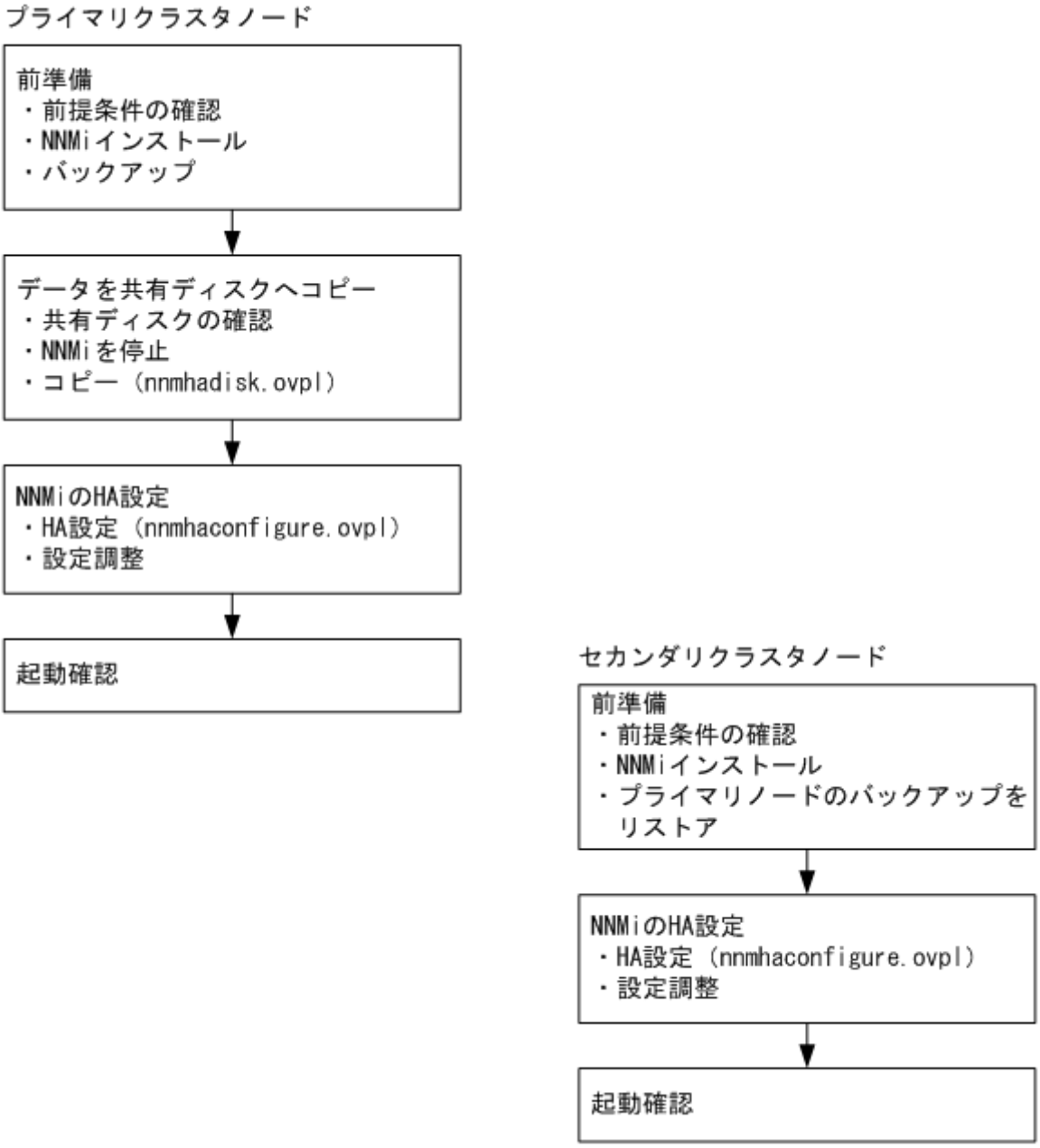
19.4.2　HA 用に NNMi を設定する

ここでは，HA 用に NNMi を設定する作業の流れ，および検討段階で決めておく設定情報について説明します。

HA 用に NNMi を設定する場合の主な作業は，次の 2 つです。

1. NNMi データファイルを共有ディスクにコピーする。
   プライマリクラスタノードでこの作業を行います。
2. HA 下で NNMi を実行するように，設定する。
   - プライマリクラスタノードでこの作業を行います。
   - セカンダリクラスタノードでこの作業を行います。

設定作業の流れを次に示します。



1 つの HA クラスタノードを，プライマリ NNMi 管理サーバーとして割り当てます。これが大部分の時間にアクティブとなるノードです。プライマリクラスタノードとして設定します。次に HA クラスタ内の残りのノードをセカンダリクラスタノードとして設定します。

重要

HTTPS 通信を使用して NNMi サーバーにアクセスする場合，プライマリクラスタノードの起動確認の前に，証明書を使用するように設定します。詳細については，「10.3.6　高可用性環境での証明書の使用」を参照してください。

メモ

- HA 用の NNMi の設定は，複数のクラスタノードで同時には行えません。プライマリクラスタノードで HA 設定プロセスが完了したあと，セカンダリクラスタノードでの HA 設定プロセスを開始する，というように，クラスタ環境内のノードを 1 つずつ HA 用に設定してください。
- HA モニタおよび CLUSTERPRO の場合は，nnmhaconfigure.ovpl を使わないで設定作業を行います。設定方法については，リリースノートを参照してください。

フェイルオーバー中には NNMi コンソールは応答しません。フェイルオーバーが完了してから，NNMi ユーザーは，サインインして NNMi コンソールのセッションを続行してください。

NNMi の`TrapReceiver` プロセス，およびそのフェイルオーバーとの関連に関する重要情報については，「21.16　NNMi NmsTrapReceiver プロセス」を参照してください。

(1)　NNMi HA 設定情報

HA 設定スクリプト（`nnmhaconfigure.ovpl`）は，NNMi HA リソースグループに関する情報を収集します。次の表に，プライマリクラスタノードの設定で必要になる情報を示します。設定作業を開始する前に，これらの情報を用意してください。

これらの情報は，設定作業時に HA 設定スクリプト（`nnmhaconfigure.ovpl`）を実行して対話形式で入力します。入力要求が OS や HA 製品の種類およびシステム構成に合わせて表示されますので，画面表示に従って入力してください。

表 19-3　NNMi HA プライマリクラスタノードの設定情報

HA 設定項目	説明
HA リソースグループ	NNMi を含む HA クラスタのリソースグループの名前です。この名前は NNMi に対して一意であり，現在使用されていない名前にする必要があります。 （例）：nnmtest1 **注記**：HA リソースグループの名前に，空白を含む文字列は使用できません。 **注記**：名前に使用できる文字種，文字数はクラスタソフトによって異なります。詳しくは，表の欄外の説明を参照してください。 **注記**：HA リソースグループの名前は，ほかのリソース名やリソースグループ名の部分文字列（相手の文字列の一部または全部に一致する文字列）にならないようにしてください。例えば，リソースグループ testA が存在する場合，test や est などの名称は使用できません。 **注記**：HA リソースグループ作成後に名前を変更することはできません。名前を変更するには，NNMi の HA 設定を解除し，新しい名称で HA 設定をし直してください。
仮想ホストの名前	仮想ホストの名前です。ドメイン名を含む FQDN 名ではなく短い名前を指定します。このホスト名は，HA リソースグループの仮想 IP アドレスにマッピングする必要があります。仮想ホストの短い名前と仮想 IP アドレスを名前解決できる必要があります。

HA 設定項目	説明
仮想ホストの名前	注記：HA 設定の完了後に仮想ホスト名を変更することはできません。仮想ホスト名を変更するには，NNMi の HA 設定を解除し，新しい名前で HA 設定をし直してください。 注記：NNMi が仮想ホストの短い名前と仮想 IP アドレスを解決できない場合は，HA 設定スクリプトによって，システムが不安定な状態になる可能性があります。したがって，NNMi HA の設定中に DNS が利用できない場合に備えて，予備の手段（例えば，Windows オペレーティングシステムの場合は，%SystemRoot%¥system32¥drivers¥etc¥hosts ファイルに，Linux オペレーティングシステムの場合は，/etc/hosts ファイルに，それぞれ情報を記述する）を用意しておくことをお勧めします。
仮想ホストのネットマスク	仮想ホスト IP アドレスで使われるサブネットマスクです。これは，IPv4 アドレスであることが必要です。
仮想ホストのネットワークインタフェース	仮想ホスト IP アドレスが使われるネットワークインタフェースです。 (例) • Windows：ローカルエリア接続 • Linux：eth0
共有ファイルシステムのタイプ	HA リソースグループで使われる共有ディスクの設定タイプです。次のどちらかになります。 • disk：共有ディスクは，標準のファイルシステムタイプを使う，物理的に接続されたディスクです。HA 設定スクリプトは，共有ディスクを設定できます。詳細については，この表のファイルシステムタイプの欄を参照してください。 • none：共有ディスクには，disk オプションで説明している設定以外の SAN や NFS 構成などを使います。HA 設定スクリプトを実行すると，共有ディスクが設定されます。 注記：JP1/NNMi では none を指定した場合の動作はサポートしていません。必ず disk を指定してください。
ファイルシステムタイプ	(Linux だけ) 共有ディスクのファイルシステムタイプです（共有ファイルシステムのタイプがdisk の場合)。HA 設定スクリプトは，ディスクの検証方法を調べるために，この値を HA 製品に渡します。 次の共有ディスクフォーマットはテスト済みです。 • VCS または SCS には ext2，ext3，および vxfs
ディスクグループ	(Linux だけ) NNMi 共有ファイルシステムのディスクグループの名前です。 (例)：shdg01
ボリュームグループ	(Linux だけ) NNMi 共有ファイルシステムのボリュームグループの名前です。 例：vg03
マウントするディレクトリ (マウントポイント)	NNMi の共有ディスクをマウントするディレクトリの場所です。このマウントポイントは，すべてのシステムで同じである必要があります（つまり，各ノードでは，マウントポイントに同じ名前を使う必要があります)。Windows の場

HA 設定項目	説明
マウントするディレクトリ (マウントポイント)	合，<ドライブ文字> または<ドライブ文字>:¥<任意のディレクトリ> を指定します。ディレクトリ名に空白を含めることはできません。 (例) • Windows： Y:またはY:¥JP1 • Linux： /nnmmount **注記**：NNMi の共有データは，上で指定したディレクトリ直下に作成される NNM という格納先ディレクトリ内に保存されます（格納先ディレクトリのパスを次に示します)。格納先ディレクトリ名（NNM）は固定です。 • Windows： *<ドライブ文字>*:¥NNM または <ドライブ文字>:¥<任意のディレクトリ>¥NNM • Linux： *<マウントポイント>*/NNM **注記**：HA の設定完了後にマウントポイントを変更することはできません。Windows の場合は，HA の設定完了後にマウントポイントのドライブ文字も変更することはできません。変更が必要な場合は，「19.7 HA クラスタ内の NNMi の設定を解除する」の手順で HA の設定を解除したあと，「19.4 HA を設定する」の手順で再度 HA を設定してください。

NNMi の HA リソースグループの名前に使える文字の種類および文字数は，クラスタソフトによって異なります。NNMi 用の HA リソースグループでは，次の範囲で名称を指定してください。

- Windows WSFC の場合
  - 文字種：
    英字（a-z，A-Z)，数字（0-9)，ハイフン（-)，アンダーバー（_)，ピリオド（.)
  - 文字数：%NnmDataDir%hacluster¥*<resource_group>*のパス名を含む文字列全体で 247 文字まで
- Linux VCS または Linux SCS の場合
  - 文字種：
    英字（a-z，A-Z)，数字（0-9)，ハイフン（-)，アンダーバー（_)
    ただし先頭は英字
  - 文字数：255 文字まで
- Linux HA モニタの場合
  - 文字種：
    英字（a-z，A-Z)，数字（0-9)
    ただし先頭は英字
  - 文字数：8 文字まで
- Windows CLUSTERPRO または Linux CLUSTERPRO の場合
  - 文字種：
    英字（a-z，A-Z)，数字（0-9)，ハイフン（-)，アンダーバー（_)
    ただし先頭は英字

- 文字数：16 文字まで

19.4.3　HA 用に NNMi を設定する（Windows の場合）

ここでは，Windows 環境で HA 用に NNMi を設定する手順を説明します。

NNMi の HA 設定では，新規に NNMi 用のリソースグループを作成します。このため，対象のリソースグループがない状態から設定作業を行ってください。

NNMi の HA 設定を行うスクリプト（`nnmhaconfigure.ovpl`）は，内部的にクラスタソフトに対してリソースグループや各リソースを作成する処理を行います。設定作業が完了すると，次のリソースグループが設定されます。

表 19-4　WSFC での NNMi 用リソースグループの構成

リソースの名前	リソースの種類	説明
仮想ホスト名	ネットワーク名	仮想ホスト名を制御する
<resource_group>-IP	IP アドレス	仮想 IP アドレスを制御する
<resource_group>-mount	物理ディスク	共有ディスクを制御する
<resource_group>-APP	汎用スクリプト	NNMi の起動/停止/監視を制御する

WSFC の場合，`nnmhaconfigure.ovpl` が`cluster.exe` などのコマンドを内部的に実行して上記のリソースの設定処理を行います。

- *<resource_group>*の部分は HA リソースグループ名に置き換わります。
- リソースの依存関係は，NNMi 用の汎用スクリプトリソース*<resource_group>*-APP の前提に，*<IP アドレスリソース>*，*<ディスクリソース>*，*<ネットワーク名リソース>*を設定します。

重要

CLUSTERPRO の場合は，`nnmhaconfigure.ovpl` を使わずに設定作業を行います。設定方法については，リリースノートを参照してください。

(1)　WSFC の各リソースの設定内容の例

設定が完了したときの WSFC の各リソースの設定内容の例を次に示します。なお，*<resource_group>*の部分は HA リソースグループ名に置き換わります。

表 19-5　<ネットワーク名リソース>

項目	詳細
[全般]	• リソース名：(仮想ホスト名)

項目	詳細
[全般]	• リソースの種類：ネットワーク名 • DNS 名：(仮想ホスト名) • フルネーム：(仮想ホスト名).test.com • ネットワーク：192.168.100.0/24 • IP アドレス：192.168.100.24 • NetBIOS 状態：OK • DNS 状態：OK • kerberos 状態：OK
[依存関係]	<IP アドレスリソース>
[ポリシー]	• [リソースが失敗状態になった場合は，現在のノードで再起動を試みる] を有効 再起動間隔：15:00 指定期間内での再起動の試行回数：0 • [再起動に失敗した場合は，このサービスまたはアプリケーションのすべてのリソースをフェールオーバーする] を有効 • 保留タイムアウト：03:00
[詳細なポリシー]	• 基本的なリソース正常性チェックの間隔 [リソースの種類の標準間隔を使用する] • 完全なリソース正常性チェックの間隔 [リソースの種類の標準間隔を使用する] • [このリソースを別のリソース モニタで実行する] を無効

表 19-6　<IP アドレスリソース>

項目	詳細
[全般]	• リソース名：*<resource_group>*-IP • リソースの種類：IP アドレス • ネットワーク：192.168.100.0/24 • 静的 IP アドレス：192.168.100.24 ※ • [このアドレスの NetBIOS を有効にする] を有効
[依存関係]	依存関係なし
[ポリシー]	• [リソースが失敗状態になった場合は，現在のノードで再起動を試みる] を有効 再起動間隔：15:00 指定期間内での再起動の試行回数：0 • [再起動に失敗した場合は，このサービスまたはアプリケーションのすべてのリソースをフェールオーバーする] を有効 • 保留タイムアウト：03:00
[詳細なポリシー]	• 基本的なリソース正常性チェックの間隔 [リソースの種類の標準間隔を使用する] • 完全なリソース正常性チェックの間隔 [リソースの種類の標準間隔を使用する] • [このリソースを別のリソース モニタで実行する] を無効

注※　DHCP は有効にしません。

表 19-7　<物理ディスクリソース>

項目	詳細
[全般]	• リソース名：<*resource_group*>-mount • リソースの種類：物理ディスク • ボリューム：Y:
[依存関係]	依存関係なし
[ポリシー]	• [リソースが失敗状態になった場合は，現在のノードで再起動を試みる] を有効 再起動間隔：15:00 指定期間内での再起動の試行回数：0 • [再起動に失敗した場合は，このサービスまたはアプリケーションのすべてのリソースをフェールオーバーする] を有効 • 保留タイムアウト：03:00
[詳細なポリシー]	• 基本的なリソース正常性チェックの間隔 [リソースの種類の標準間隔を使用する] • 完全なリソース正常性チェックの間隔 [リソースの種類の標準間隔を使用する] • [このリソースを別のリソース モニタで実行する] を無効

表 19-8　<汎用スクリプトリソース>

項目	詳細
[全般]	• リソース名：<*resource_group*>-APP • リソースの種類：汎用スクリプト • スクリプトのパス※： %NnmDataDir%hacluster/<resource_group>/hamscs.vbs
[依存関係]	<ネットワーク名リソース>，<IP アドレスリソース>および <ディスクリソース>
[ポリシー]	• [リソースが失敗状態になった場合は，現在のノードで再起動を試みる] を有効 再起動間隔：15:00 指定期間内での再起動の試行回数：0 • [再起動に失敗した場合は，このサービスまたはアプリケーションのすべてのリソースをフェールオーバーする] を有効 • 保留タイムアウト：30:00
[詳細なポリシー]	• 基本的なリソース正常性チェックの間隔 [リソースの種類の標準間隔を使用する] • 完全なリソース正常性チェックの間隔 [リソースの種類の標準間隔を使用する] [このリソースを別のリソース モニタで実行する] を有効

注※

スクリプトのパスは，環境変数を展開したフルパスが設定されます。

(例)

C:/ProgramData/Hitachi/Cm2NNMi/hacluster/jp1ha1/hamscs.vbs

(2) プライマリクラスタノードでの NNMi の設定

プライマリクラスタノードで次の手順を実行します。

(a) 前準備

1. 「19.2 HA 用 NNMi を設定するための前提条件の検証」の作業が完了していることを確認する。
2. NNMi がインストールされていない場合は，インストールする。そして，NNMi が正しく動作することを確認する。
3. 次のコマンドを使って，NNMi 設定をバックアップする。

   （例）

   ```
   nnmbackup.ovpl -scope all -target nnmi_backups
   ```

   このコマンドの詳細については，「20. NNMi のバックアップおよびリストアツール」を参照してください。

   NNMi のクラスタ環境構成において初期状態では，プライマリクラスタノードのデータと，セカンダリクラスタノードのデータが完全に一致している必要があります。このため，ここで取得したバックアップデータを，セカンダリクラスタノードの設定手順でリストアし，データを一致させます。

(b) データの共有ディスクへのコピー

1. NNMi HA リソースグループ用に，共有ディスクを用意する。

   Windows Explorer とディスクの管理ツールを使用してドライブ名を割り当てます。

   重要

   用意した共有ディスクが，次の条件を満たすことを確認してください。

   - ディスクの管理ツールを使用して，共有ディスクで［オンライン］と表示されるようにします。［予約］と表示される場合，これは WSFC が共有ディスクを制御することを示しています。WSFC ユーザーインタフェースから［削除］アクションを使用して，共有ディスクを WSFC コントロールから削除します。また，ディスクの管理ツールを使用して，［予約］フラグが［オンライン］に変更されることも確認します。
   - フォーマット済みである
   - 十分な空き容量がある
   - ほかのリソースグループで使用されていない
   - 管理者権限のユーザーへの「フルコントロール」，およびビルトイン Local Service ユーザー（Users グループ）への「読み取りと実行」の権限がある

2. NNMi を停止する。

   ```
   %NnmInstallDir%bin¥ovstop -c
   net stop NnmTrapReceiver
   ```

3. NNMi ファイルを共有ディスクにコピーする。

```
%NnmInstallDir%misc¥nnm¥ha¥nnmhadisk.ovpl NNM -to <HA_mount_point>
```

重要

*<HA_mount_point>*には，共有ディスクのドライブまたは共有ディスクドライブ配下の任意のディレクトリを指定します（例 Y:または Y:¥JP1 など）。

ディレクトリ名に空白を含めることはできません。

指定したパス直下に，ディレクトリ「NNM」が作成されます（例 Y:¥NNM または Y:¥JP1¥NNM）。

格納先ディレクトリ名は変更できません。

(c) NNMi の HA 設定

1. NNMi HA リソースグループを新規に作成する。

```
%NnmInstallDir%misc¥nnm¥ha¥nnmhaconfigure.ovpl NNM
```

このコマンドの設定項目については，「19.9.2 NNMi に付属している HA 設定スクリプト」を参照してください。

共有ディスクタイプはnone ではなく，必ずdisk を指定してください。また，共有ディスクのパスは，(b)の手順 3.で指定したパスを指定してください。

(設定例)

HA 設定項目は，nnmhaconfigure.ovpl に対話形式で入力する項目を表示順に並べています。「19.4.2 HA 用に NNMi を設定する」の「表 19-3 NNMi HA プライマリクラスタノードの設定情報」の説明によって検討した内容を入力してください。

HA 設定項目	設定例
HA リソースグループの名前	jp1ha1
仮想ホストの名前	lhost1
仮想ホストのネットワークインタフェース	ローカルエリア接続
共有ファイルシステムのタイプ	disk（必ず disk を指定）
マウントするディレクトリ	Y ドライブ

重要

設定コマンドを実行する前に，次の注意事項を確認してください。

- 既にほかのリソースグループやリソースで使われている値をnnmhaconfigure.ovplに指定すると，リソースの作成が失敗するなどエラーが発生します。ほかで使われていないことを確認してから，nnmhaconfigure.ovplを実行してください。
- 既に使われているリソースグループ名，IPアドレスやディスクを指定した場合，リソースを作成するために実行したクラスタソフトのコマンドがエラーとなります。エラー発生時点でnnmhaconfigure.ovplは異常終了し，それまでに作成されたリソースグループやリソースは残ったままとなります。エラーを対処してnnmhaconfigure.ovplを再実行する前に，クラスタソフトの操作で残っているリソースを削除してください。
- 仮想アドレスを設定するネットワークインタフェースは次を確認してください。
- フェイルオーバークラスタ管理コンソールの［ネットワーク］で論理IPアドレスのネットワークアドレスを含むリソースを確認します。

(実行例)

設定例の値を指定した場合の画面表示例です。"？"の後ろが入力する項目です。

```
C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥misc¥nnm¥ha>nnmhaconfigure.ovpl NNM
質問: HA リソース グループの名前を入力してください:  ? jp1ha1

プライマリ ノードの設定が検出されました。

質問: 有効な仮想ホストの名前を入力してください:  ? lhost1
使用可能なネットワーク インタフェース:

ネットワーク サブネット マスク   ネットワーク インタフェース
255.255.255.0           クラスタ ネットワーク 3
255.255.255.0           クラスタ ネットワーク 1

選択可能な値:
1: クラスタ ネットワーク 3
2: クラスタ ネットワーク 1
質問: 仮想ホストのネットワーク インタフェースを入力してください:  ? 2
選択可能な値:
1: disk
2: none
質問: 共有ファイル システムのタイプを入力してください (disk, none):  ? 1
質問: ディスクをマウントするディレクトリを入力してください:  ? Y:
リソース グループを作成しています。

リソース グループ 'jp1ha1' を作成しています...

グループ              ノード          状態
-------------------- --------------- ------
jp1ha1               NNMX64-33       オフライン


リソース 'lhost1' を作成しています...

リソース              グループ             ノード          状態
-------------------- -------------------- --------------- ------
lhost1               jp1ha1               NNMX64-33       オフライン
```

```
リソース 'lhost1' をリソース 'jp1ha1-IP' に依存させています...

HA 値の C:/ProgramData/Hitachi/Cm2NNMi/shared/nnm/conf/ov.conf を設定しています。
HP OpenView Process Manager サービスの自動スタートアップを無効にしています。
[SC] ChangeServiceConfig SUCCESS
注: 指定されている仮想ホスト名に一致するようにNNMi FQDNを更新しています。fqdn を lhost1.x
xx.xxx に設定しています

ドメインを xxx.xxx に設定しています

Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

新しい SSL 証明書を生成しています。

lhost1.xxx.xxx.selfsigned のキーストアの証明書を生成しています。
[成功]

生成された証明書をトラストストアにエクスポートしています。

証明書がファイル<temporary.cert>に保存されました。
証明書がキーストアに追加されました。

C:¥Program Files (x86)¥ Hitachi¥Cm2NNMi¥misc¥nnm¥ha>
```

2. 監視プロセスに異常が発生した場合，フェイルオーバーするよう<*resource_group*>を設定する。

   <*resource_group*>-APP のプロパティを開き，［**ポリシー**］タブを押下する。

   ［リソースが失敗状態になった場合は，現在のノードで再起動を試みる］が選択されていることを確認し，［指定期間内での再起動の試行回数］を 0 に設定する。

   ［再起動の試みがすべて失敗した場合は，指定した時間（hh:mm）後にもう一度再起動を開始する(S)］にチェックがある場合は外してください。

> **重要**
>
> <*resource_group*>および<*resource_group*>に登録したリソースの設定によって，エラー発生時の動作などを指定します。各設定項目の役割についてはクラスタサービスのヘルプを参照ください。

3. プライマリクラスタノード上で，NNMTrapReceiver サービスの自動起動を無効にする。

   スタートメニューの［**管理ツール**］＞［**サービス**］で［**NNM Trap Receiver**］および［**NNM Trap Receiver Manager**］を選択し，「スタートアップの種類」を「手動」に設定してください。

4. プライマリクラスタノード上で，クラスタサービスを再起動する。

   再起動によって，これまでの設定内容が反映され，NNMi の環境変数が読み込まれます。なお net stop ClusSvc，net start ClusSvc コマンドを実行することで，サービスの起動停止ができます。

重要

HTTPS 通信を使用して NNMi サーバーにアクセスする場合，証明書を使用するように設定します。詳細については，「10.3.6　高可用性環境での証明書の使用」を参照してください。

(d)　起動の確認

1. NNMi HA リソースグループを起動する。

   起動コマンドは，プライマリクラスタノードで実行します。

   - 次の起動コマンドを実行します。

```
%NnmInstallDir%misc¥nnm¥ha¥nnmhastartrg.ovpl NNM <resource_group>
```

   - <*resource_group*>が起動したことを確認します。

     NNMi を正常に起動できなかった場合は，「19.8　HA 設定のトラブルシューティング」を参照してください。

   これで，NNMi が HA 下で動作するようになりました。

重要

HA 構成の NNMi の通常のオペレーションでは，ovstart コマンドやovstop コマンドは使わないでください。これらのコマンドは，メンテナンスを目的として操作手順に明示されている場合だけ使用します。HA 構成の NNMi の起動や停止は，クラスタソフトの操作によって HA リソースグループを起動または停止するようにしてください。

(3)　セカンダリクラスタノードでの NNMi の設定

(a)　前準備

1. 「19.2　HA 用 NNMi を設定するための前提条件の検証」の作業が完了していることを確認する。
2. NNMi がインストールされていない場合は，NNMi をインストールし，正しく動作することを確認する。
3. リストアをする。

   「(2)　プライマリクラスタノードでの NNMi の設定」の(a)の手順 3.で取得したバックアップデータをセカンダリクラスタノードにリストアします。

```
%NnmInstallDir%bin¥nnmrestore.ovpl -force -partial -source <backup_data>
```

   このコマンドの詳細については，「20.　NNMi のバックアップおよびリストアツール」を参照してください。

(b)　NNMi の HA 設定

1. NNMi を停止する。

```
%NnmInstallDir%bin¥ovstop -c
net stop NnmTrapReceiver
```

2. NNMi HA リソースグループを設定する。

```
%NnmInstallDir%misc¥nnm¥ha¥nnmhaconfigure.ovpl NNM
```

コマンドの要求に応じて，HA リソースグループ名を指定します。

(実行例)

```
C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥misc¥nnm¥ha>nnmhaconfigure.ovpl NNM
質問: HA リソース グループの名前を入力してください: ? jp1ha1
セカンダリ ノードの設定が検出されました。

HP OpenView Process Manager サービスの自動スタートアップを無効にしています。
[SC] ChangeServiceConfig SUCCESS
注: 指定されている仮想ホスト名に一致するようにNNMi FQDNを更新しています。fqdn を lhost1.x
xx.xxx に設定しています

ドメインを xxx.xxx に設定しています

Microsoft (R) Windows Script Host Version 5.7
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.

新しい SSL 証明書を生成しています。

C:¥Program Files (x86)¥ Hitachi¥Cm2NNMi¥misc¥nnm¥ha>
```

3. セカンダリクラスタノード上で，NNMTrapReceiver サービスの自動起動を無効にする。

スタートメニューの［管理ツール］＞［サービス］で［NNM Trap Receiver］および［NNM Trap Receiver Manager］を選択し，「スタートアップの種類」を「手動」に設定してください。

4. 設定が正常に行われたことを確認する。

```
%NnmInstallDir%misc¥nnm¥ha¥nnmhaclusterinfo.ovpl -group <resource_group> -nodes
```

このコマンドの出力には，指定した HA リソースグループに設定されたすべてのノードがリストされます。

5. セカンダリクラスタノード上で，クラスタサービスを再起動する。

再起動によって，これまでの設定内容が反映され，NNMi の環境変数が読み込まれます。なお`net stop ClusSvc`，`net start ClusSvc` コマンドを実行することで，サービスの起動停止ができます。

6. (任意) プライマリクラスタノードのリソースグループをオフラインにし，セカンダリクラスタノードのリソースグループをオンラインにすることで，設定をテストする。

重要

作成したリソースグループについて，リソースグループおよびリソースの設定を NNMi の標準値から変更することでサービスが正常に起動しないなどの問題が発生するおそれがあります。

特に，次の設定を標準値より小さい値に変更する場合は，注意が必要です。

- リソースに障害が発生したときに，Cluster サービスがリソースを再起動するまでの期間

WSFC 標準インストールの場合

<resource_group>-APP のプロパティ［**ポリシー**］タブの保留タイムアウトの値（標準設定値 30:00 分）

<resource_group>-APP のDeadlockTimeout の値（標準設定値 2,700,000 ミリ秒）

19.4.4　HA 用に NNMi を設定する（Linux の場合）

ここでは，Linux 環境で HA 用に NNMi を設定する手順を説明します。

NNMi の HA 設定では，新規に NNMi 用のリソースグループを作成します。このため，対象のリソースグループがない状態から設定作業を行ってください。

NNMi の HA 設定を行うスクリプト（nnmhaconfigure.ovpl）は，内部的にクラスタソフトに対してリソースグループや各リソースを作成する処理を行います。設定作業が完了すると，次のリソースグループが設定されます。

表 19-9　Veritas Cluster Server または Symantec Cluster Server での NNMi 用リソースグループの構成

リソース名	リソースタイプ	説明
<resource_group>-ip	IP	仮想 IP アドレスを制御する
<resource_group>-dg	DiskGroup	ディスクグループを制御する
<resource_group>-volume	Volume	ボリュームを制御する
<resource_group>-mount	Mount	共有ファイルシステムを制御する
<resource_group>-app	Application	NNMi の起動/停止/監視を制御する

VCS または SCS の場合，nnmhaconfigure.ovpl がhagrp やhares などのコマンドを内部的に実行して上記のリソースの設定処理を行います。

- *<resource_group>*の部分は HA リソースグループ名に置き換わります。
- リソースの依存関係は，Volume の前提に DiskGroup と IP，Mount の前提に Volume，および Application の前提に Mount と IP がそれぞれ設定されます。
- VCS または SCS がネットワークインタフェースを監視するリソース（VCS または SCS の NIC）は設定されません。必要に応じて追加設定をしてください。
- NNMi の起動処理に時間が掛かりタイムアウトが発生する場合は，「19.8　HA 設定のトラブルシューティング」を参照して，<resource_group>-app の OnlineTimeout 設定を調整してください。

各リソースの設定内容の例を次に示します。

（例）VCS または SCS の設定ファイルmain.cf の定義例

<>で囲んだ部分は，nnmhaconfigure.ovpl で指定した設定項目の値になります。

```
group <resource_group> (
  SystemList = { <node1> = 1 , <node2> = 1}
  UserStrGlobal = ”NNM_INTERFACE=<virtual_host>;HA_LOCALE=<LOCALE>;HA_MOUNT_POINT=<mountpoin
t>”
  )

  Application <resource_group>-app (
     StartProgram = ”/opt/OV/misc/nnm/ha/nnmharg.ovpl NNM -start <resource_group>”
     StopProgram = ”/opt/OV/misc/nnm/ha/nnmharg.ovpl NNM -stop <resource_group>”
     CleanProgram = ”/opt/OV/misc/nnm/ha/nnmharg.ovpl NNM -clean <resource_group>”
     MonitorProgram = ”/opt/OV/misc/nnm/ha/nnmharg.ovpl NNM -monitor
                      <resource_group>”
     OnlineTimeout = 1800
     )
  DiskGroup <resource_group>-dg (
     DiskGroup = <disk_group>
     )
  IP <resource_group>-ip (
     Device = <network_interface_of_virtual_host>
     Address = ”10.208.228.159”
     NetMask = ”255.255.255.0”
     )
  Mount <resource_group>-mount (
     MountPoint = ”<mountpoint>”
     BlockDevice = ”/dev/vx/dsk/<disk_group>/<volume_group>”
     FSType = <type_of_shared_file_systems>
     FsckOpt = ”-y”
     )
  Volume <resource_group>-volume (
     Volume = <volume_group>
     DiskGroup = <disk_group>
     )
  <resource_group>-app requires <resource_group>-ip
  <resource_group>-app requires <resource_group>-mount
  <resource_group>-mount requires <resource_group>-volume
  <resource_group>-volume requires <resource_group>-dg
  <resource_group>-volume requires <resource_group>-ip
```

表 19-10　HA モニタでの NNMi 用リソースグループの構成

設定項目	設定内容（制御スクリプト）
name （起動）	/var/opt/OV/hacluster/*<resource_group>*/cm2_start.sh
termcommand （停止）	/var/opt/OV/hacluster/*<resource_group>*/cm2_stop.sh
patolcommand （監視）	/var/opt/OV/hacluster/*<resource_group>*/cm2_monitor.sh

- *<resource_group>*の部分は HA リソースグループ名に置き換わります。

重要

HA モニタおよび CLUSTERPRO の場合は，nnmhaconfigure.ovpl を使わずに設定作業を行います。設定方法については，リリースノートを参照してください。

(1) プライマリクラスタノードでの NNMi の設定

プライマリクラスタノードで次の手順を実行します。

(a) 前準備

1. 「19.2 HA 用 NNMi を設定するための前提条件の検証」の作業が完了していることを確認する。
2. NNMi がインストールされていない場合は，インストールする。そして，NNMi が正しく動作することを確認する。
3. 次のコマンドを使って，NNMi 設定をバックアップする。

   (例)

   ```
   /opt/OV/bin/nnmbackup.ovpl -scope all -target <directory>
   ```

   このコマンドの詳細については，「20. NNMi のバックアップおよびリストアツール」を参照してください。

   NNMi のクラスタ環境構成において初期状態では，プライマリクラスタノードのデータと，セカンダリクラスタノードのデータが完全に一致している必要があります。このため，ここで取得したバックアップデータを，セカンダリクラスタノードの設定手順でリストアし，データを一致させます。

(b) データの共有ディスクへのコピー

1. 共有ディスクのマウントポイントになるディレクトリを作成する。
2. NNMi HA リソースグループ用に，共有ディスクを用意する。

   重要

   用意した共有ディスクが，次の条件を満たすことを確認してください。

   - フォーマット済みである
   - 十分な空き容量がある
   - ほかのリソースグループで使用されていない

3. 共有ディスクをアクティブ化して，マウントする。

   (例)

   - Linux VCS または SCS でディスク管理に VxVM/VxFS を使う構成の場合

```
vxdg import <disk_group>
vxvol -g <disk_group> startall
mount -t vxfs /dev/vx/dsk/<disk_group>/<volume> <HA_mount_point>
```

共有ディスクのマウントポイントディレクトリが，ユーザーは root，グループは root で，パーミッションには 755 が設定されていることを確認します。

（例）

```
ls -l
```

4. NNMi を停止する。

   /opt/OV/bin/ovstop -c

   /opt/OV/bin/nettrap stop

5. NNMi ファイルを共有ディスクにコピーする。

```
/opt/OV/misc/nnm/ha/nnmhadisk.ovpl NNM -to <HA_mount_point>
```

重要

指定したマウントポイント直下に，ディレクトリ「NNM」が作成されます（<*HA_mount_point*>/NNM）。

格納先ディレクトリ名を変更することはできません。

6. 共有ディスクをマウント解除し，非アクティブ化する。

   （例）

   - VCS または SCS かつ VxVM/VxFS 使用構成の場合

     umount <*HA_mount_point*>

     vxvol -g <*disk_group*> stopall

     vxdg deport <*disk_group*>

(c) NNMi の HA 設定

1. NNMi HA リソースグループを新規に作成する。

```
/opt/OV/misc/nnm/ha/nnmhaconfigure.ovpl NNM
```

このコマンドの設定項目については，「19.9.2 NNMi に付属している HA 設定スクリプト」を参照してください。

共有ディスクタイプはnone ではなく，必ずdisk を指定してください。

（設定例）

HA 設定項目は，nnmhaconfigure.ovpl に対話形式で入力する項目を表示順に並べています。「19.4.2 HA 用に NNMi を設定する」の「表 19-3 NNMi HA プライマリクラスタノードの設定情報」の説明によって検討した内容を入力してください。

HA 設定項目	設定例
HA リソースグループの名前	jp1ha1
仮想ホストの名前	lhost1
仮想ホストのネットワークインタフェース	lan0
共有ファイルシステムのタイプ	disk（必ず disk を指定）
ディスクタイプ	vxfs
ディスクグループ（VCS または SCS だけ）	shdg3
ボリュームグループ	vg03
マウントするディレクトリ	/shdsk1

重要

設定コマンドを実行する前に，次の注意事項を確認してください。

- HA 構成の NNMi はnnmhaconfigure.ovpl 実行時のロケールを使用して起動します。nnmhaconfigure.ovpl 実行時に操作する画面に適切なロケール（LANG 環境変数）が設定されていることを確認してください。

  Linux VCS または Linux SCS の場合：ja_JP.utf8，ja_JP.UTF-8，C，en_US.utf8，en_US.UTF-8，またはzh_CN.utf8

  HA 構成の設定後にロケールを変更する場合は，「19.6　HA 設定のメンテナンス」を参照してください。

- すでにほかのリソースグループやリソースで使われている値をnnmhaconfigure.ovpl に指定すると，リソースの作成が失敗するなどエラーが発生します。ほかで使われていないことを確認してから，nnmhaconfigure.ovpl を実行してください。

- すでに使われているリソースグループ名，IP アドレスやディスクを指定した場合，リソースを作成するために実行したクラスタソフトのコマンドがエラーとなります。エラー発生時点でnnmhaconfigure.ovpl は異常終了し，それまでに作成されたリソースグループやリソースは残ったままとなります。エラーを対処してnnmhaconfigure.ovpl を再実行する前に，クラスタソフトの操作で残っているリソースを削除してください。

- nnmhaconfigure.ovpl 実行時に，次のメッセージが出力される場合がありますが，内部処理でのメッセージであり問題ありません。

  「ディスク グループが見つかりません。インポートを試みます。」

  「Unable to perform the security token exchange with cmclconfd on node xxxxx

  Cannot connect to configuration daemon (cmclconfd) on node xxxxx」

（実行例）

設定例の値を指定した場合の画面表示例です。"？"の後ろが入力する項目です。

- VCS または SCS (Linux)の場合の実行例

```
# /opt/OV/misc/nnm/ha/nnmhaconfigure.ovpl NNM
質問: HA リソース グループの名前を入力してください: ? jp1ha1

プライマリ ノードの設定が検出されました。

質問: 有効な仮想ホストの名前を入力してください: ? lhost1
情報: ネットワーク インタフェース情報の使用:

ネットワーク インタフェース: bond0
ネットワーク サブネット マスク: 255.255.255.0

選択可能な値:
1: disk
2: none
質問: 共有ファイル システムのタイプを入力してください (disk, none): ? 1

選択可能な値:
1: vxfs
2: ext2
3: ext3
質問: ディスク タイプの名前を入力してください: ? 1
質問: ディスク グループの名前を入力してください: ? shdg3
ディスク グループが見つかりません。インポートを試みます。
質問: ボリューム グループの名前を入力してください: ? shvol3
質問: ディスクをマウントするディレクトリを入力してください: ? /shdsk1
リソース グループを作成しています。
VCS NOTICE V-16-1-10136 Group added; populating SystemList and setting the Parallel attri
bute recommended before adding resources
HA 値の /var/opt/OV/shared/nnm/conf/ov.conf を設定しています。
ブート スクリプトを削除しています。
注: 指定されている仮想ホスト名に一致するようにNNMi FQDNを更新しています。fqdn を lhost1
に設定しています。

ドメインを xxx.xxx に設定しています。

新しい SSL 証明書を生成しています。

lhost1.xxx.xxx.selfsigned のキーストアの証明書を生成しています。
[成功]

生成された証明書をトラストストアにエクスポートしています。

証明書がファイル<temporary.cert>に保存されました。
証明書がキーストアに追加されました。
#
```

2. プライマリクラスタノード上で，サービスの自動起動を無効にする。

   次のコマンドを実行し，サービスの自動起動を無効化します。

   ```
   systemctl disable netmgt.service
   systemctl disable nettrap.service
   systemctl stop netmgt.service
   systemctl stop nettrap.service
   ```

3. VCS または SCS の場合，作成したリソースを有効化（Enabled を 1 に設定）する。

例

```
hares -modify <resource_group>-app Enabled 1
hares -modify <resource_group>-dg Enabled 1
hares -modify <resource_group>-ip Enabled 1
hares -modify <resource_group>-mount Enabled 1
hares -modify <resource_group>-volume Enabled 1
```

その後，VCS または SCS の設定を読み取り専用にして，VCS または SCS の設定ファイル main.cf を出力させます。

```
haconf -dump -makero
```

VCS または SCS がネットワークインタフェースを監視するリソース（VCS または SCS の NIC，MultiNICA，MultiNICB など）は設定されていませんので，必要に応じて追加設定をしてください。

重要

HTTPS 通信を使用して NNMi サーバーにアクセスする場合，証明書を使用するように設定します。詳細については，「10.3.6　高可用性環境での証明書の使用」を参照してください。

(d)　起動の確認

1. NNMi HA リソースグループを起動する。

```
/opt/OV/misc/nnm/ha/nnmhastartrg.ovpl NNM <resource_group>
```

このコマンドは， HA リソースグループの起動を待ってから，プロンプトを返します。本コマンドでリソースグループの起動を確認してください。

NNMi を正常に起動できなかった場合は，「19.8　HA 設定のトラブルシューティング」を参照してください。

これで，NNMi が HA 下で動作するようになりました。

重要

HA 構成の NNMi の通常のオペレーションでは，ovstart コマンドやovstop コマンドは使わないでください。これらのコマンドは，メンテナンスを目的として操作手順に明示されている場合だけ使用します。HA 構成の NNMi の起動や停止は，クラスタソフトの操作によって HA リソースグループを起動または停止するようにしてください。

(2) セカンダリクラスタノードでの NNMi の設定

(a) 前準備

1.「19.2 HA 用 NNMi を設定するための前提条件の検証」の作業が完了していることを確認する。

2. NNMi がインストールされていない場合は，NNMi をインストールし，正しく動作することを確認する。

3. リストアをする。

「(1) プライマリクラスタノードでの NNMi の設定」の(a)の手順 3.で取得したバックアップデータをセカンダリクラスタノードにリストアします。

```
/opt/OV/bin/nnmrestore.ovpl -force -partial -source <backup_data>
```

このコマンドの詳細については，「20. NNMi のバックアップおよびリストアツール」を参照してください。

(b) NNMi の HA 設定

1. 共有ディスクのマウントポイントを作成する。

このマウントポイントでは，「(1) プライマリクラスタノードでの NNMi の設定」の(b)の手順 1.で作成したマウントポイントと同じ名前を使う必要があります。

2. NNMi を停止する。

```
/opt/OV/bin/ovstop -c
/opt/OV/bin/nettrap stop
```

3. NNMi HA リソースグループを設定する。

```
/opt/OV/misc/nnm/ha/nnmhaconfigure.ovpl NNM
```

コマンドの要求に応じて，HA リソースグループ名を指定します。

(実行例)

```
# /opt/OV/misc/nnm/ha/nnmhaconfigure.ovpl NNM

質問: HA リソース グループの名前を入力してください: ? jp1ha1
セカンダリ ノードの設定が検出されました。
Completed the cluster update
ブート スクリプトを削除しています。
注: 指定されている仮想ホスト名に一致するようにNNMi FQDNを更新しています。fqdn を lhost1.x
xx.xxx に設定しています。

ドメインを .xxx.xxx に設定しています。

新しい SSL 証明書を生成しています。

#
```

4. セカンダリクラスタノード上で，サービスの自動起動を無効にする。

次のコマンドを実行し，サービスの自動起動を無効化します。

```
systemctl disable netmgt.service
systemctl disable nettrap.service
systemctl stop netmgt.service
systemctl stop nettrap.service
```

5. VCS または SCS の場合，HA クラスタに設定変更を反映させる。

```
haconf -dump -makero
```

6. 設定が正常に行われたことを確認する。

```
/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -group <resource_group> -nodes
```

このコマンドの出力には，指定した HA リソースグループに設定されたすべてのノードがリストされます。

7. （任意）プライマリクラスタノードのリソースグループをオフラインにし，セカンダリクラスタノードのリソースグループをオンラインにすることで，設定をテストする。

19.5　共有 NNMi データ

HA 環境で実行する NNMi は，HA クラスタ内の NNMi ノード間でファイルを共有するために，各 NNMi ノードからアクセス可能で HA 製品によって制御された共有ディスクを使う必要があります。

重要

共有ディスクに NFS 接続や CIFS 接続を使う構成はサポートしていません。

19.5.1　NNMi の共有ディスク内のデータ

NNMi を HA 下で実行する場合に，共有ディスクで管理される NNMi のデータファイルは次のとおりです。

ファイルの場所は，次のように，共有ディスク内の場所にマッピングされます。

- Windows
  －%NnmDataDir%は，%HA_MOUNT_POINT%¥NNM¥dataDir にマッピングされます。
- Linux
  －$NnmDataDir は，$HA_MOUNT_POINT/NNM/dataDir にマッピングされます。

共有ディスクに移動される主なディレクトリは，次のとおりです。

- Windows
  －%NnmDataDir%shared¥nnm¥databases¥Postgres
  組み込みデータベース。
  －%NnmDataDir%log¥nnm
  NNMi のロギングディレクトリ。
  －%NnmDataDir%shared¥nnm¥databases¥custompoller
  カスタムポーラー収集のエクスポートディレクトリ
  －%NnmDataDir%nmsas¥NNM¥log
  NNMi の監査ログディレクトリ。
  －%NnmDataDir%nmsas¥NNM¥conf
  監査ログファイルを設定するための NNMi のディレクトリ。
  －%NnmDataDir%nmsas¥NNM¥data
  ovjboss で使われるトランザクションストア。
- Linux
  －$NnmDataDir/shared/nnm/databases/Postgres
  組み込みデータベース。
  －$NnmDataDir/log/nnm

NNMi のロギングディレクトリ。

－$NnmDataDir/shared/nnm/databases/custompoller

カスタムポーラー収集のエクスポートディレクトリ

－$NnmDataDir/nmsas/NNM/log

NNMi の監査ログディレクトリ。

－$NnmDataDir/nmsas/NNM/conf

監査ログファイルを設定するための NNMi のディレクトリ。

－$NnmDataDir/nmsas/NNM/data

ovjboss で使われるトランザクションストア。

これらのファイルは，nnmhadisk.ovpl コマンドによって，ローカルディスクと共有ディスクの間でコピーされます。この項の手順に従って，このコマンドを実行します。コマンド構文の概要については，nnm-ha の man ページを参照してください。

19.5.2　設定ファイルの複製

HA 環境で実行する NNMi は，ファイルレプリケーションを使って，HA クラスタ内のすべての NNMi ノードの NNMi 設定ファイルのコピーを管理します。デフォルトでは，NNMi コマンドの nnmdatareplicator.ovpl が，ファイルレプリケーションを管理します。このコマンドは，ローカルディスクにある設定ファイルの更新を監視し，設定ファイルが更新された場合は共有ディスクにファイルをコピーします。フェイルオーバーが発生した場合，共有ディスクにコピーしておいた最新の設定ファイルをフェイルオーバー先のノードにコピーします。その後，NNMi の起動処理が行われます。

上記の設定ファイルの更新確認とコピー処理は，HA クラスタから定期的に実行される NNMi の監視処理の中で行われます。このため，設定ファイル変更後のコピー処理前にノード切り替えが発生すると，変更された設定が反映されません。このような場合は，再度設定を変更してください。

nnmdatareplicator.conf ファイルには，データレプリケーションに含める NNMi のフォルダとファイルを指定します。

データレプリケーションプロセスの詳細については，nnm-ha の man ページを参照してください。

19.6 HA 設定のメンテナンス

19.6.1 NNMi をメンテナンスモードにする

メンテナンスモードは，NNMi のメンテナンス作業を行うために一時的にフェイルオーバーを抑止する機能です。

HA 環境で実行する NNMi は，HA 製品によって NNMi の稼働状態が監視されていて，NNMi が停止した場合，異常発生と判定されて別ノードにフェイルオーバーをします。このため，メンテナンス作業を行うために意図的に NNMi を停止してもフェイルオーバーが発生してしまいます。

メンテナンスモードでは，NNMi の監視を抑止することによってフェイルオーバーの発生を抑止します。これによってアクティブなクラスタノード上で`ovstart` コマンドや`ovstop` コマンドを実行してメンテナンス作業を行うことができます。なお，パッシブなクラスタノードでは`ovstart` コマンドや`ovstop` コマンドは絶対に実行しないでください。

> **重要**
>
> NNMi を前提としている関連製品を実行している場合，NNMi だけをメンテナンスモードにしても関連製品に異常が起きるとフェイルオーバーが発生します。この場合は，関連製品を停止またはメンテナンスモード相当の状態にしてから，NNMi をメンテナンスモードにしてください。

(1) NNMi をメンテナンスモードにする

NNMi をメンテナンスモードにすると，NNMi の監視が無効になります。NNMi がメンテナンスモードになっていると，その HA リソースグループの NNMi の停止や起動を行ってもフェイルオーバーは行われません。

NNMi をメンテナンスモードにするには，アクティブなクラスタノードで次のファイルを作成します。ファイルは空でかまいません。

- Windows：`%NnmDataDir%hacluster¥<resource_group>¥maintenance`
- Linux：`$NnmDataDir/hacluster/<resource_group>/maintenance`

(2) NNMi のメンテナンスモードを解除する

NNMi のメンテナンスモードを解除すると，NNMi の監視が再び有効になります。NNMi を停止すると，HA リソースグループはパッシブなクラスタノードへフェイルオーバーします。

HA リソースグループのメンテナンスモードの解除は，次の手順を実行します。

1. NNMi が正しく実行していることを確認する。

```
ovstatus -c
```

すべての NNMi サービスで，[実行中] 状態が表示される必要があります。

2. メンテナンスが開始される前にアクティブだったクラスタノードから，メンテナンスファイルを削除する。

   メンテナンスファイルについては，「(1)　NNMi をメンテナンスモードにする」を参照してください。

19.6.2　HA クラスタ内の NNMi をメンテナンスする

メモ

HA モニタおよび CLUSTERPRO の場合は，手順が一部異なります。手順については，リリースノートを参照してください。

(1)　NNMi の起動と停止

NNMi を HA 下で実行している場合は，HA のメンテナンスが目的の指示がないかぎり，ovstart コマンドやovstop コマンドは，使わないでください。通常のオペレーションでは，HA 製品の適切なコマンドまたは NNMi の HA コマンド（nnmhastartrg.ovpl やnnmhastoprg.ovpl）を使って，HA リソースグループの起動や停止を行います。

(2)　クラスタ環境で NNMi のホスト名や IP アドレスを変更する

(a)　仮想ホスト名の変更

HA 設定の完了後に NNMi の仮想ホスト名を変更することはできません。仮想ホスト名を変更するには，NNMi の HA 設定を解除し，新しい仮想ホスト名で HA 設定をし直してください。

(b)　仮想 IP アドレスの変更

NNMi HA リソースグループの仮想 IP アドレスを変更するには，アクティブなクラスタノードで次の手順を実行します。

1. NNMi HA リソースグループを停止する。
   - Windows：%NnmInstallDir%misc¥nnm¥ha¥nnmhastoprg.ovpl NNM <*resource_group*>
     net stop NnmTrapReceiver
   - Linux：/opt/OV/misc/nnm/ha/nnmhastoprg.ovpl NNM <*resource_group*>
     /opt/OV/bin/nettrap stop
2. 新しい IP アドレスを使うように，クラスタ設定を変更する。

- Windows：クラスタの管理コンソールで IP アドレスリソースの設定を変更します。リソースグループを開き，<resource_group>-ip をダブルクリックしてパラメータータブを選択し，新しい IP アドレスを入力します。
- Linux：/opt/OV/misc/nnm/ha/nnmhargconfigure.ovpl NNM <resource_group> -set_value <resource_group>-ip Address <new_IP_address>

  haconf -dump -makero を実行して HA クラスタに設定変更を反映させます。

3. NNMi HA リソースグループを起動する。
   - Windows：%NnmInstallDir%misc¥nnm¥ha¥nnmhastartrg.ovpl NNM <resource_group>
   - Linux：/opt/OV/misc/nnm/ha/nnmhastartrg.ovpl NNM <resource_group>
4. NNMi を正常に起動できたことを確認する。
   - Windows：%NnmInstallDir%bin¥ovstatus -c
   - Linux：/opt/OV/bin/ovstatus -c

(c)　物理ホスト名の変更

クラスタ環境で物理ホスト名を変更する場合，NNMi を停止してからシステムの物理ホスト名を変更してください。NNMi として必要な設定はありません。物理ホスト名を変更後，NNMi を起動してください。

なお，Windows のクラスタ環境では，物理ホスト名（コンピューター名）を変更することはできません。

(d)　物理 IP アドレスの変更

手順については，「22.4　スタンドアロンの NNMi 管理サーバーの IP アドレスを変更する」を参照してください。

(3)　フェイルオーバーを行わせないように NNMi を停止する

NNMi のメンテナンスを行う必要がある場合は，アクティブなクラスタノードの NNMi を，パッシブなクラスタノードへフェイルオーバーさせないように停止できます。アクティブなクラスタノードで次の手順を実行します。

1. NNMi を前提としている関連製品がある場合，まず関連製品を停止またはメンテナンスモード相当の状態にする。
2. 「(1)　NNMi をメンテナンスモードにする」に従って，HA リソースグループをメンテナンスモードにする。
3. NNMi を停止する。

```
ovstop -c
```

(4)　メンテナンス後に NNMi を再起動する

フェイルオーバーしないように NNMi を停止した場合は，次の手順を実行して，NNMi と HA 監視を再起動します。

1. NNMi を起動する。

```
ovstart -c
```

2. NNMi を正常に起動できたことを確認する。

```
ovstatus -c
```

すべての NNMi サービスで，［実行中］状態が表示される必要があります。

3. 「(2)　NNMi のメンテナンスモードを解除する」に従って，HA リソースグループのメンテナンスモードを解除する。
4. NNMi の関連製品を停止またはメンテナンスモード相当の状態にしていた場合は，元の状態に戻す。

(5)　HA 構成の NNMi のバックアップ

(a)　オンラインバックアップ

オンラインバックアップを行う場合は，アクティブなクラスタノードで共有ディスクにアクセスできることを確認してから，通常のバックアップ手順を実施してください。

(b)　オフラインバックアップ

HA 構成の NNMi のオフラインバックアップデータを取得する場合は，次の手順を実施します。手順に記載しているメンテナンスモードについては「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

1. HA クラスタ内のアクティブなクラスタノードを特定する。
   - Windows：`%NnmInstallDir%misc¥nnm¥ha¥nnmhaclusterinfo.ovpl -group <resource_group> -activeNode`
   - Linux：`/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -group <resource_group> -activeNode`
2. アクティブなクラスタノードをメンテナンスモードにする。
3. NNMi を停止する。

```
ovstop -c
```

4. HA 製品の操作で共有ディスクがオンラインであることを確認する。オフラインであればオンラインに変更する。
5. 共有ディスクにアクセスできることを確認したあと，`nnmbackup.ovpl` コマンドを実行してオフラインバックアップを実施し，バックアップデータを取得する。

6. NNMi を起動させる。

```
ovstart -c
```

NNMi の起動が完了するのを待ちます。

7. NNMi サービス起動後，メンテナンスモードを解除する。

(6) HA 構成の NNMi のリストア

バックアップデータをリストアするときは次の手順を実施します。手順に記載しているメンテナンスモードについては「19.6.1 NNMi をメンテナンスモードにする」を参照してください。

重要

シングル構成の NNMi で取得したバックアップデータをクラスタ構成の NNMi にリストアしないでください。

1. クラスタとして正常に動作し，NNMi が HA 構成に設定されている状態にする。

   例えば，ハードウェア障害などでシステムの一部または全体が失われた場合，NNMi が HA 構成として動作できる状態にシステムを復旧してください。

2. リストアを実施するノードをアクティブなクラスタノードにする。
3. アクティブなクラスタノードをメンテナンスモードにする。
4. リストアを実施する。
   - `nnmbackup.ovpl` コマンドで取得したバックアップデータの場合

     `nnmrestore.ovpl` コマンドを使用してリストアを実施してください。
   - `nnmbackupembdb.ovpl` コマンドで取得したバックアップデータの場合

     `nnmrestoreembdb.ovpl` コマンドを使用してリストアを実施してください。

     重要

     別ノードで取得したバックアップデータを使用して，`nnmrestore.ovpl` コマンドでリストアを実行する場合は，別ノードのライセンスが適用されないよう，`-lic` オプションを付与しないでください。

5. NNMi を起動する。

```
ovstart -c
```

6. メンテナンスモードを解除する。
7. もう一方のノードで手順 2.～手順 6.を実施する。

   この手順は各ノードのローカルディスク上の設定ファイルを同じ状態にするために行います。同じバックアップデータを使ってリストアを行ってください。

なお，nnmrestoreembdb.ovpl コマンドでのリストアは共有ディスク上のデータベースへのリストアを行うため，任意の 1 つのノードだけで実施してください。

(7) データベースの初期化

1. HA クラスタ内のアクティブなクラスタノードを特定する。
   - Windows：%NnmInstallDir%misc¥nnm¥ha¥nnmhaclusterinfo.ovpl -group <*resource_group*> -activeNode
   - Linux：/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -group <*resource_group*> -activeNode
2. アクティブなクラスタノードをメンテナンスモードにする。
3. 共有ディスクにアクセスできることを確認する。
4. nnmresetembdb.ovpl コマンドを引数なしで実行して，データベースを初期化する。
   - Windows：%NnmInstallDir%bin¥nnmresetembdb.ovpl
   - Linux：/opt/OV/bin/nnmresetembdb.ovpl
5. ovstatus -c を実行し，NNMi サービスが起動していることを確認する。
6. メンテナンスモードを解除する。

19.7　HA クラスタ内の NNMi の設定を解除する

NNMi ノードを HA クラスタから削除する手順には，NNMi のインスタンスの HA 設定を解除する手順も含まれます。設定を解除すると，NNMi のインスタンスをスタンドアロン管理サーバーとして実行できます。また，そのノードから NNMi をアンインストールできます。

HA クラスタの NNMi の設定を完全に解除するには，次の順序で解除作業をしてください。

- 19.7.1　アクティブなクラスタノードの特定
- 19.7.2　パッシブなクラスタノードでの設定解除
- 19.7.3　アクティブなクラスタノードでの設定解除

なお，アクティブなクラスタノードの設定解除では，NNMi のデータを削除する場合と，HA 解除以降もシングルサーバーとして NNMi のデータを続けて使う場合の両方を説明します。

メモ

HA モニタおよび CLUSTERPRO の場合は，`nnmhaunconfigure.ovpl` を使わないで設定解除作業を行います。設定方法については，リリースノートを参照してください。

19.7.1　アクティブなクラスタノードの特定

1. HA クラスタ内のアクティブなクラスタノードを特定する。
   - Windows：`%NnmInstallDir%misc¥nnm¥ha¥nnmhaclusterinfo.ovpl -group <resource_group> -activeNode`
   - Linux：`/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -group <resource_group> -activeNode`

19.7.2　パッシブなクラスタノードでの設定解除

1. パッシブなクラスタノードで，HA クラスタから NNMi の設定を解除する。
   - Windows：`net stop NnmTrapReceiver`
     `%NnmInstallDir%misc¥nnm¥ha¥nnmhaunconfigure.ovpl NNM <resource_group>`
   - Linux：`/opt/OV/bin/nettrap stop`
     `/opt/OV/misc/nnm/ha/nnmhaunconfigure.ovpl NNM <resource_group>`

   このコマンドによって HA リソースグループのノード一覧から該当するノードを解除します。ほかのノードで HA 構成の NNMi を実行するための設定や共有ディスクのデータへの変更は行いません。

   なお，次のメッセージが出力される場合がありますが，問題ありません。

   - Windows の場合

警告:クラスタレジストリにあるリソースグループ xxxxx のパブリックエントリ PUBLIC.HA_MOUNT_POINT に値がありません。

> **重要**
>
> VCS または SCS で NNMi が HA 構成の時に物理ホスト名を変更した環境の場合，HA 設定を解除する前に HA 設定時の物理ホスト名に戻す必要があります。シングルサーバー構成に移行した後に継続して変更後のホスト名を使用する場合は，HA 設定を解除した後に「22.5　NNMi 管理サーバーのホスト名またはドメイン名を変更する」を参照して設定を行ってください。

2. VCS または SCS の場合，HA クラスタに設定変更を反映させる。

   `haconf -dump -makero`

3. NNMi ノードの FQDN 設定を物理ホスト名に変更する。

   a　`nms-local.properties` ファイルを編集します。

   ファイルのパス

   - Windows：`%NnmDataDir%conf¥nnm¥props¥nms-local.properties`
   - Linux：`/var/opt/OV/conf/nnm/props/nms-local.properties`

   編集内容

   `com.hp.ov.nms.fqdn = 仮想ホスト名`

   ここでは物理ホスト名ではなく仮想ホスト名を設定してください。

   b　`nmsetofficialfqdn.ovpl` コマンドを実行します。

   *<FQDN>*には物理ホスト名（`hostname` コマンドで表示されるホスト名）の FQDN を指定してください。

   - Windows：`%NnmInstallDir%bin¥nnmsetofficialfqdn.ovpl -force <FQDN>`
   - Linux：`/opt/OV/bin/nnmsetofficialfqdn.ovpl -force <FQDN>`

   このコマンドによって HA 設定時に仮想ホスト名に変更した FQDN 設定を，物理ホスト名の FQDN に変更します。

   なお，コマンド実行時に次のメッセージが出力される場合があります。

   - 「シングルサインオンが正しく機能するには，新しい証明書を手動で生成する必要があります。」が表示された場合

     シングルサインオンはサポートしていないため，このメッセージは無視してください。

   これによって自己署名証明書が使用されます。NNMi の通信で別の自己署名証明書，または認証機関（CA）署名の証明書を使用する場合は，追加の手順を実行する必要があります。証明書の詳細については，「10.　NNMi での証明書の使用」を参照してください。

4. NNMi HA リソースグループ固有のファイルを安全に保持できるように別の場所に移動する。

   NNMi HA リソースグループを再設定する予定がない場合，次のファイルのコピーを保存する必要はありません。この時点でファイルを削除してかまいません。

   - Windows

エクスプローラで，%NnmDataDir%hacluster¥<*resource_group*>¥フォルダを削除します。

- Linux

  ```
  cd /var/opt/OV/hacluster/
  rm -r <resource_group>
  ```

5. 次のファイルを削除します。

   ファイルのパス

   ・Windows

   エクスプローラで，%NnmDataDir%shared¥nnm¥databases¥nnmdatareplicator¥DataReplicator.db を削除します。

   ・Linux

   ```
   rm /var/opt/OV/shared/nnm/databases/nnmdatareplicator/DataReplicator.db
   ```

6. サービスの自動起動を有効にする。

   - Windows

     スタートメニューの［管理ツール］＞［サービス］で［NNM Trap Receiver］および［NNM Trap Receiver Manager］を選択し，「スタートアップの種類」を「自動」に設定してください。

   - Linux

     ```
     systemctl enable netmgt.service
     systemctl enable nettrap.service
     systemctl start netmgt.service
     systemctl start nettrap.service
     ```

     なお，systemctl start コマンドを実行すると NNMi サービスが起動します。

   以上の手順で終了です。

   HA クラスタから NNMi を完全に解除する場合は，パッシブなクラスタノードを解除後，アクティブなクラスタノードでの設定解除を実施してください。

19.7.3 アクティブなクラスタノードでの設定解除

1. アクティブなクラスタノードで，NNMi HA リソースグループ，NnmTrapReceiver を停止する。

   - Windows：%NnmInstallDir%misc¥nnm¥ha¥nnmhastoprg.ovpl NNM <*resource_group*>

     ```
     net stop NnmTrapReceiver
     ```

   - Linux：$NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM <*resource_group*>

     ```
     /opt/OV/bin/nettrap stop
     ```

> **メモ**
>
> VCS または SCS で NNMi が HA 構成の時に物理ホスト名を変更した環境の場合，リソースグループを停止する前に HA 設定時の物理ホスト名に戻す必要があります。シングルサーバー構成に移行した後に継続して変更後のホスト名を使用する場合は，HA 設定を解除した後に「22.5　NNMi 管理サーバーのホスト名またはドメイン名を変更する」を参照して設定を行ってください。

2. アクティブなクラスタノードで，HA クラスタから NNMi の設定を解除する。

   - Windows：`%NnmInstallDir%misc¥nnm¥ha¥nnmhaunconfigure.ovpl NNM <resource_group>`
   - Linux：`$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM <resource_group>`

   このコマンドによって HA リソースグループのフェイルオーバー対象一覧から該当するノードを解除します。

   このコマンドによって，共有ディスクへのアクセス権は失われますが，ディスクグループやボリュームグループの設定が解除されるわけではありません。

   なお，次のメッセージが出力される場合がありますが，問題ありません。

   - WSFC の場合

     警告:クラスタレジストリにあるリソースグループ xxxxx のパブリックエントリ PUBLIC.HA_MOUNT_POINT に値がありません。

   - VCS または SCS の場合

     VCS WARNING V-16-1-10133 Group does not exist: *<resource_group>*

3. NNMi ノードの FQDN 設定を物理ホスト名に変更する。

   a　`nms-local.properties` ファイルを編集します。

   ファイルのパス

   ・Windows：`%NnmDataDir%conf¥nnm¥props¥nms-local.properties`

   ・Linux：`/var/opt/OV/conf/nnm/props/nms-local.properties`

   編集内容

   `com.hp.ov.nms.fqdn` ＝ 仮想ホスト名

   ここでは物理ホスト名ではなく仮想ホスト名を設定してください。

   b　`nnmsetofficialfqdn.ovpl` コマンドを実行します。

   *<FQDN>*には物理ホスト名（hostname コマンドで表示されるホスト名）の FQDN を指定してください。

   - Windows：`%NnmInstallDir%bin¥nnmsetofficialfqdn.ovpl -force <FQDN>`
   - Linux：`/opt/OV/bin/nnmsetofficialfqdn.ovpl -force <FQDN>`

   このコマンドによって HA 設定時に仮想ホスト名に変更した FQDN 設定を，物理ホスト名の FQDN に変更します。

   なお，コマンド実行時に次のメッセージが出力される場合があります。

- 「シングルサインオンが正しく機能するには，新しい証明書を手動で生成する必要があります。」が表示された場合

  シングルサインオンはサポートしていないため，このメッセージは無視してください。

これによって自己署名証明書が使用されます。NNMi の通信で別の自己署名証明書，または認証機関（CA）署名の証明書を使用する場合は，追加の手順を実行する必要があります。

証明書の詳細については，「10. NNMi での証明書の使用」を参照してください。

4. アクティブなクラスタノードで，NNMi HA リソースグループ固有のファイルを安全に保持できるように別の場所に移動する。

   NNMi HA リソースグループを再設定する予定がない場合，次のファイルのコピーを保存する必要はありません。この時点でファイルを削除してかまいません。

   - Windows

     エクスプローラで，`%NnmDataDir%hacluster¥<resource_group>¥`フォルダを削除します。

   - Linux

     ```
     cd /var/opt/OV/hacluster
     rm -r <resource_group>
     ```

5. 次のファイルを削除します。

   ファイルのパス

   ・Windows

   エクスプローラで，`%NnmDataDir%shared¥nnm¥databases¥nnmdatareplicator¥DataReplicator.db` を削除します。

   ・Linux

   ```
   rm /var/opt/OV/shared/nnm/databases/nnmdatareplicator/DataReplicator.db
   ```

6. 共有ディスクをマウントする。

   OS やクラスタの操作によって，共有ディスクにアクセスできる状態にしてください。

   （例）

   ［サーバーマネージャ］の**［記憶域］**サービスのディスクの管理画面で，共有ディスクがマウントされていたディスクを右クリックして，**［オンライン］**をクリックします。

7. 元のアクティブなクラスタノードに共有ディスクの NNMi ファイルをコピーする。

   この手順は次の条件のどちらかに該当する場合に，実施してください。

   - HA 構成時のデータベースをシングルサーバー構成に移して NNMi を運用する場合
   - HA 構成時に，`nnmchangeembdbpw.ovpl` によって DB のパスワードを変更した場合

   次のコマンドを実行し，元アクティブなクラスタノードに共有ディスクの NNMi ファイルをローカルディスク上にコピーします。

   - Windows：`%NnmInstallDir%misc¥nnm¥ha¥nnmhadisk.ovpl NNM -from <HA_mount_point>`
   - Linux：`/opt/OV/misc/nnm/ha/nnmhadisk.ovpl NNM -from <HA_mount_point>`

8. 共有ディスク上の NNM フォルダまたは NNM ディレクトリを削除する。
9. 共有ディスクのマウントを解除する。

   (例)

   [サーバーマネージャ] の **[記憶域]** サービスのディスクの管理画面で，共有ディスクがマウントされているディスクを右クリックして，**[オフライン]** をクリックします。
10. サービスの自動起動を有効にする。
    - Windows

      スタートメニューの **[管理ツール]** > **[サービス]** で **[NNM Trap Receiver]** および **[NNM Trap Receiver Manager]** を選択し，「スタートアップの種類」を「自動」に設定してください。
    - Linux

      ```
      systemctl enable netmgt.service
      systemctl enable nettrap.service
      systemctl start netmgt.service
      systemctl start nettrap.service
      ```

      なお，`systemctl start` コマンドを実行すると NNMi サービスが起動します。

    メモ

    元のアクティブなクラスタノードで NNMi を実行する場合は，ここまでの手順で準備が完了しています。

    `ovstart` を実行して NNMi を起動してください。

    以降の手順は次の条件のどちらかに該当する場合に，実施してください。

    - HA 構成時のデータベースをシングルサーバー構成に移して，元のパッシブなクラスタノードで NNMi を運用する場合
    - HA 構成時に，`nnmchangeembdbpw.ovpl` によって DB のパスワードを変更した場合

11. 元のアクティブなクラスタノードで次のコマンドを使って，NNMi 設定をバックアップする。

    これによって手順 7.で共有ディスクからローカルにコピーしたデータを含めたバックアップが取得されます。
    - Windows：`%NnmInstallDir%bin¥nnmbackup.ovpl -type offline -scope all -target <directory>`
    - Linux：`/opt/OV/bin/nnmbackup.ovpl -type offline -scope all -target <directory>`
12. HA 構成時のデータを使って NNMi を実行したい元のパッシブなクラスタノードで，手順 11.で取得したアクティブなクラスタノードのバックアップデータをパッシブなクラスタノードにリストアする。
    - Windows：`%NnmInstallDir%bin¥nnmrestore.ovpl -force -source <backup_data>`
    - Linux：`/opt/OV/bin/nnmrestore.ovpl -force -source <backup_data>`

このコマンドの詳細については，「20.　NNMi のバックアップおよびリストアツール」を参照してください。

19.8 HA 設定のトラブルシューティング

19.8.1 一般的な設定の誤り

HA 設定での一般的な誤りの例を次に示します。

- ディスク設定が正しくない。
  - VCS または SCS を使用している場合で，リソースをプローブできないときは，設定に何らかの間違いがあります。ディスクをプローブできないとき，オペレーティングシステムはディスクにアクセスできなくなることがあります。
  - 手動でディスク設定をテストし，設定が適切であることを HA 製品のマニュアルを参照して確認してください。
- ディスクが使用中で，HA リソースグループで起動できない。

  HA リソースグループを起動する前に，ディスクがアクティブでないことを必ず確認してください。
- WSFC のネットワーク設定が正しくない。

  ネットワークトラフィックが複数の NIC カード上を流れる場合は，`ovjboss` プロセスなどのネットワーク帯域幅を大量に消費するプログラムをアクティブ化すると RDP セッションが失敗します。
- 一部の HA 製品がブート時に自動的に再起動しない。

  ブートアップ時の自動再起動の設定方法については，HA 製品のマニュアルを参照してください。
- NFS，またはほかのアクセスが OS に直接追加される。

  リソースグループ設定でこの動作を管理している必要があります。
- フェイルオーバーの間，または HA リソースグループをオフラインにする間に，カレントディレクトリを共有ディスクのマウントポイントにしている。

  HA は，共有ディスクのマウント解除を阻止するプロセスをすべて抹消します。フェイルオーバーまたはリソースグループのオフライン時には別のディレクトリに移動してください。
- HA クラスタの仮想 IP アドレスを HA リソースの仮想 IP アドレスとして再使用している。

  一方のシステムで有効で，他方では無効となります。それぞれに異なる IP アドレスを設定してください。
- タイムアウトが短過ぎる。

  製品に不具合があると，HA 製品は HA リソースをタイムアウトさせ，フェイルオーバーが実行されます。

  WSFC で，［**リソースが開始するまでの待機時間**］の設定値を確認します。NNMi では，この値は 15 分に設定されますが，この値を増やすことができます。
- メンテナンスモードを使用していない。

  メンテナンスモードは，HA の障害をデバッグするためのモードです。リソースグループがシステムでオンラインになった直後にすぐフェイルオーバーしてしまうような場合に，メンテナンスモードは，システムでリソースグループを維持し，実際に障害のある部分を見つけるのに役立ちます。

- クラスタログを再確認していない。
  クラスタログで多くの一般的な間違いを確認できます。

19.8.2 HA リソーステスト

ここでは，NNMi HA リソースグループのリソースをテストするための一般的な方法を説明します。

このテストで，ハードウェア設定の問題が特定されます。HA 用 NNMi を設定する前に，このテストを実行することをお勧めします。好ましい結果を出した設定値を記録しておき，NNMi HA リソースグループの設定で，それらの値を使用します。

ここに記載されているコマンドの詳細については，HA 製品のマニュアルを参照してください。

HA リソースのテスト手順を次に示します。

1. HA クラスタを起動する。
2. （Windows の場合）HA クラスタに，次の仮想 IP アドレスが定義されていることを確認する。
   - HA クラスタの仮想 IP アドレス
   - HA リソースグループの仮想 IP アドレス

   これらの IP アドレスは，別の場所で使用しないでください。
3. HA リソースグループを HA クラスタに追加する。
   この HA リソースグループには，`test` など，商用名でない名称を使用してください。
4. HA リソースグループへの接続をテストする。
   - 仮想 IP アドレスと，リソースグループに対応する仮想ホスト名を，リソースとして HA リソースグループに追加します。
     あとで，NNMi HA リソースグループに関連づける値を使用します。
   - アクティブなクラスタノードからパッシブなクラスタノードにフェイルオーバーし，HA クラスタが正常にフェイルオーバーすることを確認します。
   - 新しいアクティブなクラスタノードから新しいパッシブなクラスタノードにフェイルオーバーし，フェイルバックを確認します。
   - リソースグループが正しくフェイルオーバーしない場合，アクティブなノードにログオンして，IP アドレスが正しく設定され，アクセスできることを確認します。また，ファイアウォールによって IP アドレスがブロックされていないことも確認します。
   - アクティブなクラスタノードからパッシブなクラスタノードにフェイルオーバーし，HA クラスタが正常にフェイルオーバーすることを確認します。
   - 新しいアクティブなクラスタノードから新しいパッシブなクラスタノードにフェイルオーバーし，フェイルバックを確認します。

- リソースグループが正しくフェイルオーバーしない場合，アクティブなクラスタノードにログオンして，ディスクがマウントされ，使用できることを確認します。

5. 共有ディスクの設定に使用したコマンドおよび入力値の記録を取っておく。

   NNMi HA リソースグループを設定するときに，この情報が必要になる場合があります。

6. 各ノードからリソースグループを削除する。

   - IP アドレスエントリを削除します。
   - リソースグループをオフラインに設定して，ノードからリソースグループを削除します。

   この時点で，NNMi に付属しているツールを使用して，HA 下で実行するように NNMi を設定できます。

19.8.3 一般的な HA のトラブルシューティング

(1) リソースをホストするサブシステムプロセスが予期せず停止する

Windows Server 2016 オペレーティングシステムで，HA クラスタリソースを起動すると，リソースをホストするサブシステム（rhs.exe）プロセスが予期せずに停止します。

この問題の詳細については，次の Web サイトを参照してください。

```
http://support.microsoft.com/kb/978527
```

重要

NNMi リソースを実行するときは，必ず，リソースグループに固有の別個のリソースモニタ（rhs.exe）で実行してください。

(2) 製品の監視タイムアウト

システムログに，次の例のようなメッセージが含まれます。

```
VCS ERROR V-16-2-13027 Thread(...) Resource(<resource group>-app) - monitor procedure did no
t complete within the expected time.
```

このメッセージは，製品が Veritas Cluster Server または Symantec Cluster Server に設定されたタイムアウト値の範囲内でリソースを監視できなかったことを示しています。

Veritas Cluster Server または Symantec Cluster Server のデフォルトで，タイムアウトは 60 秒が適用されます。

Veritas Cluster Server または Symantec Cluster Server に設定されたタイムアウト値を変更するには，次のコマンドを，次の順番で，実行します。

```
/opt/VRTSvcs/bin/haconf -makerw
/opt/VRTSvcs/bin/hares -override <resource_group>-app MonitorTimeout
/opt/VRTSvcs/bin/hares -modify <resource_group>-app MonitorTimeout <value in seconds>
/opt/VRTSvcs/bin/haconf -dump -makero
```

(3) アクティブなクラスタノードのログファイルが更新されない

これは正常です。ログファイルは，共有ディスクにリダイレクトされているため，このような状況になります。

NNMi の場合は，ov.conf ファイル内のHA_NNM_LOG_DIR で指定された場所にあるログファイルを調べてください。

(4) HA リソースグループが特定のクラスタノードでは起動できない

nnmhargconfigure.ovpl コマンド，またはnnmhastartrg.ovpl コマンドで NNMi HA リソースグループを正常に起動／停止／切り替えできない場合は，次の情報を調べてください。

- WSFC の場合

  －フェイルオーバークラスタ管理で，リソースグループおよびそれを構成するリソースの状態を調べてください。

  －イベントビューアのログにエラーが記録されていないか調べてください。

- VCS または SCS の場合

  －/opt/VRTSvcs/bin/hares -state を実行して，リソースの状態を調べます。

  －障害が発生しているリソースでは，障害が発生しているリソース用の/var/VRTSvcs/log/<*resource*>.log ファイルを調べます。リソースは，IP*.log，Mount*.log，Volume*.log などのエージェントタイプで指定します。

原因となっているリソースを特定できない場合は，HA 製品のコマンドを使って，HA リソースグループを手動で起動します。

1. 共有ディスクをマウントする。
2. ネットワークインタフェースに仮想ホストを割り当てる。
   - WSFC の場合

     －フェイルオーバークラスタ管理を起動します。

     －リソースグループを展開します。

     －［<*resource_group*>-ip］を右クリックして，［このリソースをオンラインにする］をクリックします。

   - VCS または SCS の場合

     /opt/VRTSvcs/bin/hares -online <*resource_group*>-ip -sys <*local_hostname*>
3. HA リソースグループを起動する。

例：

- Windows：%NnmInstallDir%misc¥nnm¥ha¥nnmhastartrg.ovpl NNM -start <*resource_group*>
- Linux：$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM -start <*resource_group*>

リターンコード 0 は，NNMi を正常に起動できたことを意味します。

リターンコード 1 は，NNMi を正常に起動できなかったことを意味します。

(5) 「システム エラー XXXX が発生しました」が表示された（Windows の場合）

システム（OS やクラスタソフト）のエラーが発生している場合があります。詳しくは OS やクラスタソフトのマニュアルなどを確認してください。

エラーの例：WSFC でのエラー発生例について説明します。

- 例「システム エラー 5054 が発生しました (0x000013be)。クラスタ ネットワークが無効です。」
  NNMi 用の IP アドレスに，ハートビート用の内部用ネットワークの IP アドレスを指定した場合，IP アドレスリソースの作成のため実行したcluster.exe コマンドで上記のエラーが発生します。
- 例「システム エラー 5057 が発生しました (0x000013c1)。そのクラスタ IP アドレスは既に使われています。」
  NNMi 用の IP アドレスに，既に使われている IP アドレスを指定した場合， IP アドレスリソースの作成のため実行したcluster.exe コマンドで上記のエラーが発生します。

対処：システムエラーの内容について確認し，問題を対策してください。上記の例のように NNMi 用の IP アドレスの指定が適切でない場合は，使用する IP アドレスの見直しを行ってください。

19.8.4 NNMi 固有の HA のトラブルシューティング

この項の内容が適用されるのは，NNMi だけの HA 設定です。

(1) NNMi を HA 下で正常に起動できない

NNMi が正しく起動しない場合，仮想 IP アドレスまたはディスクに関するハードウェアの問題であるのか，アプリケーション障害の問題であるのかをデバッグする必要があります。このデバッグプロセスの間，システムをメンテナンスモードにします。

この問題を解決するには，次の手順を実行します。

1. HAクラスタのアクティブなクラスタノードで，次のメンテナンスファイルを作成して，HAリソースグループの監視を無効にする。
   - Windows：%NnmDataDir%hacluster¥<*resource_group*>¥maintenance
   - Linux：$NnmDataDir/hacluster/<*resource_group*>/maintenance

2. NNMiを起動する。

```
ovstart
```

3. NNMiを正常に起動できたことを確認する。

```
ovstatus -c
```

すべての NNMi サービスで，[実行中] 状態が表示される必要があります。このように表示されない場合，正しく開始していないプロセスをトラブルシューティングします。

4. トラブルシューティングが完了したら，メンテナンスファイルを削除する。

- Windows：%NnmDataDir%hacluster¥<*resource_group*>¥maintenance
- Linux：$NnmDataDir/hacluster/<*resource_group*>/maintenance

(2) NNMi データへの変更がフェイルオーバーのあとに表示されない

NNMi の設定で，NNMi を実行中のシステム以外のシステムが設定されています。この問題を解決するには，ov.conf ファイルに次の項目に対応した適切なエントリがあることを確認します。

- NNM_INTERFACE=<*virtual_hostname*>
- HA_RESOURCE_GROUP=<*resource_group*>
- HA_MOUNT_POINT=<*HA_mount_point*>
- NNM_HA_CONFIGURED=YES
- HA_POSTGRES_DIR=<*HA_mount_point*>/NNM/dataDir/shared/nnm/databases/Postgres
- HA_CUSTOMPOLLER_DIR=<*HA_mount_point*>/NNM/dataDir/shared/nnm/databases/custompoller
- HA_NNM_LOG_DIR=<*HA_mount_point*>/NNM/dataDir/log/nnm
- HA_JBOSS_DATA_DIR=<*HA_mount_point*>/NNM/dataDir/nmsas/NNM/data
- HA_LOCALE=<*ロケール*>（Linuxだけ）
- HA_PERFSPI_ADAPTER_DIR=<*HA_mount_point*>/NNM/dataDir/shared/perfSpi/datafiles

ov.conf ファイルの場所は，「19.9.1 NNMi HA 設定ファイル」を参照してください。

(3) HA の設定後，nmsdbmgr を起動できない

この状況は，通常，nnmhaconfigure.ovpl コマンドを実行したが，-to オプションを指定してnnmhadisk.ovpl コマンドを実行しないで，NNMi を起動した場合に発生します。この状況では，ov.conf ファイルの HA_POSTGRES_DIR エントリは，共有ディスクの場所を指していますが，この場所は NNMi からはアクセスできません。

この問題を解決するには，次の手順を実行します。

1. HA クラスタのアクティブなクラスタノードで，次のメンテナンスファイルを作成して，HA リソースグループの監視を無効にする。
   - Windows：%NnmDataDir%hacluster¥<*resource_group*>¥maintenance
   - Linux：$NnmDataDir/hacluster/<*resource_group*>/maintenance
2. NNMi データベースを共有ディスクにコピーする。
   - Windows：
     %NnmInstallDir%misc¥nnm¥ha¥nnmhadisk.ovpl NNM -to <*HA_mount_point*>
   - Linux：
     $NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM -to <*HA_mount_point*>
3. NNMi HA リソースグループを起動する。
   - Windows：
     %NnmInstallDir%misc¥nnm¥ha¥nnmhastartrg.ovpl NNM <*resource_group*>
   - Linux：
     $NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM <*resource_group*>
4. NNMi を起動する。

```
ovstart
```

5. NNMi を正常に起動できたことを確認する。

```
ovstatus -c
```

   すべての NNMi サービスで，［実行中］状態が表示される必要があります。
6. トラブルシューティングが完了したら，メンテナンスファイルを削除する。
   - Windows：%NnmDataDir%hacluster¥<*resource_group*>¥maintenance
   - Linux：$NnmDataDir/hacluster/<*resource_group*>/maintenance

(4) NNMi が 1 つの HA クラスタノードでだけ正常に実行される（Windows の場合）

Windows オペレーティングシステムには，HA クラスタ用と HA リソースグループ用の 2 つの異なる仮想 IP アドレスが必要です。HA クラスタの仮想 IP アドレスと NNMi HA リソースグループの仮想 IP アドレスが同じ場合，NNMi は，HA クラスタの IP アドレスと関連づけられているノードでだけ正常に実行されます。

この問題を修正するには，HA クラスタの仮想 IP アドレスをネットワークで一意の値に変更します。

(5) ディスクフェイルオーバーが行われない

この状況は，オペレーティングシステムが共有ディスクをサポートしていない場合に発生します。HA 製品，オペレーティングシステム，ディスクのメーカーのマニュアルなどを参照して，これらの製品を混在させて使用できるか確認してください。

ディスク障害が発生すると，NNMi はフェイルオーバーでは起動しません。nmsdbmgr が失敗する理由の多くは，HA_POSTGRES_DIR ディレクトリが存在しないことです。共有ディスクがマウント済みであり，該当するファイルにアクセスできる状態になっていることを確認してください。

(6) 共有ディスクにアクセスできない（Windows の場合）

nnmhaclusterinfo.ovpl -config NNM -get HA_MOUNT_POINT コマンドを実行しても何も表示されない場合，共有ディスクのマウントポイントの設定が不適切のため，共有ディスクにアクセスできません。

共有ディスクのマウントポイントのドライブは，HA 設定時に次のように完全に指定します。

（例）Y:

この問題を修正するには，HA クラスタの各ノードでnnmhaconfigure.ovpl コマンドを実行します。

(7) フェイルオーバー後にセカンダリクラスタノードで共有ディスク上のファイルが見つからない

この状況は，通常，共有ディスクがマウントされていないときに，-to オプションを付けたnnmhadisk.ovpl コマンドを実行した場合に発生します。この場合は，データファイルはローカルディスクにコピーされ，共有ディスクには格納されません。

この問題を解決するには，次の手順を実行します。

1. HA クラスタのアクティブなクラスタノードで，次のメンテナンスファイルを作成して，HA リソースグループの監視を無効にする。
   - Windows：%NnmDataDir%hacluster¥<*resource_group*>¥maintenance
   - Linux：$NnmDataDir/hacluster/<*resource_group*>/maintenance
2. アクティブなクラスタノードにログオンして，ディスクがマウントされ，使用できることを確認する。
3. NNMi を停止する。

   ```
   ovstop
   ```

   - Windows：net stop NnmTrapReceiver
   - Linux：/opt/OV/bin/nettrap stop
4. NNMi データベースを共有ディスクにコピーする。
   - Windows：

%NnmInstallDir%misc¥nnm¥ha¥nnmhadisk.ovpl NNM -to <*HA_mount_point*>

- Linux：

  $NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM -to <*HA_mount_point*>

5. NNMi HA リソースグループを起動する。

- Windows：

  %NnmInstallDir%misc¥nnm¥ha¥nnmhastartrg.ovpl NNM <*resource_group*>

- Linux：

  $NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM <*resource_group*>

6. NNMi を起動する。

```
ovstart
```

7. NNMi を正常に起動できたことを確認する。

```
ovstatus -c
```

すべての NNMi サービスで，**[実行中]** 状態が表示される必要があります。

8. トラブルシューティングが完了したら，メンテナンスファイルを削除する。

- Windows：%NnmDataDir%hacluster¥<*resource_group*>¥maintenance
- Linux：$NnmDataDir/hacluster/<*resource_group*>/maintenance

19.9 HA 設定リファレンス

ここでは，NNMi HA 設定ファイルと NNMi HA 設定のスクリプトおよびログファイルについて説明します。

19.9.1 NNMi HA 設定ファイル

次の表に，NNMi HA 設定ファイルを示します。これらのファイルは，NNMi に適用され，次の場所にインストールされます。

- Windows：`%NnmDataDir%shared¥nnm¥conf`
- Linux：`$NnmDataDir/shared/nnm/conf`

表 19-11 NNMi HA 設定ファイル

ファイル名	説明
ov.conf	このファイルは，NNMi HA 実装の状態を示し，`nnmhaclusterinfo.ovpl` コマンドによって更新されます。NNMi の各プロセスは，このファイルを読み取って，HA 設定を確認します。
nnmdatareplicator.conf	このファイルは，`nnmdatareplicator.ovpl` コマンドで，アクティブなクラスタノードからパッシブなクラスタノードへのデータレプリケーションに含む NNMi のフォルダとファイルを調べるために使われます。NNMi 設定のレプリケーション用に異なる手段を実装する場合は，含めるデータのリストは，このファイルを参照してください。 詳細については，このファイルのコメントを参照してください。

19.9.2 NNMi に付属している HA 設定スクリプト

次の表に，NNMi に付属している HA 設定スクリプトを示します。NNMi に付属しているスクリプトは，カスタマ Perl モジュールを持つすべての製品に HA を設定する場合に使用できる便利なスクリプトです。必要に応じて，HA 製品に付属しているコマンドを使って，NNMi 用に HA を設定できます。

NNMi 管理サーバーでは，NNMi に付属している HA 設定スクリプトは，次の場所にインストールされます。

- Windows：`%NnmInstallDir%misc¥nnm¥ha`
- Linux：`$NnmInstallDir/misc/nnm/ha`

表 19-12 NNMi HA 設定スクリプト

スクリプト名	説明
nnmhaconfigure.ovpl	NNMi を HA クラスタ用に設定します。 このスクリプトは，HA クラスタ内のすべてのノードで実行してください。

スクリプト名	説明
nnmhaunconfigure.ovpl	HA クラスタの NNMi の設定を解除します。 必要に応じて，HA クラスタ内の 1 つ以上のノードでこのスクリプトを実行します。
nnmhaclusterinfo.ovpl	NNMi に関するクラスタ情報を取得します。 このスクリプトは，必要に応じて，HA クラスタ内の任意のノードで実行します。
nnmhadisk.ovpl	データファイルを，NNMi と共有ディスクの間でコピーします。 HA の設定時には，このスクリプトはプライマリクラスタノードで実行します。 それ以外の場合は，この章の手順に従って，このスクリプトを実行します。
nnmhastartrg.ovpl	HA クラスタで NNMi HA リソースグループを起動します。 HA の設定時には，このスクリプトはプライマリクラスタノードで実行します。
nnmhastoprg.ovpl	HA クラスタで NNMi HA リソースグループを停止します。 HA の設定解除時には，このスクリプトはアクティブなクラスタノードで実行します。

表 19-13 に示した NNMi 付属のスクリプトは，表 19-12 に示したスクリプトで使用します。表 19-13 に示したスクリプトは直接実行しないでください。

表 19-13　NNMi HA サポートスクリプト

スクリプト名	説明
nnmdatareplicator.ovpl	`nnmdatareplicator.conf` 設定ファイルを調べて，リモートシステムに送信するファイルの変更やコピーを確認します。
nnmharg.ovpl	HA クラスタの NNMi を起動／停止／監視します。 VCS または SCS 設定では，VCS または SCS の起動／停止／監視のスクリプトで使用します（`nnmhargconfigure.ovpl` で，この使用法を設定します）。 また，トレースを有効／無効にするために，`nnmhastartrg.ovpl` でも使われます。
nnmhargconfigure.ovpl	HA のリソースとリソースグループを設定します。`nnmhaconfigure.ovpl` と `nnmhaunconfigure.ovpl` で使われます。
nnmhastart.ovpl	HA クラスタで NNMi を起動します。`nnmharg.ovpl` で使われます。
nnmhastop.ovpl	HA クラスタの NNMi を停止します。`nnmharg.ovpl` で使われます。
nnmhamonitor.ovpl	HA クラスタの NNMi プロセスを監視します。`nnmharg.ovpl` で使われます。
nnmhamscs.vbs	WSFC の HA クラスタで，NNMi プロセスを起動／停止／監視するスクリプトを作成するためのテンプレートです。生成されるスクリプトは，次の場所に格納され，WSFC で使われます。 `%NnmDataDir%hacluster¥<resource_group>¥hamscs.vbs`

19.9.3　NNMi HA 設定のログファイル

次のログファイルは，NNMi の HA 設定に適用されます。

- Windows 設定

－%NnmDataDir%tmp¥HA_nnmhaserver.log

－%NnmDataDir%log¥haconfigure.log

- Linux 設定

－$NnmDataDir/tmp/HA_nnmhaserver.log

－$NnmDataDir/log/haconfigure.log

- Windows 実行時

－イベントビューアのログ

－%HA_MOUNT_POINT%¥NNM¥dataDir¥log¥nnm¥ovspmd.log

－%HA_MOUNT_POINT%¥NNM¥dataDir¥log¥nnm¥postgres.log

－%HA_MOUNT_POINT%¥NNM¥dataDir¥log¥nnm¥nmsdbmgr.log

－%SystemRoot%¥Cluster¥cluster.log

これは，リソースとリソースグループの追加／削除，ほかの設定上の問題点，起動／停止上の問題点を含むクラスタ実行時の問題点に関するログファイルです。

- VCS または SCS 用の Linux

リソース	ログファイル
<*resource_group*>-app	• /var/VRTSvcs/log/Application_A.log • $HA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log • $HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log • $HA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log • Linux の場合：/var/log/messages* • Solaris の場合：/var/adm/messages*
<*resource_group*>-dg <*resource_group*>-volume <*resource_group*>-mount	• /var/VRTSvcs/log/DiskGroup_A.log • /var/VRTSvcs/log/Volume_A.log • /var/VRTSvcs/log/Mount_A.log • Linux の場合：/var/log/messages* • Solaris の場合：/var/adm/messages*
<*resource_group*>-ip	• /var/VRTSvcs/log/IP_A.log • Linux の場合：/var/log/messages* • Solaris の場合：/var/adm/messages*

注：オペレーティングシステム固有の HA リソース関連の問題は，/var/adm/messages*または/var/log/messages*ファイルを調べてください。<*resource_group*>-app では，プロセスを起動できなかったことに関するメッセージを探してください。

20 NNMi のバックアップおよびリストアツール

どのようなビジネスでも，中断することなく業務を確実に継続するには，バックアップおよびリストアに関して優れた方針を持つことが重要です。NNMi は，ネットワークを運用する上で重要な資産であり，定期的にバックアップする必要があります。NNMi インストールに関連した重要データは，ファイルシステム内のファイル，およびリレーショナルデータベースのデータの 2 種類です。この章では，重要な NNMi ファイルおよびデータをバックアップおよびリストアするために NNMi で装備しているツールについて説明しています。

20.1 バックアップコマンドとリストアコマンド

NNMi には，NNMi データをバックアップおよびリストアするために次のスクリプトがあります。

- `nnmbackup.ovpl`

  必要なすべてのファイルシステムデータ（設定情報を含む）と NNMi データベースに保管されたデータをバックアップします。

- `nnmrestore.ovpl`

  `nnmbackup.ovpl` スクリプトを使用して作成されたバックアップをリストアします。

- `nnmbackupembdb.ovpl`

  NNMi データベース（ファイルシステムデータではない）の完全バックアップを，NNMi の稼働中に作成します。

- `nnmrestoreembdb.ovpl`

  `nnmbackupembdb.ovpl` スクリプトを使用して作成されたバックアップをリストアします。

- `nnmresetembdb.ovpl`

  NNMi データベーステーブルをドロップします。`ovstart` コマンドを実行してテーブルを再作成します。

コマンド構文については，該当するリファレンスページを参照してください。

20.2 NNMi データをバックアップする

NNMi バックアップコマンド（nnmbackup.ovpl）は，主要な NNMi ファイルシステムデータおよび NNMi Postgres データベースのテーブルの一部またはすべてを，指定されたターゲットディレクトリにコピーします。各バックアップ操作によって，ターゲットディレクトリ内のnnm-bak-<TIMESTAMP>という名前の親ディレクトリにファイルが格納されます。-noTimeStamp オプションを指定すると，ディスクスペースを節約できます。-noTimeStamp オプションを使用した場合，親ディレクトリの名前はnnm-bak になります。前回のバックアップ後に-noTimeStamp オプションを使用してバックアップが行われると，前回のバックアップの名前がnnm-bak.previous に変更されて，ローリングバックアップが作成されます。この名前変更は，2 回目のバックアップの完了後，バックアップデータの喪失を防止するために行われます。

NNMi バックアップコマンドは，バックアップデータの tar 形式のアーカイブを作成できます。また，ユーザー独自のツールを使用してバックアップファイルの圧縮もできます。次に，適切なツールを使用して，バックアップのコピーを保存できます。詳細については，nnmbackup.ovpl のリファレンスページを参照してください。

20.2.1 バックアップタイプ

NNMi のバックアップコマンドでは，2 種類のバックアップがサポートされます。

- オンラインバックアップは NNMi の稼働中に行われます。NNMi では，バックアップされたデータ内でデータベーステーブルが確実に同期されます。オンラインバックアップ中でも，オペレータは制約を受けることなく NNMi コンソールを使用でき，ほかのプロセスは NNMi データベースとやり取りできます。オンラインバックアップを実行することで，バックアップ領域に記載されているように，機能に応じて NNMi のデータすべてまたはデータの一部だけをバックアップできます。NNMi データベースの場合は，nmsdbmgr サービスが実行されている必要があります。
- オフラインバックアップは，NNMi が完全に停止している間に行われます。オフラインバックアップでは，バックアップ領域がファイルシステムのファイルにだけ適用されます。オフラインバックアップには，バックアップ領域に関係なく，必ず NNMi データベースの全体が含まれます。NNMi データベースの場合，このバックアップでは Postgres データベースのファイルがコピーされます。

20.2.2 バックアップ領域

NNMi バックアップコマンドでは，NNMi のバックアップ量を定義する領域を幾つか指定できます。

設定領域

設定領域（-scope config）は，大まかには NNMi コンソールの［設定］ワークスペース内の情報と一致します。

設定領域には次のデータが含まれます。

- オンラインバックアップの場合は，NNMi 設定情報を保存しているデータベーステーブルだけ。

- オフラインバックアップの場合は，データベース全体。
- オンラインバックアップ，オフラインバックアップともに，「表 20-1　設定領域ファイルとディレクトリ」のリストに示すファイルシステム内の NNMi 設定情報。

トポロジ領域

トポロジ領域（-scope topology）は，大まかには NNMi コンソールの［インベントリ］ワークスペース内の情報と一致します。ネットワークトポロジが依存している設定はそのトポロジの検出に使用されているため，トポロジ領域には設定領域が含まれます。

トポロジ領域には次のデータが含まれます。

- オンラインバックアップの場合は，NNMi 設定情報とネットワークトポロジ情報を保存しているデータベーステーブルだけ。
- オフラインバックアップの場合は，データベース全体。
- オンラインバックアップ，オフラインバックアップともに，「表 20-1　設定領域ファイルとディレクトリ」のリストに示すファイルシステム内の NNMi 設定情報。現在，トポロジ領域に関連づけられているファイルシステムのファイルはありません。

イベント領域

イベント領域（-scope events）は，大まかには NNMi コンソールの［インシデントの参照］ワークスペース内の情報と一致します。イベントはこれらのイベントに関連したネットワークトポロジに依存しているため，イベント領域には設定領域とトポロジ領域が含まれます。

イベント領域には次のデータが含まれます。

- オンラインバックアップの場合は，NNMi 設定情報，ネットワークトポロジ情報およびイベント情報を保存しているデータベーステーブルだけ。
- オフラインバックアップの場合は，データベース全体。
- オンラインバックアップ，オフラインバックアップともに，「表 20-1　設定領域ファイルとディレクトリ」のリストに示すファイルシステム内の NNMi 設定情報と，「表 20-2　イベント領域ファイルとディレクトリ」のリストに示す NNMi イベント情報。

全領域

完全バックアップ（-scope all）には，NNMi のすべての重要ファイルとデータベース全体が含まれます。

表 20-1　設定領域ファイルとディレクトリ

ディレクトリまたはファイル名	説明
%NnmInstallDir%conf（Windows だけ）	設定情報
%NnmInstallDir%misc¥nms¥lic $NnmInstallDir/misc/nms/lic	そのほかのライセンス情報
%NnmDataDir%nmsas¥NNM¥conf $NnmDataDir/nmsas/NNM/conf	JBoss の設定
%NnmDataDir%conf	設定情報

ディレクトリまたはファイル名	説明
$NnmDataDir/conf	設定情報
%NnmDataDir%conf¥nnm¥props $NnmDataDir/conf/nnm/props	ローカル NNMi 設定のプロパティファイル
%NnmDataDir%shared¥nnm¥conf¥licensing¥LicFile.txt $NnmDataDir/shared/nnm/conf/licensing/LicFile.txt	ライセンス情報
%NnmDataDir%NNMVersionInfo $NnmDataDir/NNMVersionInfo	NNMi バージョン情報ファイル
%NnmDataDir%shared¥nnm¥user-snmp-mibs $NnmDataDir/shared/nnm/user-snmp-mibs	共有ユーザー追加の SNMP MIB 情報
%NnmDataDir%shared¥nnm¥actions $NnmDataDir/shared/nnm/actions	共有ライフサイクルの移行アクション
%NnmDataDir%shared¥nnm¥certificates $NnmDataDir/shared/nnm/certificates	共有 NNMi SSL 証明書
%NnmDataDir%shared¥nnm¥conf $NnmDataDir/shared/nnm/conf	共有 NNMi 設定情報
%NnmDataDir%shared¥nnm¥conf¥licensing $NnmDataDir/shared/nnm/conf/licensing	共有 NNMi ライセンス設定情報
%NnmDataDir%shared¥nnm¥lrf $NnmDataDir/shared/nnm/lrf	共有 NNMi コンポーネント登録ファイル
%NnmDataDir%shared¥nnm¥conf¥props $NnmDataDir/shared/nnm/conf/props	共有 NNMi 設定のプロパティファイル
%NnmDataDir%shared¥nnm¥www¥htdocs¥images $NnmDataDir/shared/nnm/www/htdocs/images	共有 NNMi ノードグループマップ背景イメージ

このコンテキストで共有ディレクトリのファイルは，NNMi アプリケーションフェイルオーバーまたは高可用性環境の別の NNMi 管理サーバーと共有されるファイルです。

表 20-2　イベント領域ファイルとディレクトリ

ディレクトリまたはファイル名	説明
%NnmDataDir%log¥nnm¥signin.log $NnmDataDir/log/nnm/signin.log	NNMi コンソールサインインログ

20.3 NNMi データをリストアする

NNMi リストアスクリプト（nnmrestore.ovpl）は，バックアップデータを NNMi 管理サーバーに配置します。バックアップの種類と領域によって，NNMi でリストア可能なバックアップデータが決まります。

メモ

nnmrestore.ovpl スクリプトを使用してデータベースレコードを 2 番目の NNMi 管理サーバーに配置する場合は，どちらの NNMi 管理サーバーも同じタイプのオペレーティングシステム，NNMi バージョンおよびパッチレベルである必要があります。

重要

クラスタ構成の NNMi で取得したバックアップデータをシングル構成の NNMi にリストアしないでください。

グローバルネットワーク管理機能を使用する場合は，バックアップデータを 2 番目の NNMi 管理サーバーに配置することは，どちらのサーバーのデータベース UUID も同じであることを意味します。2 番目の NNMi 管理サーバーに NNMi をリストアしたら，元の NNMi 管理サーバーから NNMi をアンインストールします。

- オンラインバックアップをリストアするため，NNMi は，ファイルシステムデータを正しい場所にコピーし，バックアップのデータベーステーブルの内容を上書きします。バックアップ以後に削除されたオブジェクトはリストアされます。バックアップ以後に作成されたオブジェクトは削除されます。また，バックアップの実行後に変更されたすべてのオブジェクトは，バックアップ時の状態に戻されます。NNMi データベースの場合は，nmsdbmgr サービスが実行されている必要があります。
- オフラインバックアップをリストアするため，NNMi は，ファイルシステム内の Postgres ファイルを上書きし，データベースファイルをバックアップデータで完全に置き換えます。

-force オプションを指定すると，nnmrestore.ovpl コマンドはすべての NNMi プロセスを停止し，nmsdbmgr サービスを開始し（NNMi データベースのオンラインバックアップからのリストアの場合），データをリストアし，その後すべての NNMi プロセスを再開始します。

指定されたソースが tar ファイルの場合は，NNMi リストアコマンドで，現在の作業ディレクトリの一時フォルダに tar ファイルが抽出されます。この場合，現在の作業ディレクトリに十分な空き容量があることを確認するか，リストアコマンドを実行する前にアーカイブを抽出してください。

メモ

NNMi のあるバージョンから次のバージョンへデータベースのスキーマが変わるおそれがあるため，データバックアップを NNMi の異なるバージョン間で共有することはできません。

20.3.1　同じシステムでのリストア

1 つのシステムでバックアップコマンドとリストアコマンドを使用することで，データを復旧できます。バックアップの実行時からリストアの実行時までの間に，次の項目が変更されていないようにする必要があります。

- NNMi のバージョン（パッチを含む）
- オペレーティングシステムタイプ
- キャラクタセット（言語）
- ホスト名
- ドメイン

20.3.2　異なるシステムでのリストア

バックアップコマンドとリストアコマンドを使用して，NNMi 管理サーバーからほかの管理サーバーへデータを転送できます。異なるシステムでのリストアは，システム障害時の復旧や，オペレーティングシステムのバージョンアップで NNMi の異なるシステムへの転送などに使用します。

ヒント

グローバルネットワーク管理機能を使用する場合は，NNMi UUID がデータベースのリストア中にターゲットシステムにコピーされるため，ソースとターゲットの両システムが NNMi の同じインスタンスを実行するおそれがあります。ソースシステムから NNMi をアンインストールしてください。

メモ

グローバルネットワーク管理を導入する間など，同様の設定で機能する NNMi 管理サーバーを複数作成する場合，`nnmconfigexport.ovpl` コマンド，および`nnmconfigimport.ovpl` コマンドを使用します。

異なるシステムでのリストアは，両方のシステムで次の項目が同じである必要があります。

- NNMi のバージョン（パッチを含む）
- オペレーティングシステムタイプ
- キャラクタセット（言語）

次の項目は，2 つのシステム間で異なっていてもかまいません。

- ホスト名
- ドメイン

異なるシステムでのリストアの場合，nnmrestore.ovpl コマンドはライセンス情報を新規システムにコピーしません。新しい NNMi 管理サーバーの新規ライセンスを取得して適用してください。詳細については，ライセンスのマニュアルを参照してください。

20.4　バックアップとリストアの方針

20.4.1　すべてのデータを定期的にバックアップする

ディザスタリカバリ計画には，すべての NNMi データの完全バックアップを定期的に実行するスケジュールを含めてください。このバックアップを作成するために NNMi を停止する必要はありません。バックアップをスクリプトに組み込む場合は，-force オプションを使用して，バックアップが開始される前に NNMi が正しい状態になるようにしてください。

（例）

```
nnmbackup.ovpl -force -type online -scope all -archive -target nnmi_backups¥periodic
```

ハードウェアの障害のために NNMi データの復旧が必要になった場合は，次の手順を実行します。

1. ハードウェアを再構成するか，新規ハードウェアを取得する。
2. バックアップデータの場合と同じバージョンおよびパッチレベルの NNMi をインストールする。
3. NNMi データをリストアする。
   - リカバリ NNMi 管理サーバーが「20.3.1　同じシステムでのリストア」にある要件を満たす場合は，次の例のようなコマンドを実行します。

   ```
   nnmrestore.ovpl -force -lic -source nnmi_backups¥periodic¥newest_backup
   ```

   - リカバリ NNMi 管理サーバーが同じシステムでのリストアを行うのに適格ではなくても，「20.3.2　異なるシステムでのリストア」の一覧にある要件を満たす場合は，次の例のようなコマンドを実行します。

   ```
   nnmrestore.ovpl -force -source nnmi_backups¥periodic¥newest_backup
   ```

   必要に応じてライセンスを更新します。

20.4.2　設定変更前のデータをバックアップする

設定変更を開始する前に，領域を限定したバックアップを必要に応じて実施してください。バックアップの領域については，「20.2.2　バックアップ領域」を参照してください。領域を限定したバックアップをすると，設定を変更しても期待した効果が見られない場合，周知の作動設定に戻すことが可能になります。

（例）

```
nnmbackup.ovpl -type online -scope config -target nnmi_backups¥config
```

このバックアップを同じ NNMi 管理サーバーにリストアするには，すべての NNMi プロセスを停止してから，次の例のようなコマンドを実行します。

```
nnmrestore.ovpl -force -source nnmi_backups¥config¥newest_backup
```

20.4.3　NNMi またはオペレーティングシステムのバージョンアップ前のデータをバックアップする

大規模なシステム変更（NNMi またはオペレーティングシステムのアップグレードを含む）を行う前に，すべての NNMi データの完全バックアップを実行します。バックアップの実行後 NNMi データベースが変更されないようにするため，すべての NNMi プロセスを停止し，オフラインバックアップを作成してください。

（例）

```
nnmbackup.ovpl -type offline -scope all -target nnmi_backups¥offline
```

システムの変更後に NNMi が正常に実行されなくなった場合は，変更をロールバックするか，または異なる NNMi 管理サーバーをセットアップし，「20.3.2　異なるシステムでのリストア」の一覧にある要件が確実に満たされるようにしてください。その後，次の例のようなコマンドを実行します。

（例）

```
nnmrestore.ovpl -source nnmi_backups¥offline¥newest_backup
```

必要に応じてライセンスを更新します。

20.4.4　ファイルシステムのファイルだけをリストアする

データベーステーブルに影響を与えることなく NNMi ファイルを上書きするには，次の例のようなコマンドを実行します。

（例）

```
nnmrestore.ovpl -partial -source nnmi_backups¥offline¥newest_backup
```

20.5 データベースをバックアップおよびリストアする

NNMi では，nnmbackupembdb.ovpl コマンドとnnmrestoreembdb.ovpl コマンドによって，NNMi データベースだけをバックアップおよびリストアします。この機能は，NNMi の設定でデータのスナップショットを作成する場合に便利です。

nnmbackupembdb.ovpl コマンドは，オンラインバックアップだけを実行します。最低でも，nmsdbmgr サービスが実行されている必要があります。

各バックアップ操作によって，ターゲットディレクトリ内のnnm-bak-<TIMESTAMP>という名前の親ディレクトリにファイルが格納されます。-noTimeStamp オプションを指定すると，ディスクスペースを節約できます。-noTimeStamp オプションを使用した場合，親ディレクトリの名前はnnm-bak になります。前回のバックアップ後に-noTimeStamp オプションを使用してバックアップが行われると，前回のバックアップの名前がnnm-bak.previous に変更されて，ローリングバックアップが作成されます。この名前変更は，2 回目のバックアップの完了後，バックアップデータの喪失を防止するために行われます。

ヒント

nnmresetembdb.ovpl コマンドは，データベースにデータをリストアする前に実行してください。このコマンドによってデータベースにエラーが含まれないようになるため，データベース制約違反が発生するおそれがなくなります。データベースリセットコマンドの実行については，nnmresetembdb.ovpl のリファレンスページを参照してください。

21 NNMi の保守

NNMi 管理サーバーが機能するようになったら，複数の NNMi 機能を最適化するためにメンテナンス作業を実施できます。

21.1 NNMi フォルダのアクセス制御リストの管理

NNM Action Server を実行するユーザー名の変更が必要な場合があります。権限を変更しないでアクションサーバーを実行するユーザー名を変更すると，NNM Action Server が起動しなくなり，インシデントアクションの実行中に NNMi がメッセージを記録しなくなるおそれがあります。この発生を防ぐ方法について説明します。

NNMi には，次のフォルダを変更する権限が含まれています。

- /var/opt/OV/log/nnm/public
- /var/opt/OV/shared/perfSpi

NNMi の/var/opt/OV/log/nnm/public フォルダに対する既定の権限は 755 ですが，NNMi は ACL を使用して，データベースユーザー（nmsdbmgr）およびnnmaction ユーザー（bin）のアクセス権を調整します。NNMi のポストインストール（インストールまたはアップグレードスクリプトの一部）中に，インストールスクリプトによって/var/opt/OV/log/nnm/public フォルダの権限が変更され，ACL が追加されます。

インストールスクリプトが予期しないエラーによって/var/opt/OV/log/nnm/public フォルダに ACL を設定できない場合，スクリプトは/var/opt/OV/log/nnm/public フォルダをワールド（そのほかのユーザー）によって書き込み可能にし，NNMi インストールは正常に完了します。NNMi インストールの成功後，/var/opt/OV/log/nnm/public フォルダへのワールドによる書き込み権限を制限するには，NNMi 管理サーバーのオペレーティングシステムに ACL を設定するためのシステム管理者マニュアルを参照してください。

/var/opt/OV/log/nnm/public フォルダのユーザーアクセスを調整するには，Linux ACL（アクセス制御リスト）を使用します。ACL の設定は，owner/group/other の権限を拡張するのに役立ちます。ACL は，Linux でサポートされています。

例えば，次のコマンドの実行後，USER 変数で示されたユーザーは/var/opt/OV/log/nnm/public フォルダへの書き込み権限を取得します。次のコマンドを実行しない場合，/var/opt/OV/log/nnm/public フォルダの権限は 755 で，ルート以外のユーザーはディレクトリ内のファイルに書き込めません。

```
setfacl -m user:<USER>:rwx /var/opt/OV/log/nnm/public
```

setfacl コマンドの使用方法の詳細については，該当するリファレンスページを参照してください。

21.2　ノードグループの設定

NNMi には，ノードグループの設定を自動化できるコマンドラインツールが用意されています。nnmnodegroup.ovpl コマンドでは，ノードグループの作成，表示，変更，および削除ができます。

詳細については，nnmnodegroup.ovpl のリファレンスページを参照してください。

21.3 ノードグループマップ設定の構成

ノードグループマップの設定は，NNMi コンソールだけでなく，nnmnodegroupmapsettings.ovpl コマンドラインツールを使用して行うこともできます。nnmnodegroupmapsettings.ovpl ツールでは，ノードグループマップの設定を作成，変更，および削除できます。このツールを使用して，TXT，XML，または CSV 形式で現在のノードグループマップの設定を表示することもできます。

メモ

NNMi を現在実行している Web ブラウザをリフレッシュすると，ノードグループマップの設定に加えた変更がただちに反映されます。

詳細については，nnmnodegroupmapsettings.ovpl のリファレンスページを参照してください。

21.4 通信設定の構成

nnmcommunication.ovpl コマンドラインツールを使用して，NNMi 通信設定を行うことができます。nnmcommunication.ovpl ツールでは，通信設定を作成，表示，変更，削除できます。このツールでは，テキストテーブル，テキストリスト，または XML 形式でリストを生成できます。

管理者は，nnmcommunication.ovpl ツールを使用して，管理アドレスやコミュニティ文字列などのフィールドの SNMP エージェント設定をロックして直接管理することで，通常の設定をバイパスすることもできます。

nnmcommunication.ovpl ツールは，デフォルト，ノード固有，リージョン固有，および SNMP エージェント固有の各設定において，コマンドラインインタフェース（CLI）による SNMP プロキシポートや SNMP プロキシアドレスの作成，更新，および削除をサポートしています。

詳細については，nnmcommunication.ovpl のリファレンスページを参照してください。

21.5　カスタムポーラー収集エクスポートの管理

カスタムポーラー機能では，SNMP MIB 式を使用して NNMi がポーリングする必要のある追加情報を指定することによって，積極的にネットワーク管理を行えます。カスタムポーラー収集は，収集（ポーリング）する情報およびそれらの情報の NNMi による処理方法を定義します。詳細については，NNMi ヘルプの「カスタムポーラー収集を作成する」および「カスタムポーラー設定を作成する」を参照してください。

カスタムポーラー機能を使用する場合でも，処理が終わったファイルをエクスポートディレクトリから削除するのはユーザーの責任です。長期の保存にエクスポートファイルを使用しないでください。設定された最大ディスク容量を超えると，NNMi によって古いファイルが削除され，新しいファイルが作成されます。これらのファイルを別の場所に保存していないと，ファイルは失われます。

21.5.1　カスタムポーラー収集のエクスポートディレクトリを変更する

NNMi は，ユーザーがエクスポートした収集データを次のディレクトリに書き込みます。

- Windows：`%NNM_DATA%¥shared¥nnm¥databases¥custompoller¥export`
- Linux：`$NNM_DATA/shared/nnm/databases/custompoller/export`

NNMi がカスタムポーラーファイルを書き込むディレクトリを変更するには，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-custompoller.properties`
   - Linux：`$NNM_PROPS/nms-custompoller.properties`
2. `exportdir` エントリを特定する。

   このエントリは次の行のように記述されています。

   ```
   #!com.hp.nnm.custompoller.exportdir=<base directory to export custom poller metrics>
   ```

   NNMi がカスタムポーラー収集情報を`C:¥CustomPoller` ディレクトリに書き込むように設定するには，次のように行を変更します。

   ```
   com.hp.nnm.custompoller.exportdir=C:/CustomPoller
   ```

   行の始めにある#!文字を必ず削除してください。

   > **重要**
   >
   > Windows の場合も，ディレクトリの区切り文字には「¥」ではなく「/」を使用してください。

3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.5.2 カスタムポーラー収集のエクスポートに使用する最大ディスク容量を変更する

collection_name.csv ファイルにデータをエクスポートするときに NNMi が使用する最大ディスク容量を変更するには，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-custompoller.properties`
   - Linux：`$NNM_PROPS/nms-custompoller.properties`
2. `maxdiskspace` エントリを特定する。

   このエントリは次の行のように記述されています。

   ```
   #!com.hp.nnm.custompoller.maxdiskspace=1000
   ```

   各 collection_name.csv ファイルに最大 2,000MB（2GB）のストレージ容量を確保するように NNMi を設定するには，その行を次のように変更します。

   ```
   com.hp.nnm.custompoller.maxdiskspace=2000
   ```
3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

   ```
   ovstop
   ovstart
   ```

21.5.3 カスタムポーラーメトリックスの累積周期を変更する

NNMi は，データをファイルに書き込む前に，カスタムポーラー収集メトリックスを累積する期間を分単位で設定します。カスタムポーラーメトリックスの累積周期を変更するには，次の手順に従います。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-custompoller.properties`
   - Linux：`$NNM_PROPS/nms-custompoller.properties`
2. 次のような行を特定する。

   ```
   #!com.hp.nnm.custompoller.accumulationinterval=5
   ```

   デフォルト値である 5 分間ではなく 10 分間，メトリックスを収集するように NNMi を設定するには，その行を次のように変更します。

```
com.hp.nnm.custompoller.accumulationinterval=10
```

3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.6　インシデントアクションの管理

アクションは，インシデントライフサイクルの任意の時点で自動的に実行されるように設定できます。例えば，設定しているタイプのインシデントが生成されるときにあるアクションが発生するように設定します。詳細については，NNMi ヘルプの「インシデントのアクションを設定する」を参照してください。

アクションのパラメーターを調整するには，次のセクションに示す手順に従ってください。

21.6.1　同時アクション数を設定する

NNMi が実行できる同時アクション数を変更するには，次の手順に従います。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nnmaction.properties
   - Linux：$NNM_PROPS/nnmaction.properties
2. 次のような行を特定する。

   ```
   #!com.hp.ov.nms.events.action.numProcess=10
   ```

   デフォルト値ではなく，20 個の同時アクションを実行できるように NNMi を設定するには，その行を次のように変更します。

   ```
   com.hp.ov.nms.events.action.numProcess=20
   ```

   行の始めにある#!文字を必ず削除してください。
3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

   ```
   ovstop
   ovstart
   ```

21.6.2　Jython アクションのスレッド数を設定する

jython スクリプトを実行するためにアクションサーバーが使用するスレッド数を変更するには，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nnmaction.properties
   - Linux：$NNM_PROPS/nnmaction.properties
2. 次のような行を探す。

```
#!com.hp.ov.nms.events.action.numJythonThreads=10
```

デフォルトのスレッド数ではなく，20 個のスレッドでjython スクリプトを実行できるように NNMi を設定するには，その行を次のように変更します。

```
com.hp.ov.nms.events.action.numJythonThreads=20
```

行の始めにある#!文字を必ず削除してください。

3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.6.3　アクションサーバー名のパラメーターを設定する

Windows の NNMi 管理サーバーでアクションサーバーを実行するユーザー名を変更するには，NNM Action Server サービスのLogOn プロパティを変更します。管理者権限を持つユーザー名を指定してください。

Linux の NNMi 管理サーバーでアクションサーバーを実行するユーザー名を変更するには，次の手順を実行します。

1. 次のファイルを編集する。

```
$NNM_PROPS/nnmaction.properties
```

2. 次のような行を特定する。

```
#!com.hp.ov.nms.events.action.userName=bin
```

デフォルト値ではなく，システムがアクションサーバーを実行するように NNMi を設定するには，その行を次のように変更します。

```
com.hp.ov.nms.events.action.userName=system
```

行の始めにある#!文字を必ず削除してください。

3. 変更を保存する。
4. 次のコマンドを実行して，アクションサーバーを再起動する。

```
ovstop nnmaction
ovstart nnmaction
```

21.6.4　アクションサーバーのキューサイズを変更する

短期間に大量に発生するインシデントに長時間終了しないコマンドをインシデントアクションとして設定した場合，アクションサーバーは多くのメモリを使用するおそれがあります。アクションサーバーのパフォーマンスを上げるために，アクションサーバーで使用可能なメモリサイズが制限されています。

これらの制限を変更するには，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nnmaction.properties
   - Linux：$NNM_PROPS/nnmaction.properties
2. 次のような 2 行を探す。

```
com.hp.ov.nms.events.action.jvmargs.minMemsize=-Xms6m
com.hp.ov.nms.events.action.jvmargs.maxMemsize=-Xmx30m
```

3. 上記のパラメーターでは，最小メモリサイズが 6MB に，最大が 30MB に設定されていることがわかる。これらのパラメーターをニーズに合わせて調整する。
4. 変更を保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.6.5　インシデントアクションのログ

アクションが実行されると，実行結果がインシデントアクションのログファイルに記録されます。このログの内容を確認するためには，［ツール］＞［インシデントアクションログ］を実行します。このログファイルに記録される項目については，次の表を参照してください。

表 21-1　インシデントアクションログに記録される項目一覧

項目	説明
コマンド	インシデントが設定されたライフサイクル状態になったときに実行されるコマンド
インシデント名	インシデントの名前
インシデント UUID	インシデントの UUID（［登録］タブに表示）
コマンドのタイプ	コマンドのタイプ（Jython，または ScriptOrExecutable）
ライフサイクル状態	インシデントのライフサイクル状態（Registered，In Process，Completed，または Closed）
終了コード	コマンドの戻り値

項目	説明
標準出力	標準出力への出力内容
標準エラー	標準エラー出力への出力内容
実行ステータス	アクションの実行結果

21.7 server.properties ファイルの設定の上書き

システムには 2 つのserver.properties ファイルがある場合があります。

次のファイルは製品のインストーラーによって作成され，アプリケーションインスタンス用にアプリケーションサーバーをカスタマイズするプロパティが含まれています。このファイルはユーザーによる変更は不可能で，コードメンテナンス（アップグレードおよびパッチ）で置き換えられます。

- Windows：%NnmInstallDir%NNM¥server¥server.properties
- Linux：$NnmInstallDir/NNM/server/server.properties

次のファイルは，ユーザーによって独自の環境用にアプリケーションを設定するために使用され，製品によってアップグレードまたはパッチで変更されることはありません。このファイルは，その他のファイルで設定された値を上書きします。そのため，すべてのカスタマイズはこのファイルで実行されます。

- Windows：%NnmDataDir%nmsas¥NNM¥server.properties
- Linux：$NnmDataDir/nmsas/NNM/server.properties

21.7.1 ブラウザのロケール設定の上書き

次のserver.properties ファイルを使用して，ブラウザのロケール値に関係なく，指定されたロケール値をすべての NNMi クライアントに強制的に適用できます。

- Windows：%NnmDataDir%nmsas¥NNM¥server.properties
- Linux：$NnmDataDir/nmsas/NNM/server.properties

server.properties ファイルを使用してこの値が設定されている場合，ブラウザのロケール値は無視されます。

ブラウザのロケール設定を上書きするには，次の手順を実行します。

1. server.properties ファイルを開く。
   - Windows：%NnmDataDir%nmsas¥NNM¥server.properties
   - Linux：$NnmDataDir/nmsas/NNM/server.properties
2. nmsas.server.forceClientLocale に移動する。
3. nmsas.server.forceClientLocale を次のどちらかに設定する。
   - nmsas.server.forceClientLocale= *<2文字のISO言語コード>*

   例えば，ISO 言語コードだけを使用してロケールを英語に設定するには，次のように入力します。

   （例）

   ```
   nmsas.server.forceClientLocale = en
   ```

- nmsas.server.forceClientLocale= *<2文字のISO言語コード>_<2文字のISO国コード>*

例えば，ISO 言語コードと国コードを使用してロケールを英語に設定するには，次のように入力します。

（例）

```
nmsas.server.forceClientLocale = en_US
```

4. 次のコマンドを NNMi管理サーバーで実行して，NNMi ovjboss サービスを再起動する。

```
ovstop ovjboss
ovstart
```

server.properties ファイルへの変更は，ovjboss の起動時にだけ読み取られます。

詳細については，server.properties ファイル内のコメントを参照してください。

21.7.2 SNMP Set オブジェクトアクセス権限の設定

次のファイルを使用して，ユーザーがアクセスできるノードでSNMP Set機能を使用するために必要なオブジェクトアクセス権限を設定できます。

- Windows：%NnmDataDir%nmsas¥NNM¥server.properties
- Linux：$NnmDataDir/nmsas/NNM/server.properties

SNMP Set機能の詳細については，NNMiヘルプを参照してください。オブジェクトアクセス権限の詳細については，NNMiヘルプ「管理」を参照してください。

SNMP Set機能に対するオブジェクトアクセス権限を設定するには，次の手順を実行します。

1. server.properties ファイルを開く。
   - Windows：%NnmDataDir%nmsas¥NNM¥server.properties
   - Linux：$NnmDataDir/nmsas/NNM/server.properties
2. 次の行を追加する。

```
permission.override.com.hp.nnm.SNMP_SET=<オブジェクトアクセスロール>
```

<オブジェクトアクセスロール>で有効な値は次のとおりです。

```
com.hp.nnm.ADMIN
com.hp.nnm.LEVEL2
com.hp.nnm.LEVEL1
com.hp.nnm.GUEST
```

例えば，［オブジェクト管理者］および［オブジェクトオペレーターレベル 2］オブジェクトアクセス権限でSNMP Set機能を使用できるようにするには，次のように入力します。

（例）

```
permission.override.com.hp.nnm.SNMP_SET=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2
```

3. アクセスを有効にする各オブジェクトアクセス権限を含める。
4. 次のコマンドを NNMi 管理サーバーで実行して，NNMi ovjboss サービスを再起動する。

```
ovstop ovjboss
ovstart
```

server.properties ファイルへの変更は，ovjboss の起動時にだけ読み取られます。

21.8 SNMP トラップの管理

ここでは，タスクの実行方法について説明します。

21.8.1 SNMPv3 以外で管理されているノードまたは監視対象外のノードの SNMPv3 トラップを認証するための NNMi の設定

NNMi が次のいずれかの条件を満たしているノードから SNMPv3 トラップを受信している場合，このセクションの手順を実行します。

- デバイスが SNMPv2 または SNMPv1 を使用して管理されている。
- デバイスが非 SNMP ノードとして管理されている。
- デバイスが NNMi によって検出されていない。

これらのデバイスの SNMPv3 エンジン ID を SNMPv3 キャッシュに追加するように NNMi を設定できます。

このように NNMi を設定することで，NNMi はこれらの SNMPv3 トラップを認証して保存できます。

SNMPv3 以外で管理されているノードまたは検出されていないノードの SNMPv3 トラップを受信して保存するように NNMi を設定するには，次の手順を実行します。

1. NNMi コンソールで，［設定］＞［通信の設定］に移動する。
   各受信トラップにトラップの認証に使用するための対応する設定が適用されるように，**［領域］**または**［特定ノードの設定］**レベルのデフォルトのエントリを設定します。詳細については，NNMi ヘルプの「デフォルト SNMPv3 を設定する」を参照してください。

   メモ

   SNMPv3 ノードの含まれるアドレス範囲の領域を使用するか，それぞれに対して**［特定ノードの設定］**を設定することをお勧めします。

2. NNMi コンソールで，［設定］＞［インシデント］＞［インシデントの設定］に移動する。
3. NNMi が検出していないデバイスからの SNMPv3 トラップを受信する場合，［未解決の SNMP トラップおよび Syslog メッセージを破棄する］を選択解除する。

   メモ

   ［未解決の SNMP トラップおよび Syslog メッセージを破棄する］の選択を解除すると，NNMi は検出していないデバイスから送信された SNMP トラップを受信して保存するようになります。

4. NNMi 管理サーバーで ovstop コマンドを実行する。

5. 次のファイルを編集する。

- Windows：%NNM_PROPS%¥nms-communication.properties
- Linux：$NNM_PROPS/nms-communication.properties

6. ファイルの最下部に次の行を追加する。

```
com.hp.nnm.snmp.engineid.file=<ファイルへのパス>file.txt
```

<ファイルへのパス>file.txt エントリは，デバイスを含むファイルの完全なパスとファイル名です。

これらの設定の変更によって，NNMi は NNMi プロセスが再起動されるたびにこのファイルからのエントリを SNMPv3 キャッシュに読み込みます。

重要

Linux NNMi 管理サーバーでは，ファイルパスは/var/opt/OV/etc などの通常の形式です。

Windows NNMi 管理サーバーでは，区切り文字としてスラッシュを使用します。例えば，C:/temp/file.txt などの形式になります。

7. 変更を保存する。

8. <ファイルへのパス>file.txt ファイルを編集する。

a　デバイスの IP アドレス，ポート，およびエンジン ID の各項目をカンマで区切って追加します。

b　個別の行にデバイスごとに 1 つのエントリを追加します。

エンジンIDは一連の16進数バイトです。NNMiは大文字と小文字を区別しないで，スペースを認識します。

次の例を使用してエントリを作成します。

```
16.1.2.3,161,80 00 00 09 30 00 00 1f e9 a3 33 01
16.1.2.4,161,80 00 00 11 03 00 00 2d 51 99 30 00
1050:0000:0000:0000:0005:0600:300c:326b, 161, 800000090300001f9ea33000
ff06::c3,161,80 00 00 09 03 00 00 1f 9A A3 30 00
```

a　NNMi 管理サーバーでovstart コマンドを実行し，NNMi を起動して<ファイルへのパス>file.txt ファイルを読み込みます。

b　Boot.log ファイルで，NNMi がファイルを読み込んでいることを確認します。

このファイルに，ファイルが読み込まれたことを示す次のようなログメッセージが含まれていることを確認します。

```
2012-10-17 14:44:44.876 INFO  [NnmTrapService] Start: Populate engineIDs from file
2012-10-17 14:45:08.017 INFO  [SnmpV3EngineIdCachePopulator] Successfully loaded 3 V3
Engine IDs from file /temp/patch2/v3hosts.txt
```

ノードの有効な設定へのマッピングエラーが発生した場合は，次のようなメッセージが含まれています。

```
2012-10-17 14:45:03.485 WARNING  [SnmpV3EngineIdCachePopulator] V3
Engine IDs: Could not resolve SNMPv3 configuration for 16.1.2.6
```

上記のようなメッセージが含まれている場合は，このノードの［設定］＞［通信の設定］設定を調整します。

メモ

*<ファイルへのパス>*file.txt ファイルだけでなくキャッシュからもエントリを削除する必要がある場合，*<ファイルへのパス>*file.txt からエントリを削除してから，次のコマンドを実行して，NNMi を再起動することが最良の方法です。

```
ovstop

ovstart
```

21.8.2 SNMPv1 トラップまたは SNMPv2c トラップのブロック

SNMPv3 のみを使用するようにデバイス検出を設定したにもかかわらず，一部の管理対象ノードが，引き続き NNMi 管理サーバーに SNMPv1 トラップまたは SNMPv2c トラップの送信を試みる場合があります。SNMPv1 トラップまたは SNMPv2c トラップが NNMi 管理サーバーに到達しないように，SNMPv3 トラップのみを受け入れ，SNMPv1 トラップおよび SNMPv2c トラップをすべてブロックするように NNMi を設定することをお勧めします。

注：この設定手順を完了する前に，SNMPv3 プロトコルを使用するネットワークを検出するように NNMi が設定されていることを確認します。

1. NNMi 管理サーバーにログオンします。
2. 次のコマンドを実行します。
   - Windows の場合：
     ```
     %NnmInstallDir%bin¥nnmtrapconfig.ovpl -setProp disallowV1V2 -persist
     ```
   - Linux の場合：
     ```
     /opt/OV/bin/nnmtrapconfig.ovpl -setProp disallowV1V2 -persist
     ```
3. 次のいずれかを実行します。
   - Windows の場合：**［サービス］** ウィンドウから NNM TrapReceiver サービスを再起動します。
   - Linux の場合：次のコマンドを実行します。
     ```
     /opt/OV/bin/nettrap stop
     /opt/OV/bin/nettrap start
     ```

21.8.3 Causal Engine がトラップを受け入れる期間の設定

広範囲のネットワークが一定の予測できる時間に利用できなくなる場合，NNMi では Causal Engine へのトラップの配信を阻止することで，Causal Engine の分析負荷を抑制できます。トラップの配信を阻止するには，NNMi 管理者として，NNMi Causal Engine がイベントシステムからのトラップの受け入れを停止する期間を設定します。

重要

この機能は，NNMi コンソールに配信されるトラップには影響しません。

Causal Engine に配信されるトラップは，StatePoller をトリガーし，StatePoller のポーリングポリシーによって指示されたスケジュールよりも早くノードをポーリングする場合に使用されます。トラップの配信を阻止する場合，NNMi は StatePoller から更新情報を取得する前に，スケジュールされたポーリング間隔まで待機する必要があります。あらゆる場合に，NNMi Causal Engine は NNMi StatePoller からのステートフローを使用して，トラップがあるかないかにかかわらず同じ結論に達します。

Causal Engine がトラップの受け入れを停止する期間を設定するには，次の手順を実行します。

1. 次のファイルを作成する。
   - Windows：`%NNM_PROPS%¥nms-apa.properties`
   - Linux：`$NNM_PROPS/nms-apa.properties`
2. ファイルに次の内容を追加する。

   ```
   PROPERTY NAME: com.hp.ov.nms.apa.trapGateSchedule
   ```

   次の例をガイドラインとして使用します。

   次の例では，トラップは深夜に流れ，午前 8:30 に阻止され，午前 10:00 に再度流れてから，午後 4:30 に再度阻止されます。

   ```
   com.hp.ov.nms.apa.trapGateSchedule = ENABLE_APA_TRAPS 08:30 10:00 16:30
   ```

   次の例では，トラップは深夜に阻止され，午前 8:30 に再度流れ，午前 10:00 に阻止されてから，午後 4:30 に再度流れます。

   ```
   com.hp.ov.nms.apa.trapGateSchedule = DISABLE_APA_TRAPS 08:30 10:00 16:30
   ```
3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

   ```
   ovstop
   ovstart
   ```

21.9 trapFilter.conf ファイルでインシデントをブロックする

NNMi 管理サーバー上のインシデントの数が一定のレートに達して，新しく到着するインシデントを NNMi がブロックする場合，次の点に注意してください。

- NNMi は TrapStorm インシデントを生成し，インシデントがブロックされていることを示します。
- NNMi は主要なヘルスメッセージも生成し，インシデントレートが高くてインシデントがブロックされていることを示すことがあります。

インシデント数を削減するには，次のどちらかの方法を使用します。

- nnmtrapd.conf ファイルを使用し，インシデントが NNMi に入るのをブロックしてインシデントトラフィックの削減を試みます。

  重要

  nnmtrapd.conf ファイルによる方法を使用すると，NNMi は引き続きこれらのインシデントを使用してトラップレートを計算し，トラップバイナリストアに書き込みます。nnmtrapd.conf ファイルによる方法を使用しても，インシデントがデータベースで作成されたり保存されたりすることを停止することしかできません。

  詳細については，nnmtrapd.conf のリファレンスページを参照してください。

- trapFilter.confファイルを使用し，NNMiイベントパイプラインで早期にインシデントをブロックして，このインシデントがトラップレート計算で分析されること，または NNMiトラップバイナリストアに保存されることを回避します。

  メモ

  デバイスの IP アドレスまたは OID をtrapFilter.conf ファイルに追加すると，この大量のインシデントをブロックして，インシデントのボリュームの問題を回避できます。

  詳細については，trapFilter.conf およびnnmtrapconfig.ovpl のリファレンスページを参照してください。

21.10 NNMi の文字セットエンコードの設定

NNMi 管理サーバーに設定したロケールに応じて，NNMi で SNMP OCTETSTRING データの解釈に使用するソースエンコードの設定が必要な場合があります。これを行うには，nms-jboss.properties ファイルを次のように編集します。

1. 環境に応じて次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
2. 次の行を含むテキストブロックを探す。

```
#!com.hp.nnm.sourceEncoding=UTF-8
```

3. この行のコメントを解除する。

```
com.hp.nnm.sourceEncoding=UTF-8
```

4. nms-jboss.properties ファイルに記述されているコメント文の例に従って，手順 3.で示されたプロパティ値（UTF-8）を変更する。
5. nms-jboss.properties ファイルを保存する。
6. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.11 MIB ブラウザパラメータの変更

NNMi MIB ブラウザ（[アクション] ＞ [MIB 情報] ＞ [MIB を参照] メニュー）を使用して，ノードの情報を取得し， SNMP コミュニティ文字列（省略可能）をそのノードに指定する場合は，NNMi MIB ブラウザは，MIB ブラウザ SNMP 通信用の`nms-ui.properties` ファイルにある MIB ブラウザパラメータを使用します。

重要

MIB ブラウザを使用するときにコミュニティ文字列を使用しない場合は，NNMi ではノードで確立されている [通信の設定] 設定（ある場合）を使用します。これらの設定は，[設定] ワークスペースの [通信の設定] ビューを使用して NNMi コンソールで設定されます。詳細については，NNMi ヘルプの「通信プロトコルを設定する」を参照してください。

1. 環境に応じて次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-ui.properties`
   - Linux：`$NNM_PROPS/nms-ui.properties`
2. 次の行を含むテキストブロックを探す。

```
# MIB Browser Parameters
```

3. 次のテキストを含む行を検索し，`# MIB Browser Parameters` の下にある MIB ブラウザパラメータを探す。

```
mibbrowser
```

4. `nms-ui.properties` ファイル内の手順に従って，MIB ブラウザパラメータを変更する。
5. `nms-ui.properties` ファイルを保存する。
6. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.12 レベル 2 オペレータがノードおよびインシデントを削除できるように構成する

デフォルトの NNMi では， NNMi は NNMi 管理者に対して NNMi でのノードまたはインシデントの作成，編集，削除を許可します。 NNMi オペレーターレベル 2（L2）ユーザーグループに割り当てられたアカウントに対しても，ノードまたはインシデントの削除を許可するように設定できます。この設定は次のいずれかの方法で実行できます。

- （推奨）必要なノードまたはインシデントを削除するため L2 ユーザーの必要な権限を引き上げる。この設定は， NNMi Web コンソールを使用して行うことができます。詳細については，NNMi 管理者ヘルプを参照してください。
- L2 ユーザーが全体的にノードまたはインシデントを削除できるように NNMi を設定する。この設定は，一定の NNMi プロパティファイルを変更してデフォルト権限を上書きすることで行うことができます。

重要

上書きによる方法は，全体的に許可する場合だけに使用してください。一度許可すると，NNMi Web コンソールで L2 ユーザーアクセス権限を制御できなくなります。

L2 ユーザーがノード，ノードに関連するインシデント，またはこの両方を編集または削除できるようにするには，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-topology.properties`
   - Linux：`$NNM_PROPS/nms-topology.properties`
2. 必要に応じて次の行を追加します。
   - L2 ユーザーがノードを削除できるようにするには，次の行を追加します。

     ```
     permission.override.com.hp.nnm.DELETE_OBJECT=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2
     ```

   - L2 ユーザーがインシデントを削除できるようにするには，次の行を追加します。

     ```
     permission.override.com.hp.nnm.incident.DELETE=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2
     ```

3. ファイルを保存します。
4. 次のコマンドを実行して， NNMi を再起動する。

   ```
   ovstop
   ovstart
   ```

HA 下でファイルに変更を加えるときには，クラスタの両方のノードで変更を加える必要があります。 HA 構成を使用している NNMi の場合， NNMi 管理サーバーの停止と再起動が必要な変更を加えたときには，`ovstop` および`ovstart` コマンドを実行する前に，ノードをメンテナンスモードにする必要があります。

21.13　レベル 2 オペレータがマップを編集できるように構成する

デフォルトの NNMi では，NNMi 管理者は，ノードグループの作成，変更，および削除によって，マップを編集できます。NNMi レベル 2 オペレータのユーザーグループに割り当てられたアカウントを構成して，この編集を可能にすることもできます。

NNMi を変更して，NNMi レベル 2 オペレータのユーザーグループに割り当てられたユーザーアカウントが，アクセス権を持つノード上のノードグループを作成，変更，および削除する必要がある場合は，次のようにします。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-ui.properties`
   - Linux：`$NNM_PROPS/nms-ui.properties`
2. 次のテキストブロックを探し，コメントを解除する。

```
#!com.hp.nnm.ui.level2MapEditing = true
```

3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

HA 下でファイルに変更を加えるときには，クラスタの両方のノードで変更を加える必要があります。HA 構成を使用している NNMi では，NNMi 管理サーバーの停止と再起動が必要な変更を加えた場合には，`ovstop` および`ovstart` コマンドを実行する前に，ノードをメンテナンスモードにする必要があります。

手順 1.から手順 5.までを行うと，NNMi コンソールは次のように変化します。

- **［インベントリ］** ＞ **［ノードグループ］** メニューに，NNMi レベル 2 オペレータの **［新規作成］** および **［削除］** ツールバーアイコンが表示される。
- ノードグループフォームのツールバーに **［保存して新規作成］** および **［ノードグループを削除］** ボタンが含まれる。
- **［すべてのノードグループ］** フォルダが **［トポロジマップ］** ワークスペースに表示される。詳細については，NNMi オンラインヘルプの「ワークスペースについて」を参照してください。
- ノードグループマップの場合，NNMi コンソールに **［マップの保存］** ツールバーボタンと，**［ファイル］** ＞ **［マップの保存］** メニュー項目が含まれます。
- **［レイアウトの保存］** 動作は，ノードグループマップにノードグループマップの設定が存在するかどうかによって異なります。ノードグループマップにノードグループマップの設定が存在しない場合は，作成する必要があります。

NNMi レベル 2 オペレータのユーザーにノードグループマップ設定の作成権限を付与するように NNMi を設定することもできます。

1. NNMi コンソールから，［トポロジマップ］＞［ノードグループの概要］を開く。
2. 関心のある［ノードグループ］アイコンをダブルクリックする。
   NNMi は，選択したノードグループに関連づけられたノードグループマップを開きます。
3. 次の手順を実行して，変更するノードグループマップの設定を開く。
   ［ファイル］＞［ノードグループマップの設定を開く］を選択します。
4. ［マップの保存のための最小 NNMi ロール］を［オペレーターレベル 2］に設定する。
5. 変更を保存する。

これで，NNMi レベル 2 オペレータは，ノードグループマップビューからノードグループマップの設定，編集，および削除ができます。

21.14 レベル 1 オペレータがステータスのポーリングおよび設定のポーリングを実行できるように構成する

NNMi では，NNMi レベル 2 オペレータのユーザーグループに割り当てられたユーザーアカウントは，アクセス権があるノードに対してステータスのポーリングと設定のポーリングを実行できます。それぞれの nms-topology.properties ファイルでオブジェクトアクセス権限レベルを変更するだけでなく，NNMi コンソールで［メニュー項目］設定も変更する必要があります。

NNMi を変更して，NNMi レベル 1 オペレータのユーザーグループに割り当てられたユーザーアカウントがステータスのポーリングおよび設定ポーリングを実行する場合は，次のようにします。

1. ［設定］＞［ユーザーインタフェース］＞［メニュー項目］＞［ステータスのポーリング］フォームを開く。
2. ［メニュー項目コンテキスト］タブから，変更しなければならない［必要な NNMi ロール/オブジェクトのタイプ］項目の各エントリを開く。
3. レベル 1 オペレータにステータスのポーリングを実行させたい各オブジェクトタイプについて，［必要な NNMi ロール］の値を［オペレータレベル 1］に変更する。

   この手順によって，NNMi レベル 1 オペレータユーザーグループに割り当てられたユーザーアカウントは，指定されたオブジェクトタイプのステータスのポーリングアクションを表示できるようになります。

NNMi レベル 1 オペレータのユーザーグループに割り当てられたユーザーアカウントに［設定のポーリング］メニュー項目の表示を許可するように NNMi を変更するには，次のようにします。

1. ［設定］＞［ユーザーインタフェース］＞［メニュー項目］＞［設定のポーリング］フォームを開く。
2. ［メニュー項目コンテキスト］タブから，変更しなければならない［必要な NNMi ロール/オブジェクトのタイプ］項目の各エントリを開く。
3. レベル 1 オペレータに設定のポーリングを実行させたい各オブジェクトタイプについて，［必要な NNMi ロール］の値を［オペレータレベル 1］に変更する。

   この手順によって，NNMi レベル 1 オペレータのユーザーグループに割り当てられたユーザーアカウントは，指定されたオブジェクトタイプの設定のポーリングアクションを表示できるようになります。

   次に，nms-topology.properties ファイルを手順 7.から手順 10.に示されているように編集して，NNMi レベル 1 オペレータのユーザーグループに割り当てられたユーザーアカウントが，NNMi コンソールからステータスのポーリングと設定のポーリングの両方のコマンドを実行できるようにします。これらのステップを完了しなかった場合，NNMi はアクションメニューにステータスのポーリングおよび設定のポーリングオプションを表示しますが，ユーザーがステータスのポーリングまたは設定のポーリングコマンドを実行しようとすると，エラーメッセージが表示されます。
4. ステータスのポーリングと設定のポーリングに必要なアクセスレベル（必要なオブジェクトアクセス権限レベル）を変更するには，次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-topology.properties

- Linux：$NNM_PROPS/nms-topology.properties

5. ファイルの最後までスクロールして，ステータスのポーリング変更のために次の行を追加する。

```
permission.override.com.hp.nnm.STATUS_POLL=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2,com.hp.nnm.
LEVEL1
```

6. 設定のポーリング変更のために次の行を追加する。

```
permission.override.com.hp.nnm.CONFIG_POLL=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2,com.hp.nnm.
LEVEL1
```

7. 変更を保存する。

8. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

HA 下でファイルに変更を加えるときには，クラスタの両方のノードで変更を加える必要があります。HA 構成を使用している NNMi の場合，NNMi 管理サーバーの停止と再起動が必要な変更を加えた場合には，ovstop およびovstart コマンドを実行する前に，ノードをメンテナンスモードにする必要があります。

21.15 プロキシ SNMP ゲートウェイによって送信されたトラップから元のトラップアドレスを判別する

NNMi のデフォルト設定を使用している場合，プロキシ SNMP ゲートウェイによって送信されたトラップには元のトラップアドレスが表示されない可能性があります。管理者は，元のトラップアドレスを判別するように NNMi を設定できます。

次の点に注意してください。

- NNMi にはカスタムインシデント属性cia.originaladdress が含まれます。NNMi は com.hp.nnm.trapd.useUdpHeaderIpAddress プロパティと併せてcia.originaladdress 属性の意味を判別します。
- com.hp.nnm.trapd.useUdpHeaderIpAddress パラメーターの値はデフォルトでfalse であるため，NNMi は通常cia.originaladdress 属性を無視します。
- com.hp.nnm.trapd.useUdpHeaderIpAddress 値をtrue に設定すると，cia.originaladdress 属性によって SNMP エージェントアドレスの値が提供されます。

NNMi でソースとして UDP ヘッダーアドレスを使用する一方で，管理対象デバイスの実際の SNMP アドレスへのアクセスが必要な場合，com.hp.nnm.trapd.useUdpHeaderIpAddress 値をtrue に設定すると便利です。

重要

com.hp.nnm.trapd.useUdpHeaderIpAddress 属性がfalse（デフォルト設定）の場合，cia.originaladdress とcia.address の両方の属性には同じ値が含まれます。

cia.originaladdress の値を使用して元のトラップアドレスを判別するように NNMi を設定するには，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
2. 次の行を含むテキストブロックを探す。

```
#!com.hp.nnm.trapd.useUdpHeaderIpAddress=false
```

3. この行をコメント解除し，次のように編集する。

```
com.hp.nnm.trapd.useUdpHeaderIpAddress=true
```

4. 変更を保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

重要

高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，ovstop およびovstart コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

NNMi はcia.originaladdress の値を使用して元のトラップアドレスを判別します。

21.15.1　トラップアドレスの順序

NNMi は，ソースアドレスを次のように分析します。

- com.hp.nnm.trapd.useUdpHeaderIpAddress プロパティがtrue に設定された SNMPv1 および SNMPv2c トラップは，次のアドレス順序を使用する。

```
rfc3584TrapAddress (.1.3.6.1.6.3.18.1.3.0)
nnmTrapForwardingAddress (.1.3.6.1.4.1.11.2.17.2.19.1.1.3.0)
securityPackNotificationAddress (.1.3.6.1.4.1.99.12.45.2.1.0)
proxyOid (.1.3.6.1.4.1.11.2.17.5.1.0)
IPヘッダーのソースアドレス
```

- com.hp.nnm.trapd.useUdpHeaderIpAddress プロパティがfalse に設定された SNMPv1 トラップは，次のアドレス順序を使用する。

```
rfc3584TrapAddress (.1.3.6.1.6.3.18.1.3.0)
nnmTrapForwardingAddress (.1.3.6.1.4.1.11.2.17.2.19.1.1.3.0)
securityPackNotificationAddress (.1.3.6.1.4.1.99.12.45.2.1.0)
proxyOid (.1.3.6.1.4.1.11.2.17.5.1.0)
v1トラップのagent-addrフィールド
IPヘッダーのソースアドレス
```

21.16 NNMi NmsTrapReceiver プロセス

NNMi には，フェイルオーバー時に SNMP トラップの損失を最小限に抑えるのに役立つスタンドアロン NmsTrapReceiver プロセスが備えられています。NmsTrapReceiver は，アクティブノードとスタンバイノードの両方で実行されます。

21.16.1 NmsTrapReceiver の設定

NNMi には，ユーザーが構成できる次の設定があります。

- trapReceiverJmsTTL

  trapReceiverJmsTTL オプションは，TrapReceiver でトラップをキャッシュする最大時間を設定します。デフォルト設定は 5 分です。jboss のダウン時間がこの時間を超えると，データが失われます。

  メモ

  この設定を行う前に，フェイルオーバーの所要時間を計ってベンチマークを判断してから，trapReceiverJmsTTL をその時間の 2 倍に設定します。

このような設定の変更方法については，nnmtrapconfig.ovpl のリファレンスページを参照してください。

重要

正しく動作するには，アクティブノードとスタンバイノードの間でクロックが同期されていることが重要です。同期されていないと，トラップの大量の重複または損失が生じる可能性があります。

詳細については，nnmtrapconfig.ovpl のリファレンスページを参照してください。

21.16.2 NmsTrapReceiver プロセスの開始と停止

NmsTrapReceiver プロセスは，オペレーティングシステム（Linux の場合：systemd のnettrap.service，Windows の場合：HP NNM NmsTrapReceiver サービス）によって自動的に開始されます。また，ovstart で NmsTrapReceiver プロセスが実行されていないことが検出された場合も，ovstart によって開始されます。

NmsTrapReceiver を手動で開始または停止する必要がある場合は，オペレーティングシステムのサービスを使用します。

重要

ovstart およびovstop コマンドは，リモートトラップサーバーではなく，トラップ処理の jboss パイプラインを開始および停止するだけです。

21.17 NNMi コンソールに HTTPS だけで接続する

NNMi コンソールへの HTTP アクセスを防止する最も効果的な方法は，保護されたシステムへの HTTPS アクセスだけを許可するファイアウォールの後ろに NNMi 管理サーバーを配置することです。

HTTP アクセスを防止するファイアウォール設定によって，Web サービスを使用して NNMi と通信し，HTTP だけをサポートする統合で問題が発生することがあります。統合製品のマニュアルを参照し，HTTPS をサポートしているかどうかを確認します。

より安全性に劣る方法では，次の手順によって，HTTP ポートからの NNMi コンソールアクセスリクエストを HTTPS ポートにリダイレクトします。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-ui.properties`
   - Linux：`$NNM_PROPS/nms-ui.properties`
2. 文字列`https` を検索し，次の行が含まれるテキストブロックを探す。

```
#! com.hp.ov.nms.ui.https.only=false
```

3. 次の行のコメントを解除し，次のように編集する。

```
com.hp.ov.nms.ui.https.only=true
```

4. 変更を保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

メモ

このプロパティを設定して HTTP 要求を NNMi コンソールの HTTPS にリダイレクトすると，NNMi にクロス起動するアプリケーションに問題が発生することがあります。このような問題が発生する場合は，この HTTPS リダイレクトを無効にします。

21.18　リモートアクセスには暗号化を必須とするように NNMi を設定する

NNMi をインストールして HTTPS 通信を使用するように設定したあとでも，通信の HTTP モードを引き続き使用できます。HTTP を経由した NNMi へのリモートアクセスを制限できるようにするには，次の手順を実行してください。

暗号化リモートアクセスだけを許可するように NNMi を設定する前に，グローバルネットワーク管理およびそのほかの連携製品が SSL をサポートしていることを確認します。暗号化リモートアクセスだけを許可するように NNMi を設定する前に，グローバルネットワーク管理およびそのほかの連携製品の SSL を設定してください。

アプリケーションフェイルオーバークラスターを設定したいと検討中で，まだ設定していない場合は，このタスクを実行しないでください。NNMi アプリケーションフェイルオーバークラスターをセットアップしたあと，次の手順を実行して，HTTP およびその他の非暗号化アクセスを無効にできます。

ネットワークから NNMi への HTTP アクセスを無効にするには，`server.properties` ファイルを次のように編集します。

1. 次のファイルを編集する。ファイルが存在しない場合は作成する。
   - Windows：`%NnmDataDir%nmsas¥NNM¥server.properties`
   - Linux：`$NnmDataDir/nmsas/NNM/server.properties`
2. `server.properties` ファイルに次の 4 行を追加する。

```
nmsas.server.net.bind.address = 127.0.0.1
nmsas.server.net.bind.address.ssl = 0.0.0.0
nmsas.server.net.hostname = localhost
nmsas.server.net.hostname.ssl = ${com.hp.ov.nms.fqdn}
```

3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.19 以前にサポートされていた varbind 順序を保持するように NNMi を構成する

すべての SNMPv2 トラップは，最初と 2 番目のvarbind としてsysUpTime.0 およびsnmpTrapOID.0 OID を含みます。

重要

SNMPv2 トラップ定義がsysUpTime.0 またはsnmpTrapOID.0 をトラップパラメーターとして含む場合，varbind リストの最初と 2 番目以外の位置に追加のvarbind として現れる可能性があります。

NNMi 10-10 より前では，NNMi はsysUpTime.0 およびsnmpTrapOID.0 OID のすべてのインスタンスをvarbind リストから削除していました。NNMi 10-10 からは，NNMi は，これらの OID がトラップ定義の一部であるときにこれらの OID を保持し，受信したトラップのvarbind リストの最初と 2 番目以外の位置にある可能性があります。この変更によって，sysUpTime.0 またはsnmpTrapOID.0 OID をトラップパラメーターとして持つトラップのvarbind 順序が変更されることがあります。

次の例では，1 番目のボールドのvarbind にsnmpTrapOID.0 の値が含まれ，2 番目のボールドのvarbind にsysUpTime.0 の値が含まれています。この例に示されているように，これらのvarbind はvarbind リストの 1 番目と 2 番目以外の位置に追加varbind として表示されます。

```
//0: SNMP MESSAGE (0x30): 115 bytes
//2: INTEGER VERSION (0x2) 1 bytes: 1 (SNMPv2C)
//5: OCTET-STR COMMUNITY (0x4) 6 bytes: ”public”
//13: V2-TRAP-PDU (0xa7): 102 bytes
//15: INTEGER REQUEST-ID (0x2) 2 bytes: 18079
//19: INTEGER ERROR-STATUS (0x2) 1 bytes: noError(0)
//22: INTEGER ERROR-INDEX (0x2) 1 bytes: 0
//25: SEQUENCE VARBIND-LIST (0x30): 90 bytes
//27: SEQUENCE VARBIND (0x30): 13 bytes
//29: OBJ-ID (0x6) 8 bytes: .1.3.6.1.2.1.1.3.0
//39: TIMETICKS (0x43) 1 bytes: 9
//42: SEQUENCE VARBIND (0x30): 32 bytes
//44: OBJ-ID (0x6) 10 bytes: .1.3.6.1.6.3.1.1.4.1.0
//56: OBJ-ID (0x6) 18 bytes: .1.3.6.1.6.3.1.1.5.3.1.3.6.1.4.1.9.1.14
//76: SEQUENCE VARBIND (0x30): 14 bytes
//78: OBJ-ID (0x6) 9 bytes: .1.3.6.1.2.1.2.2.1.1
//89: INTEGER (0x2) 1 bytes: 92
//92: SEQUENCE VARBIND (0x30): 23 bytes
//94: OBJ-ID (0x6) 10 bytes: .1.3.6.1.6.3.1.1.4.3.0
//106: OBJ-ID (0x6) 9 bytes: .1.3.6.1.4.1.11.2.3.14
```

メモ

NNMi でsysUpTime.0 OID とsnmpTrapOID.0 OID のすべてのインスタンスをvarbind リストから削除する場合にだけ，com.hp.nnm.events.preserveOldVarbindListOrder プロパティをtrue に設定します。

NNMi 10-10 より前と同じ動作にしたい場合は，次のようにします。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
2. 次の行を追加する。

```
com.hp.nnm.events.preserveOldvarbindListOrder=true
```

3. 変更を保存する。
4. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.20 古い SNMP トラップインシデントを自動でトリムする

NNMi のパフォーマンスを高いレベルで維持するために，データベースに一定数の SNMP トラップインシデントが存在すると，それ以上の SNMP トラップ（syslog メッセージを含む）はインシデント化されません。しかし，SNMP トラップインシデントの自動トリム機能を使って，データベースに保存されている SNMP トラップインシデントの数を調整し，受信した SNMP トラップを継続してインシデント化できます。

重要

NNMi 12-00 までは，NNMi は根本原因ではない SNMP トラップインシデントだけをトリムしました。

NNMi 12-10 以降は，「すべてのインシデント」または「 SNMP トラップインシデントだけ」のいずれかを設定できます。デフォルトは「すべてのインシデント」です。

NNMi 12-10 以降，SNMP トラップインシデントの自動トリム機能は新規インストール時に有効化されており，NNMi は古いインシデントをデータベースから削除します。（デフォルトではアーカイブは作成されません。）バージョンアップの場合は以前の設定が引き継がれます。

`com.hp.nnm.events.snmpTrapAutoTrimSetting`

自動トリム機能の有効化・無効化，トリム対象の SNMP トラップのみ・すべてのインシデントの切り替えなどは以下の具体例を参考に設定してください。

メモ

SNMP トラップインシデントを手動でデータベースから削除する場合は，`nnmtrimincidents.ovpl` コマンドを使用してください。詳細については，`nnmtrimincidents.ovpl` のリファレンスページを参照してください。

21.20.1 インシデントの自動トリムを有効にする（インシデントのアーカイブを作成しない場合）

データベース中のインシデント数（SNMP トラップ以外のすべての種別のインシデント，および syslog メッセージを含む）が 50,000 を超えた場合に，インシデントの自動トリム機能によって，10,000 個のインシデントを削除したいとき，次の手順を実行してください。インシデントは古いものから優先的に削除されます。なお，この手順ではインシデントのアーカイブは作成しません。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-jboss.properties`

- Linux：$NNM_PROPS/nms-jboss.properties

2. 次の文字列を含む行を探す。

```
com.hp.nnm.events.snmpTrapAutoTrimStartPercentage
```

3. コメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=50
```

4. 次の文字列を含む行を探す。

```
com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete
```

5. コメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=20
```

6. 次の文字列を含む行を探す。

```
com.hp.nnm.events.snmpTrapAutoTrimSetting
```

7. コメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.snmpTrapAutoTrimSetting=TrimOnly
```

アーカイブが必要な場合は TrimAndArchive を指定します。

8. 次の文字列を含む行を探す。

```
com.hp.nnm.events.allowAutoTrimAllIncidents
```

9. コメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.allowAutoTrimAllIncidents=true
```

10. 変更を保存する。
11. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

インシデント件数の上限設定であるcom.hp.nnm.events.snmpTrapMaxStoreLimit のデフォルト値は 100,000 です。この場合，データベース中のインシデント数が上限の 50%である 50,000 を超えた場合に，次の式によって 10,000 個のインシデントが古いものから削除されます。

```
(com.hp.nnm.events.snmpTrapAutoTrimStartPercentage / 100) X com.hp.nnm.events.snmpTrapMaxSto
reLimit X (com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete / 100)
※「X」は乗算記号です。
```

21.20.2 SNMP トラップインシデントの自動トリムを有効にする（インシデントのアーカイブを作成する場合）

データベース中の SNMP トラップインシデント数（syslog メッセージを含む）が 50,000 を超えた場合に，SNMP トラップインシデントの自動トリム機能によって，10,000 個の SNMP トラップインシデントを削除したいときには，次の手順を実行してください。SNMP トラップインシデントは古いものから優先的に削除されます。なお，この手順ではインシデントのアーカイブを作成します。

1. 次のファイルを編集する。
   - Windows：`%NNM_PROPS%¥nms-jboss.properties`
   - Linux：`$NNM_PROPS/nms-jboss.properties`
2. 次の文字列を含む行を探す。

```
com.hp.nnm.events.snmpTrapAutoTrimStartPercentage
```

3. コメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=50
```

4. 次の文字列を含む行を探す。

```
com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete
```

5. コメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=20
```

6. 次の文字列を含む行を探す。

```
com.hp.nnm.events.snmpTrapAutoTrimSetting
```

7. コメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.snmpTrapAutoTrimSetting=TrimAndArchive
```

   アーカイブが不要な場合は `TrimOnly` を指定します。

8. 次の文字列を含む行を探す。

```
com.hp.nnm.events.allowAutoTrimAllIncidents
```

9. コメント記号を先頭につける。またはコメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.allowAutoTrimAllIncidents=false
```

10. 変更を保存する。
11. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

インシデント件数の上限設定であるcom.hp.nnm.events.snmpTrapMaxStoreLimit のデフォルト値は 100,000 です。この場合，データベース中の SNMP トラップインシデント数（syslog メッセージを含む）が上限の 50%である 50,000 を超えた場合に，次の式によって 10,000 個の SNMP トラップインシデントが古いものから削除されます。

```
(com.hp.nnm.events.snmpTrapAutoTrimStartPercentage / 100) X com.hp.nnm.events.snmpTrapMaxSto
reLimit X (com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete / 100)
※「X」は乗算記号です。
```

削除したインシデントは次のファイルにアーカイブされます。

- Windows：%NNM_TMP%¥incidentArchive.”<*日付*>”.csv.gz
- Linux：$NNM_TMP/incidentArchive.”<*日付*>”.csv.gz

NNMi サービスを再起動するまではアーカイブファイル名は固定であり，同じファイルに追記されます。

21.20.3　アーカイブファイルのローテーション

デフォルト設定では，自動トリム機能によって作成されるアーカイブファイルのファイルサイズに上限はありません。アーカイブファイルは，特定のファイルサイズでローテーションしてディスク使用量を節約できます。サイズが 10MB に到達したときに，アーカイブファイルを 3 回ローテーションするとします。次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
2. 次の行を含むテキストブロックを探す。

   ```
   #!com.hp.nnm.events.autoTrimArchiveRotationEnabled=false
   ```

3. この行のコメントを解除し，次のように編集する。

   ```
   com.hp.nnm.events.autoTrimArchiveRotationEnabled=true
   ```

4. 次の行を含むテキストブロックを探す。

   ```
   #!com.hp.nnm.events.autoTrimArchiveRotationArchiveSize=128
   ```

5. この行のコメントを解除し，次のように編集する。

   ```
   com.hp.nnm.events.autoTrimArchiveRotationArchiveSize=10
   ```

6. 次の行を含むテキストブロックを探す。

   ```
   #!com.hp.nnm.events.autoTrimArchiveRotationRotateNumber=5
   ```

7. この行のコメントを解除し，次のように編集する。

```
com.hp.nnm.events.autoTrimArchiveRotationRotateNumber=3
```

8. NNMi 管理サーバーを再起動する。
   a. NNMi 管理サーバーでovstop コマンドを実行する。
   b. NNMi 管理サーバーでovstart コマンドを実行する。

> **重要**
>
> 高可用性 （HA）でファイルの変更を行う場合は，クラスターの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，NNMi 管理サーバーを停止および再起動する必要のある変更の場合，ovstop およびovstart コマンドを実行する前にノードをメンテナンスモードにする必要があります。

この設定では，トリムされたインシデントは以下のファイルにアーカイブされます。ファイル名にタイムスタンプは含まれないことに注意してください。

- Windows：%NNM_TMP¥incidentArchive.csv.gz
- Linux：$NNM_TMP/incidentArchive.csv.gz

アーカイブファイルのファイルサイズが指定した値 （この例では 10，単位 MB） に到達すると，アーカイブファイルの名前が「incidentArchive.csv.gz.1」に変更されます。新しいインシデントは，新しいincidentArchive.csv.gz にアーカイブされます。「incidentArchive.csv.gz.<n>」は，ローテーションによって「incidentArchive.csv.gz.<n+1>」に名前が変更されます。アーカイブファイル名の<n>が指定した値（この例では 3）になると，ローテーションによって名前が変更される代わりにファイルが削除されます。

21.20.4　保存される SNMP トラップインシデント数の最大値を変更する

SNMP トラップインシデントを長期間保存する必要がある場合や，長期間保存する必要がない場合に対応するために，データベースに保存される SNMP トラップインシデント数の最大値を変更できます。

> **重要**
>
> デフォルトではデータベース中の SNMP トラップインシデント数（syslog メッセージを含む）が 100,000 を超えると，それ以上の SNMP トラップ（syslog メッセージを含む）はインシデント化されません。最大値を 100,000 以上に変更することはパフォーマンス上の問題を引き起こすおそれがあるため，推奨されません。変更する場合は，十分に評価してご使用ください。

(1)　最大値を 100,000 未満に変更する場合

ここでは 50,000 に変更します。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
2. 次の行を探す。

```
#!com.hp.nnm.events.snmpTrapMaxStoreLimit=100000
```

3. コメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.snmpTrapMaxStoreLimit=50000
```

4. 変更を保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

(2) 最大値を 100,000 以上に変更する場合

ここでは 200,000 に変更します。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
2. 次の行を探す。

```
#!com.hp.nnm.events.snmpTrapEnforce100KLimit=true
```

3. コメント記号を削除し，次のように修正し，ファイルを保存する。

```
com.hp.nnm.events.snmpTrapEnforce100KLimit=false
```

4. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
5. 次の行を探す。

```
#!com.hp.nnm.events.snmpTrapMaxStoreLimit=100000
```

6. コメント記号を削除し，次のように修正する。

```
com.hp.nnm.events.snmpTrapMaxStoreLimit=200000
```

7. 変更を保存する。
8. 次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

21.20.5　SNMP トラップインシデントの自動トリムの状態を監視する

SNMP トラップインシデントの自動トリム機能の状態をチェックするためには，NNMi コンソールの［ヘルプ］＞［システム情報］＞［ヘルス］に SNMP トラップ数に関するメッセージが表示されていないかを確認します。また，SNMP トラップインシデントの自動トリム機能に関連して，NNMi は次のインシデントを登録します。

- データベースに保存された SNMP トラップインシデント（syslog メッセージを含む）が com.hp.nnm.events.snmpTrapMaxStoreLimit の値の 100%に到達した場合，NNMi は SnmpTrapLimitCritical を登録します。
- データベースに保存された SNMP トラップインシデント（syslog メッセージを含む）が com.hp.nnm.events.snmpTrapMaxStoreLimit の値の 95%に到達した場合，NNMi は SnmpTrapLimitMajor を登録します。
- データベースに保存された SNMP トラップインシデント（syslog メッセージを含む）が com.hp.nnm.events.snmpTrapMaxStoreLimit の値の 90%に到達した場合，NNMi は SnmpTrapLimitWarning を登録します。

重要

高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，ovstop およびovstart コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

21.20.6　SNMP トラップインシデントの自動トリムを無効にする

SNMP トラップインシデントの自動トリムを無効にするには，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
2. 次のプロパティを含む行を探す。

```
com.hp.nnm.events.snmpTrapAutoTrimSetting
```

3. 次のように修正する。

```
com.hp.nnm.events.snmpTrapAutoTrimSetting=Disabled
```

4. 変更を保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.21　NNMi 正規化プロパティを変更する

NNMi では，ホスト名とノード名の両方が大文字と小文字を区別して保存されます。NNMi コンソールのすべての検索，ソート，およびフィルタの結果も大文字と小文字を区別して返されます。使用する DNS サーバーが，すべて大文字，すべて小文字，大文字と小文字の混合などのように大文字と小文字を区別してさまざまなノード名とホスト名を返す場合，最良の結果が得られない場合があります。

ユーザーの特定のニーズに合うように，NNMi の正規化プロパティを変更できます。NNMi の初期検出シードを行う前に，これらの変更を行うことを推奨します。

導入中の初期検出を実行する前に，このセクションの設定を調整することを推奨します。

初期検出を実行してから正規化プロパティの変更を行う場合は，完全な検出を開始する nnmnoderediscover.ovpl -all スクリプトを実行できます。詳細については，nnmnoderediscover.ovpl のリファレンスページを参照してください。

次のプロパティを変更できます。

- 検出されるノード名を，UPPERCASE，LOWERCASE，またはOFF に正規化します。
- 検出されるホスト名をUPPERCASE，LOWERCASE，またはOFF に正規化します。

正規化プロパティを変更するには，次の手順に従います。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-topology.properties
   - Linux：$NNM_PROPS/nms-topology.properties
2. 検出される名称を正規化するように NNMi を設定するには，次のような行を探す。

   ```
   #!com.hp.ov.nms.topo.NAME_NORMALIZATION=OFF
   ```

   a　プロパティのコメントを解除します。

   ```
   com.hp.ov.nms.topo.NAME_NORMALIZATION=OFF
   ```

   プロパティのコメントを解除するには，行の先頭から#!文字を削除します。

   b　OFF をLOWERCASE またはUPPERCASE に変更します。

   c　変更を保存します。
3. 検出されるホスト名を正規化するように NNMi を設定するには，次のような行を探す。

   ```
   #!com.hp.ov.nms.topo.HOSTNAME_NORMALIZATION=OFF
   ```

   a　プロパティのコメントを解除します。

   ```
   com.hp.ov.nms.topo.HOSTNAME_NORMALIZATION=OFF
   ```

   プロパティのコメントを解除するには，行の先頭から#!文字を削除します。

b　OFF をLOWERCASE またはUPPERCASE に変更します。

c　変更を保存します。

4. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.21.1　初期検出後の正規化プロパティ変更時の注意事項

初期検出を実行したあとに正規化プロパティを変更すると，NNMi は，次回検出までプロパティ変更との食い違いが続きます。これを解消するには，NNMi 正規化プロパティを変更した後に，nnmnoderediscover.ovpl -all スクリプトを実行して完全検出を開始します。

21.22　データベースポートを変更する

データベースに異なるポートを使用するように NNMi を設定するには，次の手順を実行します。

1. 環境に応じて次のファイルを編集する。
   - Windows：%NNM_CONF%¥nnm¥props¥nms-local.properties
   - Linux：$NNM_CONF/nnm/props/nms-local.properties
2. 次のような行を探す。

```
#!com.hp.ov.nms.postgres.port=5432
```

3. プロパティのコメントを解除する。

```
com.hp.ov.nms.postgres.port=5432
```

   プロパティのコメントを解除するには，行の先頭から#!文字を削除します。
4. 既存の値を新しいポート番号に変更する。
5. 変更を保存する。
6. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.23 NNMi 自己監視

21.23.1 NNMi のステータスメッセージを無効にするための設定

NNMi では，メモリ，CPU，ディスクリソースなどの自己監視チェックが実行されます。NNMi 管理サーバーのリソースが少なくなる，または重大な状態が検出されると，NNMi によってインシデントが生成されます。

NNMi の稼働状態情報を表示するには，次のどれかの方法を使用します。

- NNMi コンソールで，［ヘルプ］＞［システム情報］をクリックしてから，［ヘルス］タブをクリックします。
- nnmhealth.ovpl スクリプトを実行します。

NNMi が自己監視稼働状態の例外を検出すると，NNMi コンソールの下部とフォームの上部にステータスメッセージが表示されます。次の手順を実行すると，この警告メッセージを無効にできます。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-ui.properties
   - Linux：$NNM_PROPS/nms-ui.properties
2. 次の行を含むテキストブロックを探す。

```
#!com.hp.nms.ui.health.disablewarning=false
```

3. 次の行のコメントを解除し，次のように編集する。

```
com.hp.nms.ui.health.disablewarning=true
```

4. 変更を保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.23.2 ヘルスインシデントを抑制するための設定

NNMi 管理サーバーが自己監視の異常な結果を検出すると，NNMi によってインシデントが生成されます。特定の結果のヘルスインシデントを抑制するには，次のコマンドを実行します。<conclusions>は，カンマで区切られた結果名のリストです。

- Windows の場合：

```
%InstallDir%¥bin¥nnmhealth.ovpl -suppress <conclusions>
```

- Linux の場合：

```
$InstallDir/bin/nnmhealth.ovpl -suppress <conclusions>
```

nnmhealth.ovpl による変更は，管理者が NNMi を再起動するまで続きます。NNMi の再起動後も設定を維持する場合は，次の手順を実行します。

1. 次のファイルを開く。
   - Windows：%NNM_PROPS¥nms-topology.properties
   - Linux：$NNM_PROPS/nms-topology.properties
2. 次の行を含むテキストブロックを探す。

   ```
   #!com.hp.ov.nms.health.SUPPRESSED_CONCLUSIONS=
   ```

3. 次の行のコメントを解除し，次のように編集する。

   ```
   com.hp.ov.nms.health.SUPPRESSED_CONCLUSIONS=<conclusions>
   ```

   <conclusions>は，カンマで区切られた結果名のリストです。
4. 変更を保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

   ```
   ovstop
   ovstart
   ```

結果の名前を確認する方法の詳細については，nnmhealth.ovpl のリファレンスページを参照してください。

重要

高可用性（HA）でファイルの変更を行う場合は，クラスターの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，NNMi 管理サーバーを停止および再起動する必要のある変更の場合，ovstop およびovstart コマンドを実行する前にノードをメンテナンスモードにする必要があります。

21.24　特定ノードに対して検出プロトコルを使用しないように設定する

NNMi では複数のプロトコルを使用し，ネットワークデバイス間のレイヤー 2 接続を検出しています。定義されている検出プロトコルは多数あります。例えば Link Layer Discovery Protocol（LLDP）は標準プロトコルですが，Cisco デバイス用の Cisco Discovery Protocol（CDP）のように，ベンダー固有のプロトコルも多数あります。

指定したデバイスの検出プロトコルを使用しないように NNMi を設定できます。検出プロトコルを使用しないようにすることで解決できる例を次に示します。

Enterasys デバイス：

SNMP を使用して Enterasys Discovery Protocol（EnDP）および LLDP のテーブルから一部の Enterasys デバイスに関する情報を収集すると，NNMi でメモリが不足するという問題が発生することがあります。この場合，Enterasys デバイスで EnDP および LLDP の処理をスキップするように NNMi を設定すると，この問題を防止できます。これを実行するには，デバイスの管理アドレスを `disco.SkipXdpProcessing` ファイルに追加します。詳細については，「21.24.1　検出プロトコルを使用しないように設定する」を参照してください。

一部の Enterasys デバイスの新バージョンのオペレーティングシステムでは，`set snmp timefilter break` コマンドがサポートされています。このような Enterasys デバイスでは，`set snmp timefilter break` コマンドを実行します。このコマンドを使用してデバイスを設定した場合，このデバイスを `disco.SkipXdpProcessing` ファイルに追加する必要はありません。

Nortel デバイス：

多くの Nortel デバイスでは SynOptics Network Management Protocol（SONMP）を使用し，レイヤー 2 レイアウトおよび接続を検出します。一部のデバイスでは複数のインタフェースで同一 MAC アドレスを使用するため，このプロトコルで適切に動作しません。相互接続した 2 つの Nortel デバイスがインタフェースの誤ったセット間でレイヤー 2 接続を示し，接続が接続ソース SONMP を示す場合，この問題が発生することがあります。

この例では，SONMP プロトコルを使用しないように NNMi を設定し，デバイスのレイヤー 2 接続を引き出して，誤った接続に関与しているとして表示しないことを推奨します。これを実行するには，2 つのデバイスの管理アドレスを`disco.SkipXdpProcessing` ファイルに追加します。詳細については，「21.24.1　検出プロトコルを使用しないように設定する」を参照してください。

21.24.1　検出プロトコルを使用しないように設定する

検出プロトコルを使用しないように設定する必要がある場合は，次の手順を実行します。

1. 次のファイルを作成する。
   - Windows：`%NnmDataDir%shared¥nnm¥conf¥disco¥disco.SkipXdpProcessing`
   - Linux：`$NnmDataDir/shared/nnm/conf/disco/disco.SkipXdpProcessing`

     `disco.SkipXdpProcessing` ファイルでは，大文字と小文字が区別されます。

2. 検出プロトコルを使用しないように設定するすべてのデバイスについて，デバイスの管理アドレスを disco.SkipXdpProcessing ファイルに追加する。

   詳細については，disco.SkipXdpProcessing のリファレンスページを参照してください。

3. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

重要

1 つまたは複数のノードの検出プロトコルを使用しないように設定すると，管理対象ネットワークのレイヤー 2 レイアウトの精度が多少落ちることがあります。

ovjboss サービスは起動時にdisco.SkipXdpProcessing ファイルを読み込みます。NNMi 管理サーバーの起動後に変更をした場合は，この手順で示すように NNMi 管理サーバーを再起動してください。

Enterasys デバイスでset snmp timefilter break コマンドを実行した場合は，デバイスの管理アドレスをdisco.SkipXdpProcessing ファイルから削除し，この手順で示すように NNMi 管理サーバーを再起動します。NNMi は，検出プロトコルを使用したとき，より正確なレイヤー 2 マップを表示します。

詳細については，disco.SkipXdpProcessing のリファレンスページを参照してください。

21.25　二次的な根本原因管理イベントにアクションを設定する

NNMi はデフォルトでは二次的な根本原因管理イベントに対してアクションを実行しません。

このことは不要なアクションの生成を抑止することに役立っています。例えば，NNMi がInterfaceDown インシデントを検知し，その直後に対応するカードがダウンしたと判別したら，ダンプニングが使用されている場合，CardDown インシデントが根本原因となり，InterfaceDown インシデントは二次的な根本原因インシデントとなります。

この場合，アクションは新しい根本原因（CardDown）に対して適用されるため，InterfaceDown インシデントに対しては要求されません。

二次的な根本原因管理イベントに対するアクションを有効化するには，次の手順を実行します。

1. 環境に応じて次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - Linux：$NNM_PROPS/nms-jboss.properties
2. 次の行を含むテキストブロックを探す。

```
#!com.hp.nnm.events.action.runActionOnSecRootCauseMgmtEvent=false
```

3. この行のコメントを解除し，次のように編集する。

```
com.hp.nnm.events.action.runActionOnSecRootCauseMgmtEvent=true
```

4. nms-jboss.properties ファイルを保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

21.26 計画停止

NNMi では，nnmscheduledoutage.ovpl コマンドを使用して任意のノードセットの停止をスケジュールできます。例えば，ルーターセットの毎週のメンテナンスで停止をスケジュールしたり，特定のノードの電源を交換したりできます。

詳細については，nnmscheduledoutage.ovpl のリファレンスページを参照してください。

メモ

NNMi を使用した停止のスケジュールの詳細については，NNMi ヘルプを参照してください。

21.27　センサーステータスの設定

NNMi には，ステータス判別用に監視できる次の物理センサーとノードセンサーが含まれています。

表 21-2　物理センサーとノードセンサー

物理センサー	デフォルトで物理コンポーネントにステータスを伝達	ノードセンサー	デフォルトでノードにステータスを伝達
FAN	はい	CPU	いいえ
POWER_SUPPLY	はい	MEMORY	はい
TEMPERATURE	いいえ	BUFFERS	いいえ
VOLTAGE	いいえ	DISK_SPACE	いいえ
BACK_PLANE	はい		

重要

デフォルトでは，FAN，POWER_SUPPLY，BACK_PLANE，および MEMORY がステータスを物理コンポーネントレベルに伝達します。例えば，ファンが赤色のステータスインジケーターを示している場合，対応する物理コンポーネント（シャーシ）は黄色のステータスインジケーターを受け取ります。この場合，シャーシのステータスを表示しているユーザーには，そのシャーシのコンポーネントに何らかの障害があることが警告されます。

21.27.1　物理センサーステータスの設定

次のセクションの手順を実行して，物理センサーでステータスを物理コンポーネント（シャーシなど）レベルに伝達するかどうかを設定できます。

(1)　物理コンポーネントへの物理センサーステータスの伝達

1. 次のディレクトリに`nnm-apa.properties` という名前の新しいプロパティファイルを作成する（このファイルが存在しない場合）。
   - Windows：`%NnmDataDir%shared¥nnm¥conf¥props`
   - Linux：`$NnmDataDir/shared/nnm/conf/props`
2. テキストエディタを使用して，プロパティファイル内に次のテキストを挿入する。

   ```
   com.hp.ov.nms.apa.PhysSensorPropagateToPhysicalComponentStatus_<タイプ>=true
   ```

   <タイプ>は物理センサーです。詳細については，「21.27　センサーステータスの設定」を参照してください。
3. プロパティファイルを保存する。

4. 次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

重要

高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，ovstop およびovstart コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

(2)　物理コンポーネントに伝達しない物理センサーステータスの設定

1. 次のディレクトリにnnm-apa.properties という名前の新しいプロパティファイルを作成する（このファイルが存在しない場合）。
   - Windows の場合：%NnmDataDir%shared¥nnm¥conf¥props
   - Linux の場合：$NnmDataDir/shared/nnm/conf/props
2. テキストエディタを使用して，プロパティファイル内に次のテキストを挿入する。

```
com.hp.ov.nms.apa.PhysSensorNoPropagateToPhysicalComponentStatus_<タイプ>=true
```

   <タイプ>は物理センサーです。詳細については，「21.27　センサーステータスの設定」を参照してください。
3. プロパティファイルを保存する。
4. 次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

重要

高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，ovstop およびovstart コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

(3)　物理センサーステータス値の上書き

デフォルトでは，3 つのセンサーのステータス値（**[なし]**，**[注意域]**，および **[利用不可]**）は，Causal Engine によって **[正常域]** ステータスにマッピングされます。デフォルトのステータスマッピングは，**[なし]**，**[注意域]**，**[利用不可]** を **[危険域]** にマッピングするように上書きできます。

物理センサーのステータス値を上書きするには，次の手順を実行します。

1. 次のディレクトリにnnm-apa.propertiesという名前の新しいプロパティファイルを作成する（このファイルが存在しない場合）。
   - Windowsの場合：%NnmDataDir%shared¥nnm¥conf¥props
   - Linuxの場合：$NnmDataDir/shared/nnm/conf/props
2. テキストエディタを使用して，プロパティファイル内に必要に応じて次の行の1つ，2つ，または3つすべてを挿入する。

```
com.hp.ov.nms.apa.PhysSensorValueReMappedToDown_NONE=true
com.hp.ov.nms.apa.PhysSensorValueReMappedToDown_Warning=true
com.hp.ov.nms.apa.PhysSensorValueReMappedToDown_Unavailable=true
```

3. プロパティファイルを保存する。
4. 次のコマンドを実行してNNMiを再起動する。

```
ovstop
ovstart
```

重要

- ［利用不可］の状態を［未ポーリング］状態にマッピングできます（［利用不可］は測定機能が利用できないことを指すため）。この状態は，多くの場合コンポーネントの機能不全ではなくセンサーの機能不全で発生します。［利用不可］を［未ポーリング］にマッピングするには，手順2で次のテキストを使用する以外は上記と同じ手順を実行します。

  com.hp.ov.nms.apa.PhysicalSensorValueReMappedToUnpolled_Unavailable= true
- 高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA設定を使用するNNMiでは，変更でNNMi管理サーバーの停止と再起動が必要な場合，ovstopおよびovstartコマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMiをメンテナンスモードにする」を参照してください。

21.27.2　ノードセンサーステータスの設定

次のセクションの手順を実行して，ノードセンサーでステータスをノードレベルに伝達するかどうかを設定できます。

(1)　ノードへのノードセンサーステータスの伝達

1. 次のディレクトリにnnm-apa.propertiesという名前の新しいプロパティファイルを作成する（このファイルが存在しない場合）。
   - Windowsの場合：%NnmDataDir%shared¥nnm¥conf¥props

- Linux の場合：$NnmDataDir/shared/nnm/conf/props

2. テキストエディタを使用して，プロパティファイル内に次のテキストを挿入する。

```
com.hp.ov.nms.apa.NodeSensorPropagateToNodeStatus_<タイプ>=true
```

<タイプ>は物理センサーです。詳細については，「21.27　センサーステータスの設定」を参照してください。

3. プロパティファイルを保存する。
4. 次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

重要

高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，ovstop およびovstart コマンドを実行する前にノードをメンテナンスモードにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

(2)　ステータスをノードに伝達しないようにするためのノードセンサーの設定

1. 次のディレクトリにnnm-apa.properties という名前の新しいプロパティファイルを作成する（このファイルが存在しない場合）。
   - Windows の場合：%NnmDataDir%shared¥nnm¥conf¥props
   - Linux の場合：$NnmDataDir/shared/nnm/conf/props
2. テキストエディタを使用して，プロパティファイル内に次のテキストを挿入する。

```
com.hp.ov.nms.apa.NodeSensorNoPropagateToNodeStatus_<タイプ>=true
```

<タイプ>は物理センサーです。詳細については，「21.27　センサーステータスの設定」を参照してください。

3. プロパティファイルを保存する。
4. 次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

重要

高可用性（HA）でファイルを変更する場合は，クラスタの両方のノードに変更を加える必要があります。HA 設定を使用する NNMi では，変更で NNMi 管理サーバーの停止と再起動が必要な場合，ovstop およびovstart コマンドを実行する前にノードをメンテナンスモー

ドにする必要があります。詳細については，「19.6.1　NNMi をメンテナンスモードにする」を参照してください。

21.28 Destination Unreachable を受信した場合でも，リトライ処理を継続するように設定する

NNMi はデフォルトの設定の場合，監視ノードに対してステータスポーリングを実行した際に ICMP の Destination Unreachable を受信した場合，通信不可と判断し，ICMP のリトライ処理を行いません。

Destination Unreachable を受信してもリトライ処理を継続する場合は，次の手順を実行します。

1. 環境に応じて次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-communication.properties
   - Linux：$NNM_PROPS/nms-communication.properties
2. 次のような行を探す。

```
#!com.hp.ov.icmp.allowRetryIcmpIfDestinationUnreachable=false
```

3. プロパティのコメントを解除し，次のように編集する。

```
com.hp.ov.icmp.allowRetryIcmpIfDestinationUnreachable=true
```

4. nms-communication.properties ファイルを保存する。
5. 次のコマンドを実行して，NNMi を再起動する。

```
ovstop
ovstart
```

22 NNMi 管理サーバーの変更

ほかのシステムで NNMi 設定を複製できます。例えば，テスト環境から運用環境に移動したり，NNMi 管理サーバーのハードウェアを変更したりできます。NNMi 設定に影響を及ぼさないで，NNMi 管理サーバーの IP アドレスを変更できます。

22.1 NNMi 設定移動の準備のベストプラクティス

次のベストプラクティスは，NNMi の設定を異なるシステムへ移動するときに有効です。

- ノードグループ設定で，管理対象ノードの識別にホスト名を使っている場合，運用環境およびテスト環境の NNMi 管理サーバーは同じ DNS サーバーを使う必要があります。運用環境とテスト環境で異なる DNS サーバーを使っている場合，管理対象ノードの解決済みの名前が変更されると，2 つの NNMi 管理サーバーの間でポーリング設定に差異が生じる場合があります。
- 設定の作成者を限定して，エクスポートできます。自分のグループまたは会社で一意の新しい［作成者］を作成します。次の項目を作成または変更するときは，この作成者の値を指定します。

  －デバイスのプロファイル

  －インシデントの設定

  －メニュー

  －メニュー項目

  －カスタム相関処理の設定

  －アイコン

  －MIB 式

  －トラップログ記録設定

22.2　NNMi 設定およびデータベースを移動する

NNMi の設定とデータベースを，例えばテストシステムから本番システムなどへ移動するには，ソース（テスト）システム上のすべての NNMi データをバックアップしてから，バックアップをターゲット（本稼働）システムにリストアします。バックアップの実行後 NNMi データベースが変更されないようにするため，すべての NNMi プロセスを停止し，オフラインバックアップを作成してください。

（例）

```
nnmbackup.ovpl -type offline -scope all -target nnmi_backups¥offline
```

「20.3.2　異なるシステムでのリストア」にリストされた項目が両方のシステムで同じであることを確認してから，次の例のようなコマンドを実行します。

（例）

```
nnmrestore.ovpl -source nnmi_backups¥offline¥newest_backup
```

重要

NNMi は同じ SSL 証明書を使用して，データベースへのアクセスおよび NNMi コンソールへの HTTPS アクセスをサポートします。データベースへアクセスするための証明書は，ソースシステム上で NNMi プロセスを最初に開始したときに作成されました。この証明書はバックアップおよびリストアデータに含まれています。この証明書がないと，NNMi はターゲットシステムからデータベースにアクセスできません。

ただし，NNMi コンソールへの HTTPS アクセスの場合は，SSL 証明書をターゲットシステムに生成する必要があります。JBoss の現在の実装が証明書のマージをサポートしていないため，NNMi は別のシステムからのデータをリストアして設定されたシステム上での NNMi コンソールへの HTTPS アクセスはサポートしていません。ターゲットシステムが NNMi コンソールへの HTTPS アクセスをサポートする必要がある場合は，「22.3　NNMi 設定を移動する」の手順を実行してから，ターゲットシステム上で新たにデータ収集を開始します。

22.3 NNMi 設定を移動する

nnmconfigexport.ovpl コマンドを使用して，NNMi 設定を XML ファイルに出力します。次に，nnmconfigimport.ovpl コマンドを使って，XML ファイルから新しいシステムの NNMi にこの設定をインポートします。

重要

nnmconfigimport.ovpl スクリプトを使用してファイルをインポートする前に，nnmconfigexport.ovpl スクリプトでエクスポートしたファイルを編集しないでください。

これらのコマンドの詳細については，該当するリファレンスページを参照してください。

メモ

- nnmconfigexport.ovpl コマンドでは SNMPv3 資格情報は保持されません。詳細については，nnmconfigexport.ovpl のリファレンスページを参照してください。
- NNMi 設定だけを移動できます。ある NNMi 管理サーバーから異なる NNMi 管理サーバーへのトポロジまたはインシデントデータの移動をサポートしません。

22.4　スタンドアロンの NNMi 管理サーバーの IP アドレスを変更する

NNMi 管理サーバーの IP アドレスを変更する必要がある場合は，NNMi を停止してから実施してください。アドレス変更後には，変更前のアドレスに対応するライセンスキーを削除し，変更後のアドレスに対応するライセンスキーを適用してから，NNMi を起動してください。

22.5　NNMi 管理サーバーのホスト名またはドメイン名を変更する

重要

NNMi 管理サーバーが NNMi アプリケーションフェイルオーバーを構成している，グローバルネットワーク管理を構成している，または高可用性（HA）クラスタのメンバーの場合は，サポートサービスに問い合わせてください。

NNMi 管理サーバーをディレクトリサービスと SSL で接続するよう設定している場合は，ディレクトリサーバーの証明書をトラストストアーに追加し直す必要があります。この節の手順を実行したあと，「10.3.8　ディレクトリサービスへの SSL 接続を設定する」の手順を実施してください。

NNMi 管理サーバーのホスト名，ドメイン名，またはその両方を変更するには，NNMi 管理サーバーの新しい完全修飾ドメイン名（FQDN）を使用するように NNMi を設定します。

（例）

```
nnmsetofficialfqdn.ovpl -force newnnmi.servers.example.com
```

nnmsetofficialfqdn.ovpl 実行後，設定を反映するために NNMi を再起動してください。

詳細については，nnmsetofficialfqdn.ovpl のリファレンスページを参照してください。

重要

FQDN は，ドメイン名と組み合わされたホスト名です。このどちらかを変更すると，NNMi 管理サーバーの FQDN を変更することになります。SSL 証明書は，常に FQDN にリンクされます。証明書の共通名（CN）フィールドは，サーバーの FQDN と一致する必要があります。このため，FQDN を変更する場合は，一致する CN を持つ新しい SSL 証明書が必要になります。nnmsetofficialfqdn.ovpl コマンドは，NNMi 管理サーバーの FQDN を更新し，新しい FQDN と一致する新しい自己署名証明書も作成します。ただし，CA 証明書を使用している場合は，新しい CA 証明書を生成する必要があります。詳細については，「10.3.2　CA 署名証明書の生成」を参照してください。

FQDN を変更したかどうかに関係なく，NNMi 管理サーバーの IP アドレスを変更する場合は，新しいライセンスを取得する必要があります。詳細については，「22.4　スタンドアロンの NNMi 管理サーバーの IP アドレスを変更する」を参照してください。

23 NNMi セキュリティ

セキュリティについて説明します。

23.1　組み込みデータベースツールのパスワードを入力する

NNMi で組み込みデータベースツール（psql など）を実行するには，パスワードを入力する必要があります。NNMi によってデフォルトのパスワードが設定されており，ユーザーは`nnmchangeembdbpw.ovpl` スクリプトを使用してこのパスワードを変更する必要があります。`nnmchangeembdbpw.ovpl` スクリプトを実行するには，Windows システムの場合は管理者，Linux システムの場合はルートとしてログインする必要があります。詳細については，`nnmchangeembdbpw.ovpl` リファレンスページを参照してください。

HA 環境では，プライマリクラスタノードでだけ，`nnmchangeembdbpw.ovpl` スクリプトを実行します。アプリケーションによって自動的にセカンダリクラスタノードにパスワードがコピーされるため，その後のユーザーの操作は必要ありません。

23.2 TLS プロトコルの設定

NNMi はデフォルトでは，HTTPS 通信に対して TLSv1.2 および TLSv1.3 プロトコルをサポートしています。

従来のクライアントをサポートするために，以前の安全性の低いプロトコルが必要な場合を除き，NNMi では TLSv1.2 以降のみを使用することをお勧めします。

使用するプロトコルバージョンを変更するには，次の手順に従います。

1. NNMi 管理サーバーにログオンします。
2. テキストエディタで次のファイルを開きます。
   - Windows の場合：`%NnmDataDir%nmsas¥NNM¥server.properties`
   - Linux の場合：`/var/opt/OV/nmsas/NNM/server.properties`
3. 使用するプロトコルのカンマ区切りリストを使用して，`com.hp.ov.nms.ssl.PROTOCOLS` プロパティを追加，または更新します。
   例えば，TLSv1，TLSv1.1，TLSv1.2，および TLSv1.3 プロトコルを使用する場合は，次のような設定行が`server.properties` ファイルに存在するようにします。

   ```
   com.hp.ov.nms.ssl.PROTOCOLS=TLSv1.3,TLSv1.2,TLSv1.1,TLSv1
   ```

4. 次のコマンドを実行して，NNMi プロセスを再起動します。
   - Windows の場合：

     ```
     %NnmInstallDir%bin¥ovstop -c
     %NnmInstallDir%bin¥ovstart -c
     ```

   - Linux の場合：

     ```
     /opt/OV/bin/ovstop -c
     /opt/OV/bin/ovstart -c
     ```

23.3 NNMi データ暗号化

NNMi では製品のさまざまなエリアにデータ暗号化が組み込まれています。

（例）

NNMi は，ユーザーアカウント用のパスワードを暗号化された形式で NNMi データベースに保存します。

23.3.1 暗号化およびユーザーアカウントパスワード

重要

この情報は，ライトウェイトディレクトリアクセスプロトコル（LDAP）または Common Access Card（CAC）アカウントには適用されません。

NNMi コンソールを使用して作成された NNMi ユーザーアカウントは NNMi データベースに保存されます。これらのユーザーのパスワードはハッシュされ，データベースに保存されます。

ユーザーが NNMi コンソールにサインインするか，コマンドラインインタフェース（CLI）ツールを使用する場合，指定したパスワードはハッシュされ，データベースに保存されたハッシュ値と比較されます。ユーザーが正しいパスワードを指定すると，これらの 2 つのハッシュされた文字列が一致し，ユーザーは認証されます。

NNMi の従来のバージョン（10-50 以前）はユーザーパスワードをハッシュするための暗号化アルゴリズムを使用していましたが，この方式は古くなりました。NNMi 11-00 はユーザーアカウントパスワードにより強力なアルゴリズムを使用しています。ただし，ハッシュは一方向の暗号化であるため，復号化は不可能であり，NNMi 10-50 から 11-00 へのアップグレード中に再暗号化することになります。

アップグレード時に，すべての既存のユーザーは従来の暗号化アルゴリズムを使用したデータベースに保存されたパスワードを保持します。ただし，従来のアルゴリズムを使用してハッシュされたパスワードを持つユーザーがログオンに成功すると，指定したパスワードは自動的に暗号設定ファイルで指定された新しいハッシュアルゴリズムを使用して再暗号化されます。

つまり，アップグレード後に各ユーザーが初めてログインするたびに，すべてのパスワードが少しずつ新しいアルゴリズムに更新されることになります。同じことが，将来的に暗号設定が変更された場合にも言えます。ユーザーパスワードは，次にログオンに成功したときに新しいハッシュアルゴリズムにアップグレードされます。

重要

- ユーザーパスワードをアップグレードするには，<allowed>ブロックにリストされている従来のアルゴリズム（例えば，MD5）が存在している必要があります。したがって，すべ

てのパスワードが移行されるまで<allowed>ブロックにリストされている従来のアルゴリズムを残しておいてください。

- <allowed>ブロックに従来のアルゴリズムが存在していないと，データベースでハッシュされた既存のパスワードは再ハッシュすることができません。したがって，関連づけられたユーザーはログオンできないため，NNMi は新しいアルゴリズムを使用してパスワードを再暗号化できません。
- 従来のアルゴリズムを<allowed>ブロックから削除した場合，管理者は影響を受けるユーザーを削除して再作成するか，パスワードが従来のアルゴリズムで暗号化されたユーザーのそれぞれのパスワードをリセットする必要があります。

次のコマンドを使用して，ユーザーのパスワードが暗号設定ファイルにリストされているアルゴリズムを使用しているか，またはユーザーのパスワードが暗号設定ファイルで指定されなくなった従来のアルゴリズムで暗号化されているかを判断します。

```
nnmsecurity.ovpl -listUserAccounts legacy
```

詳細については，nnmsecurity.ovpl のリファレンスページを参照してください。

24 バージョン 9・10・11・12 の NNMi からの移行

この章では，幾つかの想定されるバージョンアップの例について説明します。NNMi アプリケーションフェイルオーバー設定で実行しているバージョン 9，バージョン 10，バージョン 11，バージョン 12 の NNMi 管理サーバーをバージョンアップする場合，一時的なアプリケーションフェイルオーバーの設定解除，NNMi 管理サーバーのバージョンアップ，アプリケーションフェイルオーバーの再設定という順番のアップグレードパスがサポートされています。高可用性クラスタ（HA）で実行しているバージョン 9，バージョン 10，バージョン 11，バージョン 12 の NNMi をバージョンアップする場合は，リリースノートを参照してください。

24.1　NNMi 管理サーバーをバージョンアップする

24.1.1　バージョン 13-00 の NNMi 管理サーバーをバージョンアップする

ここでは，バージョン 13-00 で実行中の NNMi 管理サーバーをバージョンアップする手順を次に示します。

メモ

NNMi 管理サーバーをバージョンアップする前に，「1.　インストール前チェックリスト」を参照してください。

1. `nnmbackup.ovpl` スクリプトを使用して，NNMi 管理サーバーをバックアップする。

   このバックアップを使用するのは移行が失敗した場合だけです。詳細については，`nnmbackup.ovpl` のリファレンスページを参照してください。

2. 「2.　NNMi のインストールとアンインストール」の手順に従って，NNMi 13-10 の NNMi 管理サーバーにインストールする。

3. NNMi 管理サーバーの情報が正しく移行されたことを確認する。

24.1.2　バージョン 9・10・11・12 の NNMi 管理サーバーをバージョンアップする

ここでは，バージョン 9，バージョン 10，バージョン 11，バージョン 12 で実行中の NNMi 管理サーバーをバージョンアップする手順を次に示します。

1. NNMi 13-00 のセットアップガイド※，およびリリースノートの手順に従って，実行中の NNMi 管理サーバーをバージョン 13-00 の NNMi 管理サーバーにバージョンアップする。

   注※　セットアップガイドとは，次のマニュアルを指します。

   JP1/Network Node Manager i セットアップガイド（3021-3-L32）

2. 「24.1.1　バージョン 13-00 の NNMi 管理サーバーをバージョンアップする」を参照して，バージョン 13-00 で実行中の NNMi 管理サーバーをバージョンアップする。

24.2　別の NNMi 管理サーバーにバージョンアップする

ここでは，既存（以降，ソースといいます）の NNMi 管理サーバーの設定を維持しながら，新規システム上で NNMi 13-10 にバージョンアップする手順について説明します。

> **メモ**
>
> NNMi 管理サーバーをバージョンアップする前に，「1.　インストール前チェックリスト」を参照してください。

次の手順は，ソースの NNMi 管理サーバーからターゲットの NNMi 管理サーバーにデータをコピーする方法を説明したものです。この手順は，NNMi 13-00 がソースの NNMi 管理サーバーで実行されていることを前提としています。

1. nnmbackup.ovpl スクリプトを使用して，ソースの NNMi 管理サーバーをバックアップする。バックアップファイルにラベルを付ける。

   このバックアップを使用するのは移行が失敗した場合だけです。詳細については，nnmbackup.ovpl のリファレンスページを参照してください。

2. 「2.　NNMi のインストールとアンインストール」の手順に従って，ソースの NNMi 管理サーバー上に NNMi 13-10 をインストールする。

3. NNMi 13-10 がソースの NNMi 管理サーバー上で正しく動作していることを確認する。

4. nnmbackup.ovpl スクリプトを使用して，NNMi 13-10 をソースの NNMi 管理サーバー上にバックアップする。このバックアップファイルにラベルを付ける。

   データをターゲットの NNMi 管理サーバーにコピーしてください。詳細については，nnmbackup.ovpl のリファレンスページを参照してください。

5. 「2.　NNMi のインストールとアンインストール」の手順に従って，ターゲットの NNMi 管理サーバー上に NNMi 13-10 をインストールする。

   手順 4.からデータを移行するには，ターゲットの NNMi 管理サーバーが同じオペレーティングシステムで実行中である必要があります。NNMi では，別のオペレーティングシステム上で実行中の NNMi 管理サーバーへのデータ移行はサポートされていません。

6. nnmrestore.ovpl スクリプトを使用して，NNMi のデータベース情報をターゲットサーバーにコピーする。

   詳細については，nnmrestore.ovpl のリファレンスページを参照してください。

7. 新規ライセンスを取得し，ターゲットの NNMi 管理サーバーにインストールする。

8. ターゲットの NNMi 管理サーバー情報が既存の NNMi 管理サーバーから正常に移行されたことを確認する。

24.3 NNMi 13-00 からのグローバルマネージャーとリージョナルマネージャーのアップグレード

24.3.1 グローバルネットワーク管理によってサポートされている NNMi のバージョン

NNMi 13-10 が実行されているグローバルマネージャーに接続された，NNMi 13-00 以前が実行されているリージョナルマネージャーはサポートしていません。グローバルマネージャーとリージョナルマネージャーの両方で，同一バージョンの NNMi を実行する必要があります。

24.3.2 グローバルネットワーク管理のアップグレード手順

グローバルネットワーク管理環境で設定された NNMi 管理サーバーを NNMi 13-10 にアップグレードする場合，グローバルマネージャーとリージョナルマネージャー間の接続は，グローバルマネージャーとリージョナルマネージャーの両方が NNMi 13-10 にアップグレードされるまで切断されます。そのため，全体のダウンタイムを最小限に抑えるには，すべてのサーバーをほぼ同時にアップグレードすることをお勧めします。

例えば，次の手順で NNMi 管理サーバーをアップグレードできます。

1. リージョナルマネージャーを NNMi 13-10 にアップグレードし，正しく動作することを確認する。

   アップグレード後，リリースノートの手順に従って新しいライセンスを適用してください。

   証明書のリポジトリを PKCS #12 リポジトリに移行していない場合は，アップグレード完了後に，リージョナルマネージャーにて「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従い，PKCS #12 リポジトリに移行してください。

   リージョナルマネージャーのアップグレード中，グローバルマネージャーは切断されたままになります。

2. グローバルマネージャーを NNMi 13-10 にアップグレードする。

   アップグレード後，リリースノートの手順に従って新しいライセンスを適用してください。

   証明書のリポジトリを PKCS #12 リポジトリに移行していない場合は，アップグレード完了後に，グローバルマネージャーにて「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従い，PKCS #12 リポジトリに移行してください。

重要

次の点に注意してください。

- アップグレード後，ステータスおよびインシデントへの更新が遅延することがあります。

24.4 アプリケーションフェイルオーバー構成の NNMi 13-10 へのアップグレード

24.4.1 アプリケーションフェイルオーバー構成の NNMi 13-00 からのアップグレード

NNMi アプリケーションフェイルオーバー設定で実行している 13-00 の NNMi をアップグレードする場合，次の手順に従ってください。

(1) アプリケーションフェイルオーバー構成の NNMi 13-10 へのアップグレード

アプリケーションフェイルオーバーを設定している NNMi 管理サーバーをアップグレードするには，次の手順を実行します。

1. 万一に備えて，以降の操作を行う前に，アクティブおよびスタンバイ NNMi 管理サーバーの両方で，`nnmconfigexport.ovpl` スクリプトを実行する。

   詳細については，「4.2　ベストプラクティス：既存の設定を保存する」を参照してください。

2. 万一に備えて，以降の操作を行う前に，アクティブおよびスタンバイ NNMi 管理サーバーの NNMi データをバックアップする。

   詳細については，「20.2.2　バックアップ領域」を参照してください。

3. アクティブ NNMi 管理サーバーで次の手順を実行する。

   `nnmcluster` の手順が機能するには，NNMi を実行している必要があります。この手順を完了すると，手順 7.で示すスタンバイ NNMi 管理サーバーの起動が速くなります。

   a　`nnmcluster` コマンドを実行します。

   b　NNMi に入力を求められたら，「`dbsync`」と入力し，[Enter] キーを押します。表示される情報に次のメッセージが含まれていることを確認します。

   - `ACTIVE_DB_BACKUP`：アクティブ NNMi 管理サーバーが新しいバックアップを実行しています。
   - `ACTIVE_NNM_RUNNING`：アクティブ NNMi 管理サーバーが，前のメッセージによって示されたバックアップを完了しました。
   - `STANDBY_RECV_DBZIP`：スタンバイ NNMi 管理サーバーは，アクティブ NNMi 管理サーバーから新しいバックアップを取得しています。
   - `STANDBY_READY`：スタンバイ NNMi 管理サーバーは，アクティブ NNMi 管理サーバーで障害が発生した場合に実行できる準備が整えられています。

   c　`exit` または`quit` を実行して，手順 a で開始したインタラクティブ`nnmcluster` プロセスを停止します。

4. スタンバイ NNMi 管理サーバーで`nnmcluster -shutdown` コマンドを実行する。

   スタンバイ NNMi 管理サーバーのすべての`nnmcluster` プロセスをシャットダウンします。

5. スタンバイ NNMi 管理サーバーでnnmcluster ノードが動作していないことを確認するには，スタンバイ NNMi 管理サーバーで次の手順を実行する。

a　nnmcluster コマンドを実行します。

b　(SELF)とマークされているもの以外にnnmcluster ノード(ローカル)が存在しないことを確認します。1 つ以上のリモートノードが存在する場合があります。

c　exit またはquit を実行して，手順 a で開始したインタラクティブnnmcluster プロセスを停止します。

6. 次の手順をスタンバイ NNMi 管理サーバーで実行し，アプリケーションフェイルオーバーを一時的に無効にする。

a　次のファイルを編集します。

- Windows：%NNM_SHARED_CONF%¥props¥nms-cluster.properties
- Linux：$NNM_SHARED_CONF/props/nms-cluster.properties

b　com.hp.ov.nms.cluster.name パラメーターをコメントにします。

c　変更を保存します。

d　次のファイルを編集します。

- Windows：%NNM_DB%¥Postgres¥postgresql.conf
- Linux：$NNM_DB/Postgres/postgresql.conf

e　次に示すアプリケーションフェイルオーバーの設定内容を削除します。

・"# The following lines were added by the NNM cluster."で始まる行
・"archive_command = "で始まる行
・"archive_timeout = "で始まる行
・"max_wal_senders = "で始まる行
・"archive_mode = "で始まる行
・"wal_level = "で始まる行
・"hot_standby = "で始まる行
・"wal_keep_segments = "で始まる行
・"listen_addresses = "で始まる行

f　変更を保存します。

g　次の空ファイルを作成します。

- Windows：%NNM_TMP%¥postgresTriggerFile
- Linux：$NNM_TMP/postgresTriggerFile

7. スタンバイ NNMi 管理サーバーでプロセスを開始してから停止する。

a　スタンバイ NNMi 管理サーバーでovstart コマンドを実行します。ovstart コマンドを実行すると，スタンバイ NNMi 管理サーバーはトランザクションログをアクティブ NNMi 管理サーバーからインポートします。

b　ovstart コマンドの完了後，ovstatus -v コマンドを実行します。すべての NNMi サービスで，[実行中] 状態が表示されます。

c　スタンバイ NNMi 管理サーバーでovstop コマンドを実行します。

8.「2.　NNMi のインストールとアンインストール」およびリリースノートの指示に従い，スタンバイ NNMi 管理サーバーを NNMi 13-10 にアップグレードする。

アップグレード後，リリースノートの手順に従って新しいライセンスを適用してください。

証明書のリポジトリを PKCS #12 リポジトリに移行していない場合は，アップグレード完了後に，以前のスタンバイ NNMi 管理サーバーにて「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従い，PKCS #12 リポジトリに移行してください。

以前のアクティブ NNMi 管理サーバーが NNMi 13-00 以前を実行し，以前のスタンバイ NNMi 管理サーバーが NNMi 13-10 を実行しています。両方の NNMi 管理サーバーが個別に動作し，データベースは同期していません。つまり両方の NNMi 管理サーバーがネットワークを並行して監視しています。

アップグレードを完了してこの状況を解決するには，以前のアクティブなクラスタノードを NNMi 13-10 にアップグレードします。このアップグレードを完了する間，以前のスタンバイノードをオペレータに一時的に使用させてネットワークを監視させます。

この手順の残りの部分では，以前のアクティブなクラスタノードのデータベース情報を維持して，以前のスタンバイノードのデータベース情報を破棄することを想定しています。

9. 以前のアクティブ NNMi 管理サーバーでnnmcluster -halt コマンドを実行する。

10. 以前のアクティブ NNMi 管理サーバーでnnmcluster ノードが動作していないことを確認するには，以前のアクティブ NNMi 管理サーバーで次の手順を実行する。

a　nnmcluster コマンドを実行します。

b　(SELF)とマークされているもの以外にnnmcluster ノード(ローカル)が存在しないことを確認します。1 つ以上のリモートノードが存在する場合があります。

c　exit またはquit を実行して，手順 a で開始したインタラクティブnnmcluster プロセスを停止します。

11. 次の手順を以前のアクティブ NNMi 管理サーバーで実行し，アプリケーションフェイルオーバーを一時的に無効にする。

a　次のファイルを編集します。

- Windows：%NNM_SHARED_CONF%¥props¥nms-cluster.properties
- Linux：$NNM_SHARED_CONF/props/nms-cluster.properties

b　com.hp.ov.nms.cluster.name パラメーターをコメントにします。

c　変更を保存します。

12.「2.　NNMi のインストールとアンインストール」の指示に従い，以前のアクティブ NNMi 管理サーバーを NNMi 13-10 にアップグレードする。

アップグレード後，リリースノートの手順に従って新しいライセンスを適用してください。

証明書のリポジトリを PKCS #12 リポジトリに移行していない場合は，アップグレード完了後に，以前のアクティブ NNMi 管理サーバーにて「10.2　アップグレードされた NNMi 環境で新しいキーストアーを使用するための設定」の手順に従い，PKCS #12 リポジトリに移行してください。

2 つのサーバーで NNMi 13-10 を実行していますが，データベースが同期していないため，まだ個別に動作しています。

13. 以前のアクティブ NNMi 管理サーバーで次の手順を実行する。
   a ovstop コマンドを実行します。
   b 次のファイルを編集します。
   - Windows：%NNM_SHARED_CONF%¥props¥nms-cluster.properties
   - Linux：$NNM_SHARED_CONF/props/nms-cluster.properties
   c 13-00 からアップグレードした場合，com.hp.ov.nms.cluster.name パラメーターの値を入力します。
   13-00 からアップグレードした場合コメントにしたプロパティは保持されません。したがって，クラスタ名は再入力する必要があります。
   d com.hp.ov.nms.cluster.name パラメーターのコメントを解除します。
   e 変更を保存します。
14. ovstart コマンドまたはnnmcluster -daemon コマンドを以前のアクティブ NNMi 管理サーバーで実行する。これがアクティブなクラスタノードとなる。
15. アクティブなクラスタノードを使用してネットワークを監視するように，オペレータに指示する。
   以前のスタンバイ NNMi 管理サーバーは，手順 9.から手順 13.のメンテナンス中に発生したすべてのデータベースアクティビティを破棄します。
16. 以前のスタンバイ NNMi 管理サーバーで次の手順を実行する。
   a ovstop コマンドを実行します。
   b 次のファイルを編集します。
   - Windows：%NNM_SHARED_CONF%¥props¥nms-cluster.properties
   - Linux：$NNM_SHARED_CONF/props/nms-cluster.properties
   c 13-00 からアップグレードした場合，com.hp.ov.nms.cluster.name パラメーターの値を入力します。
   d com.hp.ov.nms.cluster.name パラメーターのコメントを解除します。
   e 変更を保存します。
17. ovstart コマンドまたはnnmcluster -daemon コマンドを以前のスタンバイ NNMi 管理サーバーで実行する。
   この NNMi 管理サーバーはスタンバイノードになり，アクティブなクラスタノードからデータベースのコピーを受信します。

(2) アプリケーションフェイルオーバー構成の NNMi 13-10 への修正パッチ適用手順

両方の NNMi 管理サーバーで同じバージョンとパッチレベルの NNMi を実行している必要があります。アクティブおよびスタンバイの NNMi 管理サーバーにパッチを追加するには，次のどちらかの方法を使用します。

- アプリケーションフェイルオーバー用にパッチを適用する（アクティブとスタンバイの両方をシャットダウン）

ネットワーク監視が中断されても問題にならない場合は，この手順を使用してください。

- アプリケーションフェイルオーバー用にパッチを適用する（1 つのアクティブ NNMi 管理サーバーを保持）

  ネットワーク監視の中断を回避する必要がある場合は，この手順を使用してください。

(a) アプリケーションフェイルオーバー用にパッチを適用する（アクティブとスタンバイの両方をシャットダウン）

この手順を実行すると，パッチプロセス中の一定期間，両方の NNMi 管理サーバーが非アクティブになります。アプリケーションフェイルオーバーを設定している NNMi 管理サーバーにパッチを適用するには，次の手順を実行します。

1. 万一に備えて，以降の操作を行う前に，アクティブおよびスタンバイ NNMi 管理サーバーの両方で，nnmconfigexport.ovpl スクリプトを実行する。

   詳細については，「4.2　ベストプラクティス：既存の設定を保存する」を参照してください。

2. 万一に備えて，以降の操作を行う前に，アクティブおよびスタンバイ NNMi 管理サーバーの NNMi データをバックアップする。

   詳細については，「20.2.2　バックアップ領域」を参照してください。

3. 万一に備えて，アクティブ NNMi 管理サーバーで，次の手順を実行する。

   a　nnmcluster コマンドを実行します。

   b　NNMi に入力を求められたら，「dbsync」と入力し，[Enter] キーを押します。表示される情報に次のメッセージが含まれていることを確認します。

   - ACTIVE_DB_BACKUP：アクティブ NNMi 管理サーバーが新しいバックアップを実行しています。
   - ACTIVE_NNM_RUNNING：アクティブ NNMi 管理サーバーが，前のメッセージによって示されたバックアップを完了しました。
   - STANDBY_READY：スタンバイ NNMi 管理サーバーの前のステータスを示します。
   - STANDBY_RECV_DBZIP：スタンバイ NNMi 管理サーバーは，アクティブ NNMi 管理サーバーから新しいバックアップを取得しています。
   - STANDBY_READY：スタンバイ NNMi 管理サーバーは，アクティブ NNMi 管理サーバーで障害が発生した場合に実行できる準備が整えられています。

   c　exit またはquit を実行して，手順 a で開始したインタラクティブnnmcluster プロセスを停止します。

4. アクティブ NNMi 管理サーバーでnnmcluster -halt コマンドを実行する。

   アクティブおよびスタンバイ NNMi 管理サーバーのすべてのnnmcluster プロセスをシャットダウンします。

5. 両方のサーバーでnnmcluster ノードが実行していないことを確認するには，アクティブおよびスタンバイ NNMi 管理サーバーの両方で次の手順を実行する。

   a　nnmcluster コマンドを実行します。

   b　(SELF)とマークされているもの以外にnnmcluster ノードが存在しないことを確認します。

c　exit またはquit を実行して，手順 a で開始したインタラクティブnnmcluster プロセスを停止します。

6. アクティブ NNMi 管理サーバーで，nms-cluster.properties ファイルのcom.hp.ov.nms.cluster.name パラメーターをコメントにする。

a　次のファイルを編集します。

- Windows：%NNM_SHARED_CONF%¥props¥nms-cluster.properties
- Linux：$NNM_SHARED_CONF/props/nms-cluster.properties

b　com.hp.ov.nms.cluster.name パラメーターをコメントにします。

c　変更を保存します。

7. パッチに同梱されている RELEASE.TXT の指示に従い，アクティブ NNMi 管理サーバーに NNMi パッチを適用する。

8. アクティブ NNMi 管理サーバーで，nms-cluster.properties ファイルのcom.hp.ov.nms.cluster.name パラメーターのコメントを解除する。

a　次のファイルを編集します。

- Windows：%NNM_SHARED_CONF%¥props¥nms-cluster.properties
- Linux：$NNM_SHARED_CONF/props/nms-cluster.properties

b　com.hp.ov.nms.cluster.name パラメーターのコメントを解除します。

c　変更を保存します。

9. アクティブ NNMi 管理サーバーでovstart コマンドを実行する。

10. NNMi コンソールの［ヘルプ］＞［システム情報］ウィンドウにある［製品］タブで情報を表示し，アクティブ NNMi 管理サーバーにパッチが正しくインストールされたことを確認する。

11. nnmcluster -dbsync コマンドを実行して，新しいバックアップを作成する。

12. 手順 6.の a～c に示されているように，スタンバイ NNMi 管理サーバーで，nms-cluster.properties ファイルのcom.hp.ov.nms.cluster.name パラメーターをコメントにする。

13. NNMi パッチをスタンバイ NNMi 管理サーバーに適用する。

14. 手順 8.の a～c に示されているように，スタンバイ NNMi 管理サーバーで，nms-cluster.properties ファイルのcom.hp.ov.nms.cluster.name パラメーターのコメントを解除する。

15. スタンバイ NNMi 管理サーバーでovstart コマンドを実行する。

(b)　アプリケーションフェイルオーバー用にパッチを適用する（1 つのアクティブ NNMi 管理サーバーを保持）

この手順を実行すると，パッチプロセスの間，1 つの NNMi 管理サーバーが常にアクティブになります。

このプロセスでは，ネットワークが継続的に監視されますが，NNMi でパッチプロセス中に生じたトランザクションログは失われます。

アプリケーションフェイルオーバーを設定しているNNMi管理サーバーにNNMiパッチを適用するには，次の手順を実行します。

1. 万一に備えて，以降の操作を行う前に，アクティブおよびスタンバイNNMi管理サーバーの両方で，nnmconfigexport.ovplスクリプトを実行する。

   詳細については，「4.2　ベストプラクティス：既存の設定を保存する」を参照してください。

2. 万一に備えて，以降の操作を行う前に，アクティブおよびスタンバイNNMi管理サーバーのNNMiデータをバックアップする。

   詳細については，「20.2.2　バックアップ領域」を参照してください。

3. ノードのどれかでnnmclusterコマンドを実行する。

4. 前の手順で2つのデータベースの同期に使用したNNMi管理サーバーでdbsyncを入力する。

   dbsyncオプションは，組み込みデータベースを使用するNNMi管理サーバーで機能します。

5. アクティブNNMi管理サーバーがACTIVE_NNM_RUNNINGに戻り，スタンバイNNMi管理サーバーがSTANDBY_READYに戻るまで待機してから，次に進む。

6. nnmclusterを終了または中断させる。

7. 次のコマンドをスタンバイNNMi管理サーバーで実行して，スタンバイNNMi管理サーバーのクラスタを停止する。

```
nnmcluster -shutdown
```

8. 次のプロセスとサービスが終了しているのを確認してから，次に進む。

```
postgres
ovjboss
```

9. nnmclusterプロセスが終了しているのを確認してから，次に進む。

   nnmclusterプロセスが終了していない場合，ほかに方法がなければ，nnmclusterプロセスを手動で強制終了します。

10. スタンバイNNMi管理サーバーで，次のファイルを編集する。

    - Windows：%nnmDataDir%shared¥nnm¥conf¥props¥nms-cluster.properties
    - Linux：$nnmDataDir/shared/nnm/conf/props/nms-cluster.properties

11. 行の先頭に#を入れてクラスタ名をコメントにして変更を保存する。

```
#com.hp.ov.nms.cluster.name = NNMicluster
```

12. スタンバイNNMi管理サーバーにNNMiパッチをインストールする。

13. この時点で，スタンバイNNMi管理サーバーはパッチが適用済みで停止中，アクティブNNMi管理サーバーはパッチが未適用で実行中である。

    アクティブNNMi管理サーバーを停止し，ただちにスタンバイNNMi管理サーバーを起動してネットワークを監視させます。

14. アクティブ NNMi 管理サーバーで次のコマンドを実行して，アクティブ NNMi 管理サーバーのクラスタをシャットダウンする。

```
nnmcluster -halt
```

15. nnmcluster プロセスの終了を確認する。

    数分以内に終了しない場合は，nnmcluster プロセスを手動で終了してください。

16. スタンバイ NNMi 管理サーバーで，nms-cluster.properties ファイルからクラスタ名のコメントを解除する。

17. 次のコマンドをスタンバイ NNMi 管理サーバーで実行して，スタンバイ NNMi 管理サーバーのクラスタを起動する。

```
nnmcluster -daemon
```

18. アクティブ NNMi 管理サーバーに NNMi パッチをインストールする。

19. この時点で，以前のアクティブ NNMi 管理サーバーはパッチが適用済みですが，オフラインである。

    次の手順を実行して，（スタンバイ NNMi 管理サーバーとして）クラスタに復帰させます。

    a　アクティブ NNMi 管理サーバーで，nms-cluster.properties ファイルのエントリのコメントを解除します。

    b　次のコマンドを使用して，アクティブ NNMi 管理サーバーを起動します。

```
nnmcluster -daemon
```

20. 進行状況を監視するには，アクティブとスタンバイの両方の NNMi 管理サーバーで次のコマンドを実行する。

```
nnmcluster
```

    以前のアクティブ NNMi 管理サーバーが，以前のスタンバイ NNMi 管理サーバーからデータベースの取得を完了するまで待機します。

21. 以前のアクティブ NNMi 管理サーバーにSTANDBY_READY が表示されたら，以前のアクティブ NNMi 管理サーバーで次のコマンドを実行する。

```
nnmcluster -acquire
```

25 NNMi Northbound インタフェース

NNMi には，NNMi Northbound インタフェースが用意されています。NNMi Northbound インタフェースを使用すると，SNMPv2c トラップを受信できるアプリケーションに NNMi インシデントを転送できます。各 NNMi 管理サーバーに，別々に設定された複数の NNMi Northbound インタフェースを実装できます。この章では，NNMi インシデントを任意の Northbound アプリケーションに転送するように NNMi を設定する方法を説明します。特定の Northbound アプリケーションの詳細については，アプリケーションのマニュアルを参照してください。なお，異なる Northbound アプリケーションとの統合についても，記載されています。

25.1 NNMi Northbound インタフェースの概要

NNMi Northbound インタフェースの概要を次に示します。

- NNMi 管理イベントを SNMPv2c トラップとして Northbound アプリケーションに転送します。

  Northbound アプリケーションは，NNMi トラップをフィルタリング，処理，および表示します。Northbound アプリケーションには，NNMi トラップのコンテキストで NNMi コンソールにアクセスするツールも用意されています。
- インシデントライフサイクルの状態変更通知，インシデント相関処理通知，およびインシデント削除通知を Northbound アプリケーションに送信できます。

  このように，Northbound アプリケーションは NNMi の因果関係分析の結果を複製できます。
- NNMi が受信する SNMP トラップを Northbound アプリケーションに転送することもできます。
- サードパーティまたはカスタムイベント統合アプリケーションでイベント統合を実行できます。
- そのほかのアプリケーションと NNMi の統合に使用できる情報でイベントを強化します。

この章では，次の用語を使用します。

- Northbound アプリケーション：SNMPv2c トラップを受信および処理できる任意のアプリケーションです。
- トラップ受信コンポーネント：SNMP トラップを受信する，Northbound アプリケーションの一部分です。

  一部のアプリケーションには，SNMP トラップを受信して処理用に別のコンポーネントに転送する，個別にインストール可能なコンポーネントが含まれます。

  そのようなコンポーネントがない Northbound アプリケーションの場合，「トラップ受信コンポーネント」は「Northbound アプリケーション」と同義語です。
- NNMi Northbound インタフェース：NNMi インシデントを SNMPv2c トラップとして Northbound アプリケーションに転送する NNMi の機能です。
- Northbound 転送先：Northbound アプリケーションのトラップ受信コンポーネントへの接続を定義し，NNMi がその Northbound アプリケーションに送信するトラップのタイプを指定する NNMi Northbound インタフェースの設定の 1 つです。

25.2 NNMi Northbound インタフェースの有効化

NNMi は，UDP を使用して SNMP トラップで送信される情報の量を制限しません。トラップデータのサイズが大きくて処理できないネットワークハードウェアが伝送経路上にあったり，ネットワークトラフィックの量が多かったりすると，トラップが失われることがあります。そのため，Northbound アプリケーションのトラップ受信コンポーネントを NNMi 管理サーバーにインストールすることをお勧めします。Northbound アプリケーションは，信頼性のある情報を転送する役割を担います。

NNMi Northbound インタフェースを有効にするには，次の手順を実行します。

1. 必要に応じて，NNMi トラップ定義を認識できるように Northbound アプリケーションを設定する。
2. NNMi 管理サーバーで，NNMi インシデント転送を設定する。

   a　NNMi コンソールで，［**HP NNMi-Northbound インタフェースデスティネーション**］フォーム（［**統合モジュールの設定**］＞［**Northbound インタフェース**］）を開き，［**新規作成**］をクリックします。

   使用できる転送先を選択してある場合，［**リセット**］をクリックして，［**新規作成**］ボタンを使用できるようにしてください。

   b　［**有効にする**］チェックボックスをオンにし，フォームの残りのフィールドを入力できるようにします。

   c　Northbound アプリケーションへの接続情報を入力します。

   これらのフィールドの詳細は，「25.8.1　NNMi Northbound アプリケーションの接続パラメーター」を参照してください。

   d　送信オプションおよび Northbound アプリケーションに送信する内容に対するインシデントフィルターを指定します。

   これらのフィールドの詳細は，「25.8.2　NNMi Northbound インタフェース統合の内容」を参照してください。

   e　フォームの下部にある［**送信**］をクリックします。

   新しいウィンドウが開き，ステータスメッセージが表示されます。設定に問題があることを示すメッセージが表示されたら，［**戻る**］をクリックして，エラーメッセージを参考に値を調整してください。

3. （任意）Northbound アプリケーションから NNMi ビューにアクセスするための URL を作成し，NNMi とのコンテキストインタラクションを作成する。

   NNMi は，UDP を使用して SNMP トラップで送信される情報の量を制限しません。トラップデータのサイズが大きくて処理不能なネットワークハードウェアが伝送経路上にあったり，ネットワークトラフィックの量が多かったりすると，トラップが失われることがあります。そのため，Northbound アプリケーションのトラップ受信コンポーネントを NNMi 管理サーバーにインストールすることをお勧めします。Northbound アプリケーションは，信頼性のある情報を転送する役割を担います。

   詳細については，NNMi コンソールで，［**ヘルプ**］＞［**NNMI ドキュメントライブラリ**］＞［**NNMi を別の場所で URL と統合**］をクリックしてください。

25.3　NNMi Northbound インタフェースの使用法

NNMi Northbound インタフェースを有効にすると，Northbound 転送先によって NNMi が Northbound アプリケーションに送信する情報が決まります。Northbound アプリケーションを設定して，転送されるトラップがネットワーク環境に応じて表示および解釈されるようにします。NNMi が Northbound アプリケーションに送信するトラップの内容および形式の詳細については，hp-nnmi-nbi.mib および hp-nnmi-registrations.mib ファイルを参照してください。

NNMi は，各管理イベント，SNMP トラップ，または通知トラップのコピーを 1 つだけ Northbound 転送先に送信します。NNMi はトラップをキューに入れません。NNMi がトラップを転送するときに Northbound アプリケーションのトラップ受信コンポーネントに接続できないと，トラップは失われます。

このセクションでは，統合で送信できるトラップのタイプを説明します。コンテンツ設定の詳細については，「25.8.2　NNMi Northbound インタフェース統合の内容」を参照してください。

25.3.1　インシデント転送

(1)　管理イベント

Northbound に管理イベントが含まれる場合，そのインシデントのライフサイクル状態が［登録済み］に変更されると，NNMi は各管理イベントを Northbound アプリケーションに転送します。

転送される管理イベントの OID は，NNMi コンソールの［管理イベントの設定］フォームに表示される SNMP オブジェクト ID です。NNMi は，OID が 1.3.6.1.4.1.11.2.17.19.2.0.9999 のすべてのカスタム管理イベントを転送します。

(2)　サードパーティ SNMP トラップ

Northbound 転送先にサードパーティの SNMP トラップが含まれる場合，関連インシデントのライフサイクル状態が［登録済み］に変更されると，NNMi は SNMPv1，v2c，または v3 形式の各受信ラップを Northbound アプリケーションに転送します。NNMi は，MIB で定義される元のトラップ varbind の順序を維持し，メッセージペイロードに NNMi 固有の varbind を追加します。元のトラップに含まれていない定義済み varbind がある場合，NNMi は，その欠落している varbind の部分に NULL 値を付与します。MIB が NNMi にロードされていない場合，NNMi 固有の varbind だけがトラップに追加され，次にこのトラップが転送されます。

サードパーティの SNMP トラップの場合は，次の点に注意してください。

- NNMi は SNMP トラップインシデントからのトラップを再構成するため，転送されるトラップの形式は，NNMi が受信した元のトラップの形式に関係なく，SNMPv2c となります。
- 転送される SNMP トラップは，NNMi 管理サーバーをソースオブジェクトとして示します。元のソースオブジェクトを判断するには，（n ＋ 21）番目の varbind の値 nnmiIncidentSourceNodeHostname

（1.3.6.1.4.1.11.2.17.19.2.2.21）と，（n ＋ 24）番目の varbind の値 nnmiIncidentSourceNodeMgmtAddr（1.3.6.1.4.1.11.2.17.19.2.2.24）を調べてください。n は MIB でトラップに定義されている varbind の数です。

NNMi が管理するデバイスのどれかが Northbound アプリケーションにトラップを送信する場合，Northbound アプリケーションで重複デバイストラップを管理する必要があります。

トラップ転送メカニズムの比較については，「8.1.2　トラップおよびインシデント転送」を参照してください。

25.3.2　インシデントライフサイクル状態変化通知

このセクションの情報は，［NNMi-Northbound インタフェースデスティネーション］ページの［送信オプション］の選択によって異なります。

(1)　エンハンスド解決済みしたトラップ

Northbound 転送先にエンハンスド解決済み通知が含まれる場合，NNMi のインシデントのライフサイクル状態が［解決済み］に変化したときに，NNMi は nnmiEvClosed（1.3.6.1.4.1.11.2.17.19.2.0.1000）トラップを Northbound アプリケーションに転送します。nnmiEvClosed トラップは，元のインシデントのデータの多くを含んでいます。前のライフサイクル状態の値は含んでいません。

nnmiEvClosed トラップは，6 番目の varbind である nnmiIncidentUuid（1.3.6.1.4.1.11.2.17.19.2.2.6）で元のインシデントを識別します。

(2)　状態変化トラップ

Northbound 転送先にライフサイクル状態変更通知が含まれる場合，NNMi のインシデントのライフサイクル状態が［進行中］，［完了］，または［解決済み］に変化したときに，NNMi は nnmiEvLifecycleStateChanged（1.3.6.1.4.1.11.2.17.19.2.0.1001）トラップを Northbound アプリケーションに送信します。Northbound アプリケーションは，nnmiEvLifecycleStateChanged と元のインシデントを関連づけできます。

nnmiEvLifecycleStateChanged トラップは，次の varbind で元のインシデントとライフサイクル状態の変化を識別します。

- nnmiIncidentUuid，6 番目の varbind（1.3.6.1.4.1.11.2.17.19.2.2.6）
  この値は，管理イベントの 6 番目の varbind の値，またはサードパーティ SNMP トラップ varbind の（n ＋ 6）番目の varbind の値と一致します。
- nnmiIncidentLifecycleStatePreviousValue，7 番目の varbind（1.3.6.1.4.1.11.2.17.19.2.2.200）
- nnmiIncidentLifecycleStateCurrentValue，8 番目の varbind（1.3.6.1.4.1.11.2.17.19.2.2.201）

次の表は，ライフサイクル状態に使用できる整数値を示したものです。

名前	整数値
登録済み	1
進行中	2
完了	3
解決済み	4
抑止済み	5

25.3.3　インシデント相関処理通知

Northbound 転送先にインシデント相関処理通知が含まれる場合，NNMi の因果関係分析でインシデントが相関処理されると，NNMi はインシデント相関処理トラップを Northbound アプリケーションに送信します。Northbound アプリケーションはトラップ内の情報を使用して相関変更を複製できます。

(1)　単一相関トラップ

単一相関トラップオプションの場合，この統合では，次の相関トラップを送信します。

- nnmiEvCorrelationDedup（1.3.6.1.4.1.11.2.17.19.2.0.1100）
- nnmiEvCorrelationImpact（1.3.6.1.4.1.11.2.17.19.2.0.1101）
- nnmiEvCorrelationPairwise（1.3.6.1.4.1.11.2.17.19.2.0.1102）
- nnmiEvCorrelationRate（1.3.6.1.4.1.11.2.17.19.2.0.1103）
- nnmiEvCorrelationApa（1.3.6.1.4.1.11.2.17.19.2.0.1104）
- nnmiEvCorrelationCustom（1.3.6.1.4.1.11.2.17.19.2.0.1105）

各トラップは，次の varbind で，1 つの親子インシデント相関関係を示します。

- nnmiIncidentUuid，6 番目の varbind（1.3.6.1.4.1.11.2.17.19.2.2.6）
- nnmiCorrelatedChildUuid，7 番目の varbind（1.3.6.1.4.1.11.2.17.19.2.2.300）

(2)　グループ相関トラップ

グループ相関トラップオプションの場合，この統合では，次の相関トラップを送信します。

- nnmiEvCorrelationGrpDedup（1.3.6.1.4.1.11.2.17.19.2.0.2100）
- nnmiEvCorrelationGrpImpact（1.3.6.1.4.1.11.2.17.19.2.0.2101）
- nnmiEvCorrelationGrpPairwise（1.3.6.1.4.1.11.2.17.19.2.0.2102）
- nnmiEvCorrelationGrpRate（1.3.6.1.4.1.11.2.17.19.2.0.2103）
- nnmiEvCorrelationGrpApa（1.3.6.1.4.1.11.2.17.19.2.0.2104）

- nnmiEvCorrelationGrpCustom（1.3.6.1.4.1.11.2.17.19.2.0.2105）

各トラップは，次の varbind で，親子インシデント相関関係を示します。

- nnmiIncidentUuid，6 番目の varbind（1.3.6.1.4.1.11.2.17.19.2.2.6）
- nnmiCorrelatedChildrenCount，7 番目の varbind（1.3.6.1.4.1.11.2.17.19.2.2.301）
- nnmiCorrelatedChildrenUuidCsv，8 番目の varbind（1.3.6.1.4.1.11.2.17.19.2.2.302）
  この値は子インシデント UUID のコンマ区切りリストです。

25.3.4　インシデント削除通知

Northbound 転送先にインシデント削除通知が含まれる場合，インシデントが NNMi で削除されると，NNMi は nnmiEvDeleted（1.3.6.1.4.1.11.2.17.19.2.0.3000）トラップを Northbound アプリケーションに送信します。nnmiEvDeleted トラップは，6 番目の varbind である nnmiIncidentUuid（1.3.6.1.4.1.11.2.17.19.2.2.6）で元のインシデントを識別します。

25.3.5　イベント転送フィルター

Northbound 転送先にインシデントフィルターが含まれる場合，選択した設定オプションに応じて，フィルターのオブジェクト ID（OID）には，次のイベントタイプが包含または除外されます。

- NNMi 管理イベントインシデント
- サードパーティ SNMP トラップ
- nnmiEvClosed トラップ
- nnmiEvLifecycleStateChanged トラップ
- nnmiEvDeleted トラップ
- 相関関係通知トラップ ※

注※　相関関係通知トラップについて次の注意が必要です。

- インシデントフィルターが相関処理に親インシデントを転送しない場合，NNMi は相関関係通知トラップを Northbound アプリケーションに送信しません。
- インシデントフィルターが相関処理に子インシデントを転送しない場合，転送される相関関係通知トラップにその子インシデントの UUID は含まれません。つまり，相関関係通知トラップに子インシデント UUID が含まれない場合，NNMi はそのトラップを Northbound アプリケーションに送信しません。
- DuplicateCorrelation 管理イベントは，nnmiEvCorrelationDedup または nnmiEvCorrelationGrpDedup 相関関係通知トラップとは無関係に転送されます。同様に，RateCorrelation 管理イベントは nnmiEvCorrelationRate または nnmiEvCorrelationGrpRate 相

関関係通知トラップとは無関係に転送されます。インシデントフィルターがこれらの相関関係通知トラップのどれかを転送しない場合でも，NNMi によって関連管理イベントが転送される場合があります。

25.4　NNMi Northbound インタフェースの変更

NNMi Northbound インタフェースの設定パラメーターを変更するには，次の手順を実行します。

1. NNMi コンソールで，［NNMi-Northbound インタフェースデスティネーション］フォーム（［統合モジュールの設定］ ＞ ［Northbound インタフェース］）を開く。
2. 転送先を選択し，［編集］をクリックする。
3. 該当するように値を変更する。

   このフォームのフィールドの詳細は，「25.8　［NNMi-Northbound インタフェースデスティネーション］フォームのリファレンス」を参照してください。
4. フォームの上端の［有効にする］チェックボックスがオンであることを確認し，フォームの下端の［送信］をクリックする。

   変更は直ちに有効になります。

25.5 NNMi Northbound インタフェースの無効化

Northbound 転送先が無効な間は，SNMP トラップはキューイングされません。

Northbound アプリケーションへの NNMi の転送を中止するには，次の手順を実行します。

1. NNMi コンソールで，［NNMi-Northbound インタフェースデスティネーション］フォーム（［統合モジュールの設定］ ＞ ［Northbound インタフェース］）を開く。
2. 転送先を選択し，［編集］をクリックする。または，［削除］をクリックして，選択した転送先の設定をすべて削除する。
3. フォームの上端の［有効にする］チェックボックスをオフにし，フォームの下端の［送信］をクリックする。

   変更は直ちに有効になります。

25.6 NNMi Northbound インタフェースのトラブルシューティング

NNMi Northbound インタフェースが正常に機能しない場合は，次の手順を実行して問題を解決してください。

1. トラップ転送先ポートがファイアウォールによってブロックされていないことを確認する。

   NNMi 管理サーバーが，ホストとポートによって Northbound アプリケーションを直接処理できることを確認します。

2. 統合が正常に実行されていることを確認する。

   a NNMi コンソールで，［NNMi-Northbound インタフェースデスティネーション］フォーム（［統合モジュールの設定］ > ［Northbound インタフェース］）を開きます。

   b 転送先を選択し，［編集］をクリックします。

   c ［有効にする］オプションが選択されていることを確認します。

3. Northbound 転送先に管理イベントが含まれる場合は，この機能を確認する。

   a NNMi コンソールの［解決済みの重要なインシデント］ビューで，任意のインシデントを開きます。

   b インシデントライフサイクル状態を［登録済み］に設定して，［保存］をクリックします。

   c インシデントライフサイクル状態を［解決済み］に設定して，［保存して閉じる］をクリックします。

   d 30 秒後，Northbound アプリケーションがこのインシデントの nnmiEvClosed トラップ（または nnmiEvLifecycleStateChanged トラップ）を受信したかどうかを確認します。

   - Northbound アプリケーションがトラップを受信した場合は，手順 4.を続行します。
   - Northbound アプリケーションがトラップを受信しなかった場合は，異なる Northbound アプリケーションに接続する新規 Northbound 転送先を設定してから，手順 a からこのテストを繰り返します。

     再テストに合格した場合，問題は最初の Northbound アプリケーションにあります。アプリケーションのドキュメントでトラブルシューティング情報を参照してください。再テストに不合格になった場合は，サポートサービスに問い合わせてください。

4. Northbound 転送先に SNMP トラップが含まれる場合は，この機能を確認する。

   a NNMi 管理サーバーで次のコマンドを入力することで，NNMi トポロジ内のノードに対する SNMP トラップを生成します。

   ```
   nnmsnmpnotify.ovpl -a ¥
   discovered_node NNMi_node .1.3.6.1.6.3.1.1.5.1
   ```

   `discovered_node` は，NNMi トポロジのノードのホスト名または IP アドレスです。`NNMi_node` は，NNMi 管理サーバーのホスト名または IP アドレスです。

   b 30 秒後に，Northbound アプリケーションが転送されたトラップを受信したかどうかを確認します。

   - Northbound アプリケーションがトラップを受信した場合，NNMi Northbound インタフェースは正常に機能しています。

- Northbound アプリケーションがトラップを受信しなかった場合は，異なる Northbound アプリケーションに接続する新規 Northbound 転送先を設定してから，手順 a からこのテストを繰り返します。

再テストに合格した場合，問題は最初の Northbound アプリケーションにあります。アプリケーションのドキュメントでトラブルシューティング情報を参照してください。再テストに不合格になった場合は，サポートサービスに問い合わせてください。

25.7 アプリケーションフェイルオーバーと NNMi Northbound インタフェース

NNMi 管理サーバーが NNMi アプリケーションフェイルオーバーに関係することになる場合，ここでの情報は，Northbound レシーバーにトラップを送信する NNMi Northbound アプリケーションを実装するすべての統合に適用されます。

NNMi が Northbound アプリケーションに送信するトラップには，NmsUrl varbind（1.3.6.1.4.1.11.2.17.19.2.2.2）の NNMi URL が含まれます。アプリケーションフェイルオーバー前に受信したトラップは，現在のスタンバイ NNMi 管理サーバーを参照します。

URL がスタンドバイ NNMi 管理サーバーを指す場合，その URL 値を使用するすべてのアクション（例えば，NNMi コンソールの起動）は失敗します。

25.7.1 ローカル Northbound アプリケーション

Northbound アプリケーションのトラップ受信コンポーネントが NNMi 管理サーバー上にある場合は，次のことが NNMi Northbound インタフェースの設定に適用されます。

- Northbound アプリケーションのトラップ受信コンポーネントは，アクティブおよびスタンバイ NNMi 管理サーバーに同じようにインストールおよび設定する必要があります。両方の NNMi 管理サーバーの同じポートで SNMP トラップ受信を設定します。
- プライマリ NNMi 管理サーバーだけで NNMi Northbound インタフェースを設定します。
  ［NNMi-Northbound インタフェースデスティネーション］フォームの**［ホスト］**識別で，**［NNMi FQDN］**または**［ループバックを使用］**オプションを選択します。

NNMi Northbound インタフェースは，起動時に，現在の NNMi 管理サーバーの正しい名前または IP アドレスを判断します。このように，Northbound インタフェースは，トラップをアクティブな NNMi 管理サーバー上の Northbound アプリケーションのトラップ受信コンポーネントに送信します。

25.7.2 リモート Northbound アプリケーション

Northbound アプリケーションのトラップ受信コンポーネントが NNMi 管理サーバー上にない場合は，NNMi Northbound インタフェースをプライマリ NNMi 管理サーバーだけで設定します。**［NNMi-Northbound インタフェースデスティネーション］**フォームの**［ホスト］**識別で，**［その他］**オプションを選択します。

25.8 ［NNMi-Northbound インタフェースデスティネーション］フォームのリファレンス

［HP NNMi-Northbound インタフェースデスティネーション］フォームには，NNMi と Northbound アプリケーション間の通信設定パラメーターがあります。このフォームは，［統合モジュールの設定］ワークスペースから使用できます。［NNMi-Northbound インタフェースデスティネーション］フォームで，［新規作成］をクリックするか，または転送先を選択して，［編集］をクリックします。

- Administrator ロールの NNMi ユーザーだけが［NNMi-Northbound インタフェースデスティネーション］フォームにアクセスできます。

  ［NNMi-Northbound インタフェースデスティネーション］フォームには，次の領域の情報が表示されます。

  - 「25.8.1　NNMi Northbound アプリケーションの接続パラメーター」
  - 「25.8.2　NNMi Northbound インタフェース統合の内容」
  - 「25.8.3　NNMi Northbound インタフェース転送先のステータス情報」

統合設定に変更を適用するには，［NNMi-Northbound インタフェースデスティネーション］フォームの値を更新し，［送信］をクリックします。

25.8.1　NNMi Northbound アプリケーションの接続パラメーター

次の表は，NNMi Northbound アプリケーションへの接続設定用パラメーターを示したものです。

表 25-1　NNMi Northbound アプリケーションの接続情報

フィールド	説明
ホスト	Northbound アプリケーションのトラップ受信コンポーネントを含むサーバーの完全修飾ドメイン名（推奨）または IP アドレス。 統合では，次のサーバーの識別方法がサポートされています。 • NNMi FQDN NNMi が NNMi 管理サーバー上の Northbound アプリケーションへの接続を管理し，［ホスト］フィールドが読み取り専用になります。 これが，NNMi 管理サーバー上での Northbound アプリケーションの推奨設定です。 • ループバックを使用 NNMi が NNMi 管理サーバー上の Northbound アプリケーションへの接続を管理し，［ホスト］フィールドが読み取り専用になります。 • その他 Northbound アプリケーションサーバーを識別するホスト名または IP アドレスを，［ホスト］フィールドに入力します。 NNMi は，［ホスト］フィールドのホスト名または IP アドレスがループバックアダプターとして設定されていないことを確認します。

フィールド	説明
ホスト	これがデフォルト設定です。 注　NNMi 管理サーバーが NNMi アプリケーションフェイルオーバーに参加する場合にアプリケーションフェイルオーバーが統合に与える影響については，「25.7　アプリケーションフェイルオーバーと NNMi Northbound インタフェース」を参照してください。
ポート	Northbound アプリケーションが SNMP トラップを受信する UDP ポート。 Northbound アプリケーション固有のポート番号を入力します。 注　Northbound アプリケーションのトラップ受信コンポーネントが NNMi 管理サーバー上にある場合，このポート番号は，NNMi コンソールの［通信の設定］フォームの［SNMP ポート］フィールドで設定した，NNMi が SNMP トラップを受信するために使用するポートと別にする必要があります。
コミュニティ文字列	トラップを受信する Northbound アプリケーションの読み取り専用コミュニティ文字列。 Northbound アプリケーション設定で，受信した SNMP トラップにコミュニティ文字列が必要な場合は，その値を入力します。 Northbound アプリケーション設定で，特定のコミュニティ文字列が不要な場合は，デフォルト値の public を使用します。

25.8.2　NNMi Northbound インタフェース統合の内容

NNMi Northbound インタフェースが Northbound アプリケーションに送信する内容を設定するためのパラメーターを次の表に示します。

表 25-2　NNMi Northbound インタフェースの内容設定情報

フィールド	説明
インシデント	インシデント転送の指定。 • 管理 NNMi は，NNMi が生成した管理イベントだけを Northbound アプリケーションに転送します。 • サードパーティ SNMP トラップ NNMi は，NNMi が管理対象デバイスから受信する SNMP トラップだけを Northbound アプリケーションに転送します。 • Syslog NNMi は，NNMi が管理対象デバイスから受信する ArcSight Syslog メッセージだけを Northbound 統合モジュールを使用して Northbound アプリケーションに転送します。 NNMi は，Northbound 転送先を有効にすると直ちにインシデントの転送を開始します。詳細については，「25.3.1　インシデント転送」を参照してください。

フィールド	説明
ライフサイクル状態の変化	インシデント変更通知の仕様。 • エンハンスド解決済み NNMi は，ライフサイクル状態が［解決済み］に変化したインシデントごとに，インシデント解決済みトラップを Northbound アプリケーションに送信します。 これがデフォルト設定です。 • 変化した状態 NNMi は，ライフサイクル状態が［進行中］，［完了］，または［解決済み］に変化したインシデントごとに，インシデントのライフサイクル状態変化トラップを Northbound アプリケーションに送信します。 • 両方 NNMi は，ライフサイクル状態が［解決済み］に変化したインシデントごとに，インシデント解決済みトラップを Northbound アプリケーションに送信します。また，この統合では，ライフサイクル状態が［進行中］，［完了］，または［解決済み］に変化したインシデントごとに，インシデントのライフサイクル状態変化トラップを Northbound アプリケーションに送信します。 注　この場合，インシデントが［解決済み］ライフサイクル状態に変化するたびに，インシデント解決済みトラップとインシデントライフサイクル状態変更トラップの 2 つの通知トラップが統合によって送信されます。 詳細については，「25.3.2　インシデントライフサイクル状態変化通知」を参照してください。
相関処理	インシデント相関処理通知の仕様。 • なし NNMi は，NNMi 因果関係分析によるインシデント相関処理結果を Northbound アプリケーションに通知しません。 これがデフォルト設定です。 • 単一 NNMi は，NNMi 因果関係分析で判明した親子インシデント相関関係ごとにトラップを 1 つ送信します。 • グループ NNMi は，親インシデントに相関するすべての子インシデントをリストした相関処理ごとに，トラップを 1 つ送信します。 詳細については，「25.3.3　インシデント相関処理通知」を参照してください。
削除	インシデント削除の仕様。このセクションは，［インシデント］フィールドでの選択内容に対して，削除トラップを Northbound アプリケーションに送信するかどうかを設定します。 • 送信しない NNMi は，インシデントが NNMi で削除されても Northbound アプリケーションに通知しません。 これがデフォルト設定です。 • 送信

フィールド	説明
削除	NNMi は，NNMi で削除されるインシデントごとに，削除トラップを Northbound アプリケーションに送信します。 詳細については，「25.3.4　インシデント削除通知」を参照してください。
NNMi コンソールアクセス	Northbound アプリケーションから NNMi コンソールを参照する URL の接続プロトコル仕様。NNMi が Northbound アプリケーションに送信するトラップの NmsUrl varbind（1.3.6.1.4.1.11.2.17.19.2.2.2）には，NNMi URL が含まれます。 設定ページのデフォルトは，NNMi 設定と一致する設定になります。 NNMi コンソールが HTTP と HTTPS 両方の接続を承認するよう設定されている場合，NNMi URL で HTTP 接続プロトコルの指定を変更できます。例えば，Northbound アプリケーションのすべてのユーザーがイントラネット上にある場合は，Northbound アプリケーションから NNMi コンソールへのアクセスを HTTP 経由に設定できます。 Northbound アプリケーションから NNMi コンソールに接続するプロトコルを変更する場合は，必要に応じて，［HTTP］オプションまたは［HTTPS］オプションを選択します。
Incident Filter（インシデントフィルター）	Northbound アプリケーションに送信されたイベントをフィルターするために統合で使用されるオブジェクト ID（OID）のリスト。各フィルターエントリーは，有効な数値 OID（例えば，.1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.9）または OID プレフィックス（例えば，.1.3.6.1.6.3.1.1.5.*）にすることができます。 次のオプションの 1 つを選択します。 • なし NNMi はすべてのイベントを Northbound アプリケーションに送信します。 これがデフォルト設定です。 • 含む NNMi は，フィルターで識別された OID と一致する特定のイベントだけを送信します。 • 除外する NNMi は，フィルターで識別された OID と一致する特定のイベントを除くすべてのイベントを送信します。 インシデントフィルターを指定します。 • フィルターエントリーを追加するには，下側のテキストボックスにテキストを入力してから，［追加］をクリックします。 • フィルターエントリーを削除するには，上側のボックスのリストからエントリーを選択して，［削除］をクリックします。 詳細については，「25.3.5　イベント転送フィルター」を参照してください。

25.8.3 NNMi Northbound インタフェース転送先のステータス情報

Northbound 転送先の読み取り専用ステータス情報を次の表に示します。この情報は，統合が現在機能しているか確認する場合に役立ちます。

表 25-3 NNMi Northbound インタフェース転送先のステータス情報

フィールド	説明
トラップ転送先 IP アドレス	転送先ホスト名の解決先となる IP アドレス。 この値は，このノースバウンド転送先に固有です。
アップタイム（秒）	Northbound コンポーネントが最後に起動されてからの時間（秒）。NNMi が Northbound アプリケーションに送信するトラップの sysUptime フィールド（1.3.6.1.2.1.1.3.0）にはこの値が含まれます。 この値は，NNMi Northbound インタフェースを使用するすべての統合に対して同じです。最新の値を表示するには，リフレッシュするか，フォームを閉じて再び開いてください。
NNMi URL	NNMi コンソールに接続するための URL。NNMi が Northbound アプリケーションに送信するトラップの NmsUrl varbind（1.3.6.1.4.1.11.2.17.19.2.2.2）にはこの値が含まれます。 この値は，このノースバウンド転送先に固有です。

25.8.4 NNMi Northbound インタフェースで使用される MIB 情報

特定の MIB を NNMi にロードし，NNMi Northbound 統合によって送信されるインシデント通知で使用される管理情報を表示するには，次の手順を実行します。

1. 次のディレクトリに移動する。
   - Windows：`%NnmInstallDir%misc¥nnm¥snmp-mibs¥Vendor¥Hewlett-Packard`
   - Linux：`/opt/OV/misc/nnm/snmp-mibs/Vendor/Hewlett-Packard`
2. 次のコマンドを実行して，`hp-nnmi.mib` ファイルをロードする。

```
nnmloadmib.ovpl -load hp-nnmi.mib
```

3. 次のコマンドを実行して，`hp-nnmi-registrations.mib` ファイルをロードする。

```
nnmloadmib.ovpl -load hp-nnmi-registrations.mib
```

4. 次のコマンドを実行して，`hp-nnmi-nbi.mib` ファイルをロードする。

```
nnmloadmib.ovpl -load hp-nnmi-nbi.mib
```

5. NNMi コンソールから，［設定］ワークスペースを開く。
6. ［MIB］ ＞ ［ロード済み MIB］をクリックします。
7. ロードした各 MIB をダブルクリックし，［MIB 変数］をクリックして MIB 情報を表示します。

25.8.5 NNMi Northbound インタフェースで使用される SNMP トラップ情報

Northbound インタフェースで使用される SNMP トラップについては，hp-nnmi-nbi.mib ファイルに定義されています。

NNMi を Northbound アプリケーションとして使用する場合は，次の手順を実行して SNMP トラップインシデントの定義を追加してください。

1. 「25.8.4 NNMi Northbound インタフェースで使用される MIB 情報」の手順 1.から手順 4.を実行する。
2. 次のコマンドを実行して，SNMP トラップインシデントの定義を追加する。

```
nnmincidentcfg.ovpl -loadTraps HP-NNMI-NBI-MIB
```

26 JP1/IM3 のインテリジェント統合管理基盤との連携

26.1　JP1/IM3 のインテリジェント統合管理基盤との連携

JP1/IM3 のインテリジェント統合管理基盤と連携することで，下記ができるようになります。

- NNMi が管理しているノードの情報を JP1/IM3 の統合オペレーション・ビューアーに表示できる
- JP1/IM3 の統合オペレーション・ビューアーから，NNMi のコンソール画面やノードフォーム，L2/L3 近接接続ビューなどを表示できる
- JP1/IM3 の対処アクション提案機能によって，JP1 イベント（NNMi インシデント）が発生したノードに対し，ステータスポーリングの実行やノードフォームの表示などの対処アクションを行うことができる
- JP1/IM3 のカスタム UI 連携機能によって，JP1/IM3 の統合オペレーション・ビューアー上で選択したノードのノードフォームや L2 近接接続ビューを表示できる

JP1/IM3 のインテリジェント統合管理基盤と連携するためには，JP1/IM3 用のプラグインをセットアップする必要があります。

なお，JP1/IM2 をご使用の場合は，上記の JP1/IM3 を JP1/IM2 に読み替えてください。

連携機能の詳細，およびセットアップ方法については，リリースノートを参照してください。

27 RESTful API

NNMi の RESTful API を用いて，NNMi とほかの製品を連携させることができます。

27.1 RESTful API

RESTful API では，NNMi のインシデント，ノード，IP アドレス，およびインタフェースの情報を取得したり，更新したりできます。

ほかの製品や Web ポータルなどから，RESTful API を使用して NNMi と連携することで，ほかの製品や Web ポータルで NNMi の情報を表示したり，更新したりできます。

RESTful API の詳細については，リリースノートを参照してください。

付録

付録 A　NNMi の man ページを表示できない場合（Linux）

NNMi 管理サーバーに NNMi の man ページを表示できない場合は，MANPATH 変数に/opt/OV/man の場所が含まれていることを確認します。含まれていない場合は，/opt/OV/man の場所をMANPATH 変数に追加します。

付録 B　新規インストール中に読み込む MIB 一覧

次の表は，NNMi が新規インストール中に読み込む MIB を一覧で示しています。

NNMi をバージョンアップした場合は読み込まれません。

なお，表に示す MIB ファイルは，次のパスからの相対パスになります。

- Windows：%NnmInstallDir%misc¥nnm¥snmp-mibs¥
  また，Windows の場合，パスの区切り文字が/ではなく¥になります。
- Linux：$NnmInstallDir/misc/nnm/snmp-mibs/

表 B-1　NNMi が新規インストール中に読み込む MIB

MIB 名	MIB ファイル
ATM-FORUM-MIB	Vendor/Cisco/ATM-FORUM-MIB.my
ATM-FORUM-TC-MIB	Vendor/Cisco/ATM-FORUM-TC-MIB.my
ATM-MIB	Standard/rfc2515-ATM-MIB.mib
ATM-TC-MIB	Standard/rfc2514-ATM-TC-MIB.mib
ATM2-MIB	Standard/rfc3606-ATM2-MIB.mib
ArcsightModule	Vendor/Hewlett-Packard/hp-arcsight.mib
BGP4-MIB	Standard/rfc4273-BGP4-MIB.mib
BRIDGE-MIB	Standard/rfc4188-BRIDGE-MIB.mib
CISCO-AAL5-MIB	Vendor/Cisco/CISCO-AAL5-MIB.my
CISCO-ATM-IF-MIB	Vendor/Cisco/CISCO-ATM-IF-MIB.my
CISCO-ATM-SWITCH-ADDR-MIB	Vendor/Cisco/CISCO-ATM-SWITCH-ADDR-MIB.my
CISCO-C2900-MIB	Vendor/Cisco/CISCO-C2900-MIB.my
CISCO-CDP-MIB	Vendor/Cisco/CISCO-CDP-MIB.my
CISCO-DOT11-ASSOCIATION-MIB	Vendor/Cisco/CISCO-DOT11-ASSOCIATION-MIB.my
CISCO-DOT11-IF-MIB	Vendor/Cisco/CISCO-DOT11-IF-MIB.my
CISCO-ENTITY-FRU-CONTROL-MIB	Vendor/Cisco/CISCO-ENTITY-FRU-CONTROL-MIB.my
CISCO-ENTITY-VENDORTYPE-OID-MIB	Vendor/Cisco/CISCO-ENTITY-VENDORTYPE-OID-MIB.my
CISCO-ENVMON-MIB	Vendor/Cisco/CISCO-ENVMON-MIB.my
CISCO-FLASH-MIB	Vendor/Cisco/CISCO-FLASH-MIB.my
CISCO-FRAME-RELAY-MIB	Vendor/Cisco/CISCO-FRAME-RELAY-MIB.my
CISCO-HSRP-MIB	Vendor/Cisco/CISCO-HSRP-MIB.my

MIB 名	MIB ファイル
CISCO-IETF-IP-MIB	Vendor/Cisco/CISCO-IETF-IP-MIB.my
CISCO-IETF-IPMROUTE-MIB	Vendor/Cisco/CISCO-IETF-IPMROUTE-MIB.my
CISCO-IETF-PIM-EXT-MIB	Vendor/Cisco/CISCO-IETF-PIM-EXT-MIB.my
CISCO-IETF-PIM-MIB	Vendor/Cisco/CISCO-IETF-PIM-MIB.my
CISCO-IETF-PW-ENET-MIB	Vendor/Cisco/CISCO-IETF-PW-ENET-MIB.my
CISCO-IETF-PW-MIB	Vendor/Cisco/CISCO-IETF-PW-MIB.my
CISCO-IETF-PW-MPLS-MIB	Vendor/Cisco/CISCO-IETF-PW-MPLS-MIB.my
CISCO-IETF-PW-TC-MIB	Vendor/Cisco/CISCO-IETF-PW-TC-MIB.my
CISCO-MEMORY-POOL-MIB	Vendor/Cisco/CISCO-MEMORY-POOL-MIB.my
CISCO-MVPN-MIB	Vendor/Cisco/CISCO-MVPN-MIB.my
CISCO-NBAR-PROTOCOL-DISCOVERY-MIB	Vendor/Cisco/CISCO-NBAR-PROTOCOL-DISCOVERY-MIB.my
CISCO-PIM-MIB	Vendor/Cisco/CISCO-PIM-MIB.my
CISCO-PRODUCTS-MIB	Vendor/Cisco/CISCO-PRODUCTS-MIB.my
CISCO-QOS-PIB-MIB	Vendor/Cisco/CISCO-QOS-PIB-MIB.my
CISCO-RF-MIB	Vendor/Cisco/CISCO-RF-MIB.my
CISCO-RHINO-MIB	Vendor/Cisco/CISCO-RHINO-MIB.my
CISCO-RTTMON-MIB	Vendor/Cisco/CISCO-RTTMON-MIB.my
CISCO-RTTMON-TC-MIB	Vendor/Cisco/CISCO-RTTMON-TC-MIB.my
CISCO-SMI	Vendor/Cisco/CISCO-SMI.my
CISCO-STACK-MIB	Vendor/Cisco/CISCO-STACK-MIB.my
CISCO-TC	Vendor/Cisco/CISCO-TC.my
CISCO-VTP-MIB	Vendor/Cisco/CISCO-VTP-MIB.my
CISCOWAN-SMI	Vendor/Cisco/CISCOWAN-SMI.my
DHCP-MIB	Vendor/Microsoft/dhcp.mib
DIFFSERV-DSCP-TC	Standard/rfc3289-DIFFSERV-DSCP-TC.mib
DIFFSERV-MIB	Standard/rfc3289-DIFFSERV-MIB.mib
DISMAN-NSLOOKUP-MIB	Standard/rfc4560-DISMAN-NSLOOKUP-MIB.mib
DISMAN-PING-MIB	Standard/rfc4560-DISMAN-PING-MIB.mib
DISMAN-TRACEROUTE-MIB	Standard/rfc4560-DISMAN-TRACEROUTE-MIB.mib
DRAFT-MSDP-MIB	Vendor/Cisco/MSDP-MIB.my
DS1-MIB	Standard/rfc4805-DS1-MIB.mib

MIB 名	MIB ファイル
DS3-MIB	Standard/rfc3896-DS3-MIB.mib
DVMRP-MIB	Vendor/Nortel/DVMRP-MIB.mib
ENTITY-MIB	Standard/rfc4133-ENTITY-MIB.mib
ENTITY-STATE-MIB	Standard/rfc4268-ENTITY-STATE-MIB.mib
ENTITY-STATE-TC-MIB	Standard/rfc4268-ENTITY-STATE-TC-MIB.mib
EXTREME-BASE-MIB	Vendor/Extreme/v730b49.mib
EXTREME-CABLE-MIB	Vendor/Extreme/v730b49.mib
EXTREME-DLCS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-DOS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-EAPS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-EDP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-ENH-DOS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-ESRP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-FDB-MIB	Vendor/Extreme/v730b49.mib
EXTREME-FILETRANSFER-MIB	Vendor/Extreme/v730b49.mib
EXTREME-NETFLOW-MIB	Vendor/Extreme/v730b49.mib
EXTREME-NP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-OSPF-MIB	Vendor/Extreme/v730b49.mib
EXTREME-PBQOS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-POE-MIB	Vendor/Extreme/v730b49.mib
EXTREME-PORT-MIB	Vendor/Extreme/v730b49.mib
EXTREME-POS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-QOS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-RTSTATS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-SERVICES-MIB	Vendor/Extreme/v730b49.mib
EXTREME-SLB-MIB	Vendor/Extreme/v730b49.mib
EXTREME-SNMPV3-MIB	Vendor/Extreme/v730b49.mib
EXTREME-STP-EXTENSIONS-MIB	Vendor/Extreme/v730b49.mib
EXTREME-SYSTEM-MIB	Vendor/Extreme/v730b49.mib
EXTREME-TRAP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-TRAPPOLL-MIB	Vendor/Extreme/v730b49.mib

MIB 名	MIB ファイル
EXTREME-V2TRAP-MIB	Vendor/Extreme/v730b49.mib
EXTREME-VC-MIB	Vendor/Extreme/v730b49.mib
EXTREME-VLAN-MIB	Vendor/Extreme/v730b49.mib
EXTREME-WIRELESS-MIB	Vendor/Extreme/v730b49.mib
EXTREMEdot11AP-MIB	Vendor/Extreme/v730b49.mib
EXTREMEdot11f-MIB	Vendor/Extreme/v730b49.mib
EtherLike-MIB	Standard/rfc3635-EtherLike-MIB.mib
FDDI-SMT73-MIB	Standard/Historic/rfc1512-FDDI-SMT73-MIB.mib
FOUNDRY-SN-ROOT-MIB	Vendor/Foundry/FOUNDRY-SN-ROOT-MIB.mib
FRAME-RELAY-DTE-MIB	Standard/rfc2115-FRAME-RELAY-DTE-MIB.mib
FtpServer-MIB	Vendor/Microsoft/ftp.mib
HC-RMON-MIB	Standard/rfc3273-HC-RMON-MIB.mib
HCNUM-TC※	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/hibernate/rfc2856-HCNUM-TC.mib
HOST-RESOURCES-MIB	Standard/rfc2790-HOST-RESOURCES-MIB.mib
HOST-RESOURCES-TYPES	Standard/rfc2790-HOST-RESOURCES-TYPES.mib
HP-ICF-OID	Vendor/Hewlett-Packard/ProCurve/hpicfOid.mib
HP-SITESCOPE-MIB	Vendor/Hewlett-Packard/HP-SITESCOPE-MIB.mib
HP-SN-AGENT-MIB	Vendor/Hewlett-Packard/hpEtherSwitch/hp-sn-agent.mib
HP-SN-ROOT-MIB	Vendor/Hewlett-Packard/hpEtherSwitch/hp-sn-root.mib
HP-SN-SWITCH-GROUP-MIB	Vendor/Hewlett-Packard/hpEtherSwitch/hp-sn-switch.mib
HP-UNIX	Vendor/Hewlett-Packard/hp-unix
HttpServer-MIB	Vendor/Microsoft/http.mib
IANA-ADDRESS-FAMILY-NUMBERS-MIB	Standard/IANA-ADDRESS-FAMILY-NUMBERS-MIB.mib
IANA-MAU-MIB	Standard/rfc4836-IANA-MAU-MIB.mib
IANA-RTPROTO-MIB	Vendor/Cisco/IANA-RTPROTO-MIB.my
IANATn3270eTC-MIB	Standard/IANATn3270eTC-MIB.mib
IANAifType-MIB	Standard/IANAifType-MIB.mib
IEEE8021-TC-MIB	IEEE/IEEE8021-TC-MIB.mib
IEEE8023-LAG-MIB	IEEE/IEEE8023-LAG-MIB.mib
IEEE802dot11-MIB	IEEE/IEEE802dot11-MIB.mib

MIB 名	MIB ファイル
IF-MIB	Standard/rfc2863-IF-MIB.mib
IGMP-MIB	Vendor/Cisco/IGMP-MIB.my
IGMP-STD-MIB	Vendor/Cisco/IGMP-STD-MIB.my
INET-ADDRESS-MIB	Standard/rfc4001-INET-ADDRESS-MIB.mib
INTEGRATED-SERVICES-MIB	Standard/rfc2213-INTEGRATED-SERVICES-MIB.mib
IP-FORWARD-MIB	Standard/rfc4292-IP-FORWARD-MIB.mib
IP-MIB	Standard/rfc4293-IP-MIB.mib
IPMCAST-MIB	Standard/rfc5132-IPMCAST-MIB.mib
IPMROUTE-MIB	Vendor/Cisco/IPMROUTE-MIB.my
IPMROUTE-STD-MIB	Vendor/Cisco/IPMROUTE-STD-MIB.my
IPV6-FLOW-LABEL-MIB	Standard/rfc3595-IPV6-FLOW-LABEL-MIB.mib
IPV6-MIB	Standard/rfc2465-IPV6-MIB.mib
IPV6-TC	Standard/rfc2465-IPV6-TC.mib
ISDN-MIB	Standard/rfc2127-ISDN-MIB.mib
InternetServer-MIB	Vendor/Microsoft/inetsrv.mib
JUNIPER-CHASSIS-DEFINES-MIB	Vendor/Juniper/mib-jnx-chas-defines
JUNIPER-JS-IF-EXT-MIB	Vendor/Juniper/mib-jnx-js-if-ext
JUNIPER-JS-SMI	Vendor/Juniper/mib-jnx-js-smi
JUNIPER-MIB	Vendor/Juniper/mib-jnx-chassis
JUNIPER-SMI	Vendor/Juniper/mib-jnx-smi
JUNIPER-V1-TRAPS	Vendor/Juniper/v1_traps
JUNIPER-VPN-MIB	Vendor/Juniper/mib-jnx-vpn
Juniper-MIBs	Vendor/Juniper/Juniper-MIBs.mib
Juniper-UNI-SMI	Vendor/Juniper/Juniper-UNI-SMI.mib
LANGTAG-TC-MIB	Standard/rfc5131-LANGTAG-TC-MIB.mib
LLDP-MIB	IEEE/lldp.mib
LanMgr-Mib-II-MIB	Vendor/Microsoft/lmmib2.mib
MAU-MIB	Standard/rfc4836-MAU-MIB.mib
MGMD-STD-MIB	Standard/rfc5519-MGMD-STD-MIB.mib
MPLS-L3VPN-STD-MIB	Standard/rfc4382-MPLS-L3VPN-STD-MIB.mib
MPLS-LSR-MIB	Vendor/Cisco/MPLS-LSR-MIB.my

MIB 名	MIB ファイル
MPLS-LSR-STD-MIB	Standard/rfc3813-MPLS-LSR-STD-MIB.mib
MPLS-MIB	Vendor/Juniper/mib-jnx-mpls
MPLS-TC-STD-MIB	Standard/rfc3811-MPLS-TC-STD-MIB.mib
MPLS-TE-MIB	Vendor/Cisco/MPLS-TE-MIB.my
MPLS-TE-STD-MIB	Standard/rfc3812-MPLS-TE-STD-MIB.mib
MPLS-VPN-MIB	Vendor/Cisco/MPLS-VPN-MIB.my
MSDP-MIB	Vendor/Nortel/MSDP-MIB.mib
Nortel-Magellan-Passport-StandardTextualConventionsMIB	Vendor/Nortel/Nortel-Magellan-Passport-StandardTextualConventionsMIB.mib
Nortel-Magellan-Passport-TextualConventionsMIB	Vendor/Nortel/Nortel-Magellan-Passport-TextualConventionsMIB.mib
Nortel-Magellan-Passport-UsefulDefinitionsMIB	Vendor/Nortel/Nortel-Magellan-Passport-UsefulDefinitionsMIB.mib
Nortel-MsCarrier-MscPassport-StandardTextualConventionsMIB	Vendor/Nortel/Nortel-MsCarrier-MscPassport-StandardTextualConventionsMIB.mib
Nortel-MsCarrier-MscPassport-TextualConventionsMIB	Vendor/Nortel/Nortel-MsCarrier-MscPassport-TextualConventionsMIB.mib
Nortel-MsCarrier-MscPassport-UsefulDefinitionsMIB	Vendor/Nortel/Nortel-MsCarrier-MscPassport-UsefulDefinitionsMIB.mib
OLD-CISCO-CHASSIS-MIB	Vendor/Cisco/OLD-CISCO-CHASSIS-MIB.my
OLD-CISCO-INTERFACES-MIB	Vendor/Cisco/OLD-CISCO-INTERFACES-MIB.my
OLD-CISCO-SYS-MIB	Vendor/Cisco/OLD-CISCO-SYS-MIB.my
OSPF-MIB	Standard/rfc4750-OSPF-MIB.mib
P-BRIDGE-MIB	Standard/rfc4363-P-BRIDGE-MIB.mib
PIM-MIB	Vendor/Cisco/PIM-MIB.my
PIM-STD-MIB	Standard/rfc5060-PIM-STD-MIB.mib
POWER-ETHERNET-MIB	Standard/rfc3621-POWER-ETHERNET-MIB.mib
PerfHist-TC-MIB	Standard/rfc3593-PerfHist-TC-MIB.mib
Q-BRIDGE-MIB	Standard/rfc4363-Q-BRIDGE-MIB.mib
RAPID-CITY	Vendor/Nortel/RAPID-CITY.mib
RFC-1212※	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/hibernate/rfc1212-RFC1212.mib
RFC-1215	Standard/rfc1215-RFC1215.mib

MIB 名	MIB ファイル
RFC1155-SMI※	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/hibernate/rfc1155-RFC1155-SMI.mib
RFC1213-MIB※	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/hibernate/rfc1213-RFC1213-MIB.mib
RFC1271-MIB※	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/hibernate/rfc1271-RFC1271-MIB.mib
RFC1315-MIB	Standard/rfc1315-RFC1315-MIB.mib
RIPv2-MIB	Standard/rfc1724-RIPv2-MIB.mib
RMON-MIB	Standard/rfc2819-RMON-MIB.mib
RMON2-MIB	Standard/rfc4502-RMON2-MIB.mib
RS-232-MIB	Standard/rfc1659-RS-232-MIB.mib
SMON-MIB	Standard/rfc2613-SMON-MIB.mib
SNMP-FRAMEWORK-MIB	Standard/rfc3411-SNMP-FRAMEWORK-MIB.mib
SNMP-REPEATER-MIB	Standard/rfc2108-SNMP-REPEATER-MIB.mib
SNMP-TARGET-MIB	Standard/rfc3413-SNMP-TARGET-MIB.mib
SNMP-VIEW-BASED-ACM-MIB	Standard/rfc3415-SNMP-VIEW-BASED-ACM-MIB.mib
SNMPv2-CONF※	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/hibernate/rfc1904-SNMPv2-CONF.mib
SNMPv2-MIB※	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/hibernate/rfc3418-SNMPv2-MIB.mib
SNMPv2-SMI※	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/hibernate/rfc2578-SNMPv2-SMI.mib
SNMPv2-TC※	NNM/server/lib/nms-mib-model.jar/com/hp/ov/nms/mib/model/hibernate/rfc2579-SNMPv2-TC.mib
SONET-MIB	Standard/rfc3592-SONET-MIB.mib
TOKEN-RING-RMON-MIB	Standard/Historic/rfc1513-TOKEN-RING-RMON-MIB.mib
TRANSPORT-ADDRESS-MIB	Standard/rfc3419-TRANSPORT-ADDRESS-MIB.mib
TUNNEL-MIB	Standard/rfc4087-TUNNEL-MIB.mib
VMWARE-AGENTCAP-MIB	Vendor/VMware/VMWARE-AGENTCAP-MIB.mib
VMWARE-ENV-MIB	Vendor/VMware/VMWARE-ENV-MIB.mib
VMWARE-OBSOLETE-MIB	Vendor/VMware/VMWARE-OBSOLETE-MIB.mib
VMWARE-PRODUCTS-MIB	Vendor/VMware/VMWARE-PRODUCTS-MIB.mib
VMWARE-RESOURCES-MIB	Vendor/VMware/VMWARE-RESOURCES-MIB.mib

MIB 名	MIB ファイル
VMWARE-ROOT-MIB	Vendor/VMware/VMWARE-ROOT-MIB.mib
VMWARE-SYSTEM-MIB	Vendor/VMware/VMWARE-SYSTEM-MIB.mib
VMWARE-TC-MIB	Vendor/VMware/VMWARE-TC-MIB.mib
VMWARE-VC-EVENT-MIB	Vendor/VMware/VMWARE-VC-EVENT-MIB.mib
VMWARE-VMINFO-MIB	Vendor/VMware/VMWARE-VMINFO-MIB.mib
VPN-TC-STD-MIB	Standard/rfc4265-VPN-TC-STD-MIB.mib
VRRP-MIB	Standard/rfc2787-VRRP-MIB.mib
WINDOWS-NT-PERFORMANCE	Vendor/Microsoft/WINDOWS-NT-PERFORMANCE.mib
WINS-MIB	Vendor/Microsoft/wins.mib
X-DDI-MIB	Vendor/Nortel/x-ddi-adapter-mib
XYLAN-BASE-MIB	Vendor/OTHER-VENDORS/XYLAN-BASE-MIB.mib
XYLAN-HEALTH-MIB	Vendor/OTHER-VENDORS/XYLAN-HEALTH-MIB.mib

注※　一部の MIB ファイルは，jar ファイルに含まれています。表中の MIB ファイルは，次に示す jar ファイル内での相対パスとなります。

- Windows：%NnmInstallDir%NNM¥server¥lib¥nms-mib-model.jar
- Linux：$NnmInstallDir/NNM/server/lib/nms-mib-model.jar

付録 C　NNMi 環境変数

NNMi には，ファイルシステム内の移動やスクリプトの作成に使用できる多数の環境変数があります。

付録 C.1　マニュアルで使用する環境変数

このマニュアルでは，主に次の 2 つの NNMi 環境変数を使用して，ファイルやディレクトリの場所を参照します。次に示す変数はデフォルト値です。実際の値は，NNMi のインストール時に行った選択内容によって異なります。

Windows

-%NnmInstallDir%: <*drive*>:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥

-%NnmDataDir%: <*drive*>:¥ProgramData¥Hitachi¥Cm2NNMi¥

メモ

Windows システムでは，NNMi のインストールプロセスによってこれらのシステム環境変数が作成されるため，すべてのユーザーがいつでも使用できます。

重要

パス名にスペースを含む場合，必ずクォート（""）で囲んでください。

（例）

```
"%NnmInstallDir%bin¥ovstatus" -c
```

Linux

-$NnmInstallDir: /opt/OV

-$NnmDataDir: /var/opt/OV

メモ

Linux システムでは，これらの環境変数を使用する場合は手動で作成する必要があります。

なお，このドキュメントでは以下の環境変数を使用しています。以下のパスのように読み替えてください。

Windows

-%jdkdir%: %NnmInstallDir%nonOV¥jdk¥zulu¥zulu8.56.0.21-ca-jdk8.0.302-win_x64

Linux

-$jdkdir: /opt/OV/nonOV/jdk/zulu/zulu8.56.0.21-ca-jdk8.0.302-linux_x64

また，このドキュメントには，NNMi 管理サーバーでユーザーログオン設定を行うときに使用する NNMi 環境変数も一部掲載されています。これらの変数の形式はNNM_*です。NNMi 環境変数の詳細リストについては，「付録 C.2　ほかの使用可能な環境変数」を参照してください。

付録 C.2　ほかの使用可能な環境変数

NNMi 管理者は，NNMi のファイルには定期的にアクセスします。NNMi には，通常アクセスする場所へ移動するための環境変数を設定するスクリプトが用意されています。

NNMi 環境変数の拡張リストをセットアップするには，次の例のようなコマンドを使用します。

Windows

```
"%NnmInstallDir%bin¥nnm.envvars.bat"
```

Linux

```
. /opt/OV/bin/nnm.envvars.sh
```

「.」と「/」の間には必ず空白を入れてください。

上記の各 OS 用のコマンドを実行したあとで，表 C-1（Windows）または表 C-2（Linux）で示す NNMi 環境変数を使用して，頻繁に使用する NNMi ファイルの場所へ移動できます。

表 C-1　Windows OS での環境変数のデフォルトの場所

変数	Windows（例）
%NNM_BIN%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥bin
%NNM_CONF%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥Conf
%NNM_DATA%	C:¥ProgramData¥Hitachi¥Cm2NNMi
%NNM_DB%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥shared¥nnm¥databases
%NNM_JAVA%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥nonOV¥jdk¥hpsw¥bin¥java.exe
%NNM_JAVA_DIR%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥java
%NNM_JBOSS%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥nmsas
%NNM_JBOSS_DEPLOY%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥nmsas¥server¥nms¥deploy
%NNM_JBOSS_LOG%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥log¥nnm
%NNM_JBOSS_SERVERCONF%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥nmsas¥server¥nms
%NNM_JRE%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥nonOV¥jdk¥hpsw
%NNM_LOG%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥log
%NNM_LRF%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥shared¥nnm¥lrf

変数	Windows（例）
%NNM_PRIV_LOG%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥log
%NNM_PROPS%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥shared¥nnm¥conf¥props
%NNM_SHARED_CONF%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥shared¥nnm¥conf
%NNM_SHARE_LOG%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥log
%NNM_SNMP_MIBS%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥misc¥nnm¥snmp-mibs
%NNM_TMP%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥tmp
%NNM_USER_SNMP_MIBS%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥shared¥nnm¥user-snmp-mibs
%NNM_WWW%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥shared¥nnm¥www

表 C-2　Linux OS での環境変数のデフォルトの場所

変数	Linux（例）
$NNM_BIN	/opt/OV/bin
$NNM_CONF	/var/opt/OV/conf
$NNM_DATA	/var/opt/OV
$NNM_DB	/var/opt/OV/shared/nnm/databases
$NNM_JAVA	/opt/OV/nonOV/jdk/hpsw/bin/java
$NNM_JAVA_DIR	/opt/OV/java
$NNM_JBOSS	/opt/OV/nmsas
$NNM_JBOSS_DEPLOY	/opt/OV/nmsas/server/nms/deploy
$NNM_JBOSS_LOG	/var/opt/OV/log/nnm
$NNM_JBOSS_SERVERCONF	/opt/OV/nmsas/server/nms
$NNM_JRE	/opt/OV/nonOV/jdk/hpsw
$NNM_LOG	/var/opt/OV/log
$NNM_LRF	/var/opt/OV/shared/nnm/lrf
$NNM_PRIV_LOG	/var/opt/OV/log
$NNM_PROPS	/var/opt/OV/shared/nnm/conf/props
$NNM_SHARED_CONF	/var/opt/OV/shared/nnm/conf
$NNM_SHARE_LOG	/var/opt/OV/log
$NNM_SNMP_MIBS	/opt/OV/misc/nnm/snmp-mibs
$NNM_USER_SNMP_MIBS	/var/opt/OV/shared/nnm/user-snmp-mibs
$NNM_TMP	/var/opt/OV/tmp

変数	Linux（例）
$NNM_WWW	/var/opt/OV/shared/nnm/www

付録 D　Causal Engine と NNMi インシデント

通信とデータネットワークは規模と複雑さが著しく伸び，発生する障害の数も増えています。障害が 1 つ発生しただけでもたくさんの警報が発生することもあり，ネットワークオペレータにとっての障壁は，あらゆる逸話的警報の中から本当の問題を見分けることになりました。従来のイベント相関システムによって警報の数を減らすことはできましたが，これらのシステムは根本原因を自動化された方法で突き止めるという点で劣る傾向があります。

NNMi Causal Engine 技術は，因果関係ベースのアプローチを使用して，**根本原因解析**（RCA）をネットワーク症状に適用します。

付録 D.1　因果関係解析－高度な考察

Causal Engine 技術によって，次の高度な機能が可能になります。

- `NmsApa` jboss サービスを使用して，ネットワークを解析する
- RCA へのモデルベースのアプローチ
  －管理対象オブジェクト同士の間の行動的関連をモデル化する
  －イベント因果関係に加えてオブジェクトモデルを使用して解析を進める
  －根本原因と影響を判定する
  －MINCAUSE アルゴリズムをベースにする
  －あいまい性および部分的症状に対処可能である
- 動的
  －解析中に症状を積極的に誘発させる
  －トポロジの変化に動的に反応する
- 拡張性
  －モジュールの階層を採用する（インポート／エクスポート）
  －ネットワーク障害のエンドツーエンドの診断を提供する
  －将来の製品でのルールセット追加を可能にする

付録 D.2　Causal Engine の概念

Causal Engine 技術では，次の逐次的アプローチを使用します。

1. 根本原因問題と症状を形式的に定義する。
2. モデルを使用して症状を根本原因問題に関連づけることで，解析を行う。
   症状の源は，次の 2 つです。
   - StatePoller（症状が状態の変化の場合）

- イベント（症状がトラップの場合）

3. 根本原因に関連する結論を生み出す。

   Causal Engine の結論には，モデルに関連したアーチファクト（成果物）が含まれています。アーチファクトには，次の詳細が含まれます。

   - インシデント発生
   - インシデント相関
   - インシデント抑制
   - インシデント中止
   - 関連するオブジェクトのステータス

付録 D.3　ステータスの概念

インシデント操作に加えて，NmsApa サービスは関連オブジェクトのステータスを設定します。ステータスはオブジェクトの状態全般を示すために使用され，未解決結論の結果として計算されます。どの結論にも重大度が関連づけられてあり，報告されるステータスはすべての未解決結論のうちで最も深刻なものになります。さらに，結論はユーザーに，オブジェクトのステータスについての根本原因（つまり理由）を知らせます。

NmsApa サービスは，次のオブジェクトを管理します。

- SNMP エージェント
- IPv4 アドレス
- インタフェース
- 接続
- ノード
- ノードグループ

NmsApa サービスは，重大度の高いものから順に次のステータスカテゴリを使用します。

- 不明
- 使用不可
- 危険域
- 重要警戒域
- 警戒域
- 注意域
- 正常域
- ステータスなし

付録 D.4　エピソードとは

NmsApa サービスの目標は，オペレータやネットワークエンジニアが対処できるたった 1 つのインシデントを提示することです。そのために，NmsApa サービスは**エピソード**の概念を使用します。エピソードは特定の期間存在し，その間に 2 番目の障害は設定に基づいて相関付けられるか抑制されます。

例

- AddressNotResponding インシデントは，InterfaceDown インシデントによって，次のシナリオに従って抑制されます。
  - IPv4 アドレスが ICMP への応答を停止すると，エピソードが開始して 60 秒間存続します。
  - その期間内に，IPv4 アドレスに関連づけられたインタフェースが停止すると，NmsApa サービスは*インタフェース停止*状態が原因で IPv4 アドレスが応答を停止したと結論付けます。
  - したがって，AddressNotResponding インシデントは発生しません。InterfaceDown インシデントだけが発生します。
  - InterfaceDown インシデントがその期間内に検出されるようにするために，NmsApa サービスは*指定ポーリング*をそのインタフェースに対して発行します。これによってネットワークエンジニアは，問題の根本原因（この場合はインタフェース）を修正できるようになります。
  - インタフェースがエピソード中に停止しない場合，NmsApa サービスはAddressNotResponding インシデントを発生します。インタフェースがエピソード後に停止すると，InterfaceDown インシデントが発生します。この場合，ネットワークエンジニアは 2 つの問題に個々に対処しなければなりません。
- NodeDown インシデントは，1 ホップネイバー（隣接）インタフェースからのInterfaceDown インシデントを，次のシナリオに従って相関付けします。
  - インタフェースが停止すると，NodeDown エピソードが隣接ノードに対して開始され，300 秒間存続します。
  - その期間内に，ノードが停止すると，InterfaceDown インシデントがNodeDown インシデントの下で相関付けされます。
  - すべての 1 ホップネイバーからのInterfaceDown インシデントがNodeDown インシデントの下に相関付けされます。InterfaceDown インシデントを，NodeDown インシデントを裏付ける証拠として検討できます。

付録 D.5　NNMi は何を解析するのか？

NNMi は SNMP プロトコルを使用して管理対象ノードから情報を，SNMP エージェント（管理対象ノードで稼働しているプロセスで，管理機能を提供する）を使用して取得します。SNMP エージェントは，管理対象ノード上のインタフェースおよびポートを管理し，1 つ以上のノードと関連づけが可能です。

SNMP エージェントに関連づけられた可能な NNMi ステータスカテゴリの一覧を次に示します。

- 不明－適用不可。

- 使用不可－適用不可。
- 危険域－SNMP エージェントは SNMP クエリーに応答しません。
- 警戒域－適用不可。
- 注意域－適用不可。
- 正常域－SNMP エージェントは SNMP クエリーに応答します。
- ステータスなし－SNMP エージェントはポーリングされません。

IPv4 アドレスは，ICMP に応答するルーティング可能なアドレスです。IPv4 アドレスは，通常はノードに関連づけされます。NNMi は，ノードのステータスを次のように報告します。

- 不明－適用不可。
- 使用不可－この IPv4 アドレスに関連づけられたインタフェースは管理できないまたは使用不可にされています。
- 危険域－IPv4 アドレスは ICMP クエリーに応答しません（デバイスを ping します）。
- 警戒域－適用不可。
- 注意域－適用不可。
- 正常域－IPv4 アドレスは ICMP クエリーに応答します。
- ステータスなし－IPv4 アドレスはポーリングされません。

インタフェースとは，ノードをネットワークに接続するために使用する物理的なポートです。NNMi はインタフェースのステータスを次のように報告します。

- 不明－インタフェースに関連づけられた SNMP エージェントは，SNMP クエリーに応答しません。不明は，`NmsApa` サービスが，`ifAdminStatus` と`ifOperStatus` を測定できないため，稼働状況を判定できないことを示します。
- 使用不可－インタフェースは管理できません（`ifAdminStatus`=down）。
- 危険域－インタフェースは操作できません（`ifOperStatus`=down）。
- 警戒域－適用不可。
- 注意域－適用不可。
- 正常域－インタフェースは操作可能です（`ifOperStatus`=up）。
- ステータスなし－インタフェースはポーリングされていません。

ノードとは，NNMi がスパイラル検出プロセスの結果として見つけ出すデバイスです。ノードには，インタフェース，ボード，およびポートを含むことができます。ノードは，次の 2 つのカテゴリに分けることができます。

1. ネットワークノード：スイッチ，ルーター，ブリッジおよびハブなどのアクティブデバイス
2. エンドノード：Linux サーバーや Windows サーバーなど

NNMi は通常はネットワークノードを管理し，ノードステータスを次のように報告します。

- 不明－ノードに関連した SNMP エージェントは SNMP クエリーに応答せず，ポーリングした IPv4 アドレスは ICMP クエリーに応答しません。これは，NNMi がノードを管理できないことを示します。
- 使用不可－適用不可。
- 危険域－次のどれかになります。
  - ノードは，隣接解析の決定によって停止しています。
  - ノードは重要とマークされており，*管理が困難です*（ノードに NNMi サーバーからアクセスできません）。
  - ノードはアイランドであり（近隣ノードがない），そのため管理が困難です。
  - `NmsApa` サービスは，ノードが停止しているか，または着信接続が停止しているかを判定できません。
- 警戒域－次のどれかになります。
  - ノードに関連づけられた SNMP エージェントは，SNMP クエリーに応答しません。
  - ノード内の 1 つ以上のインタフェースが停止しています。
  - ノード上の 1 つ以上の IPv4 アドレスが ICMP に応答していません。
- 注意域－適用不可。
- 正常域－ノードの SNMP エージェント，ポーリングしたインタフェース，およびポーリングした IPv4 アドレスは稼働しています。
- ステータスなし－ノードの SNMP エージェント，すべてのインタフェース，およびすべての IPv4 アドレスはポーリングされていません。

接続はレイヤー 2 物理接続とレイヤー 3 ネットワーク接続です。NNMi は，転送データベース（FDB）表をほかのネットワークデバイスから読み取り，CDP や EDP などの検出プロトコルをサポートするデバイスを使用することで，接続情報を検出します。NNMi は接続のステータスを，次のように報告します。

- 不明－接続のすべてのエンドポイントが不明なステータスを持っています。
- 使用不可－接続のどれか 1 つのエンドポイントが使用不可です。
- 危険域－すべてのエンドポイントは操作できません。
- 警戒域－エンドポイントのどれか 1 つが停止しています。
- 注意域－エンドポイントは，不明だが危険でないステータスを持っています。
- 正常域－すべてのエンドポイントは，操作可能です。
- ステータスなし－どれか 1 つのエンドポイントがポーリングされません。

ノードグループはノードの論理的コレクションで，ポーリング設定を分離するために使用します。管理者は，ノードタイプのグループ化を作成します。例えばルーターなど一部のノードは業務上絶対不可欠であるため，これらのルーターはより頻繁にポーリングするのがよいでしょう。そのためには，重要なルーターが入ったノードグループを定義して，これらのグループによって短いポーリングサイクルを設定します。

NNMi は，ノードグループのステータスを次のように報告します。

- 不明－グループ内のすべてのノードが不明なステータスを持っています。
- 使用不可－適用不可。
- 危険域－グループ内のすべてのノードが危険なステータスを持っています。
- 警戒域－グループ内の 1 つ以上のノードが危険なステータスを持っています。
- 注意域－ノードは，不明だが危険でないステータスを持っています。
- 正常域－グループ内のすべてのノードが正常なステータスを持っています。
- ステータスなし－グループ内のすべてのノードがステータスを持っていません。

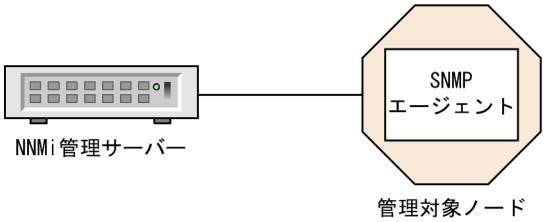
付録 D.6　失敗のシナリオは何ですか？

次のシナリオは，ネットワークでの問題の例と，Causal Engine がこれらの問題を診断するために行う作業を示しています。これらのシナリオが示すインシデントの例をほかの例とともに次の表に示します。

表 D-1　インシデントの定義

インシデント名	説明
AddressNotResponding	IPv4 アドレスは ICMP に応答していません。次の理由が考えられます。 1. ノードが停止している。 2. デバイス（ルーターなど）の設定に誤りがあるため，幾つかの IPv4 アドレスに到達できない。
InterfaceDown	インタフェースの動作状態が停止中であることを意味します。
ConnectionDown	接続の末端部の両方（またはすべて）が停止しています。
NodeDown	このインシデントは，NmsApa サービスが次の解析に基づいてノードが停止していると判定したことを示しています。 • このノードに割り当てられている IPv4 アドレスの 100%が到達できない。 • このマシンにインストールされている SNMP エージェントが応答していない。少なくとも 2 つの隣接デバイスが到達可能であり，このノードへの接続性について問題を報告している。
NodeOrConnectionDown	このインシデントは，ノードが ICMP または SNMP クエリーに応答していないことを示します。また，隣接インタフェースが 1 つだけ停止しているため，ノードが停止しているのか接続が停止しているのかNmsApa サービスが判断できないことを示しています。

(1)　SNMP エージェントが SNMP クエリーに応答しない



(説明)
管理対象ノード　：ネットワークデバイス（Ethernetスイッチなど）
SNMPエージェント：管理対象ノード用の新しいコミュニティ文字列あり
MS通信設定　　：新しいコミュニティ文字列での更新なし

シナリオ：SNMP エージェントが応答していません。例えば，この*SNMP エージェント*のコミュニティ文字列が変更され，NNMi の通信設定がまだ更新されていないが，ノードが稼働しています（IPv4 アドレスを ping 可能です）。

根本原因：SNMP エージェントが応答していません。

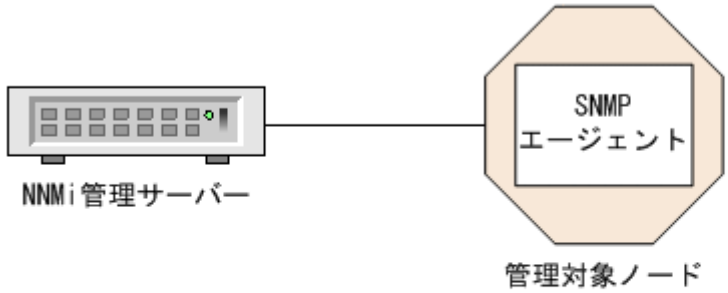
インシデント：SNMPAgentNotResponding インシデントが発生しました。

ステータス：SNMP エージェントが危険な状態です。

結論：`SNMPAgentNotResponding`

結果：ノードステータスは警戒域であり，ノードについての結論は`UnresponsiveAgentInNode` です。ポーリングされたすべてのインタフェースは，NNMi で管理できないため，不明ステータスです。各インタフェースについての結論は`InterfaceUnmanageable` です。

(2)　SNMP エージェントが SNMP クエリーに応答している



(説明)
管理対象ノード　：ネットワークデバイス（Ethernetスイッチなど）
SNMPエージェント：管理対象ノード用の新しいコミュニティ文字列あり
MS通信設定　　：新しいコミュニティ文字列で更新済

シナリオ：このシナリオは，「(1)　SNMP エージェントが SNMP クエリーに応答しない」のシナリオに続いています。NNMi 管理者が通信設定を更新して新しいコミュニティ文字列を含めることを想定します。管理対象ノードの SNMP エージェントが SNMP クエリーへの応答を開始します。

根本原因：SNMP エージェントが応答しています。

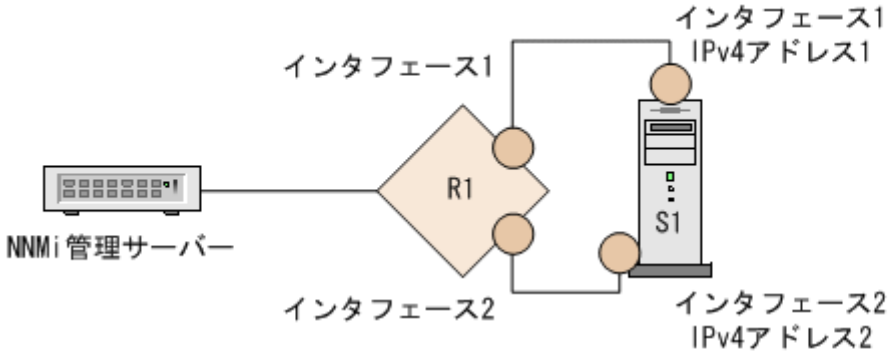
インシデント：発生なし。SNMPAgentNotRespondingインシデントがクローズしました。

ステータス：SNMPエージェントは正常な状態です。

結論：SNMPAgentResponding

結果：ノードステータスは正常域であり，ノードについての結論はResponsiveAgentInNodeです。InterfaceUnmanageableはポーリングされたすべてのインタフェースから除去されて，インタフェースは前のステータスに戻ります。

(3) IPv4アドレスがICMPに応答しない



（説明）
R1　：ルーター1
経路　：ルーター上でインタフェース1からインタフェース2に変更された
S1　：サーバー1
管理対象ノードS1　：マルチホームサーバー
S1インタフェース1　：IPv4アドレス1と関連
S1インタフェース2　：IPv4アドレス2と関連

シナリオ：S1のIPv4アドレス1が応答していません。例えば，ルーター1（R1）の経路がインタフェース1からインタフェース2に変わったことによって，S1のインタフェース1を宛て先としていたパケットが現在はR1のインタフェース2からルーティングされていると想定します。関連づけられているインタフェースは稼働しており，幾つかのIPv4アドレスをpingできるので，ノードは到達可能です。SNMPエージェントは稼働しています。

根本原因：IPv4アドレスが応答していません。

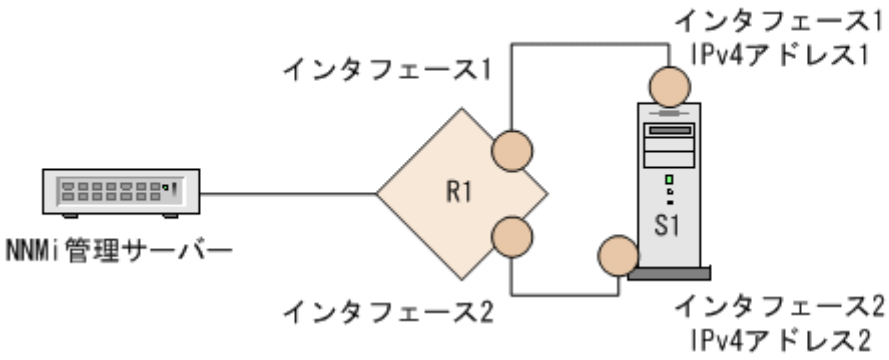
インシデント：AddressNotRespondingインシデントが発生しました。

ステータス：IPv4アドレスは危険な状態です。

結論：AddressNotResponding

結果：ノードステータスは警戒域であり，ノードについての結論はSomeUnresponsiveAddressesInNodeです。

(4)　ICMP への IPv4 アドレス応答



（説明）
R1　：ルーター1
経路　：ルーター上でインタフェース2からインタフェース1に変更された
S1　：サーバー1
管理対象ノードS1　：マルチホームサーバー
S1インタフェース1　：IPv4アドレス1と関連
S1インタフェース2　：IPv4アドレス2と関連

シナリオ：このシナリオは，「(3)　IPv4 アドレスが ICMP に応答しない」のシナリオに続いています。IPv4 アドレスが現在は応答しており，関連づけられたインタフェースが稼働しており，ノードに到達可能であることを想定してください。例えば，幾つかの IPv4 アドレスを ping できたり，SNMP エージェントが稼働していたりする状況です。

根本原因：IPv4 アドレスが応答しています。

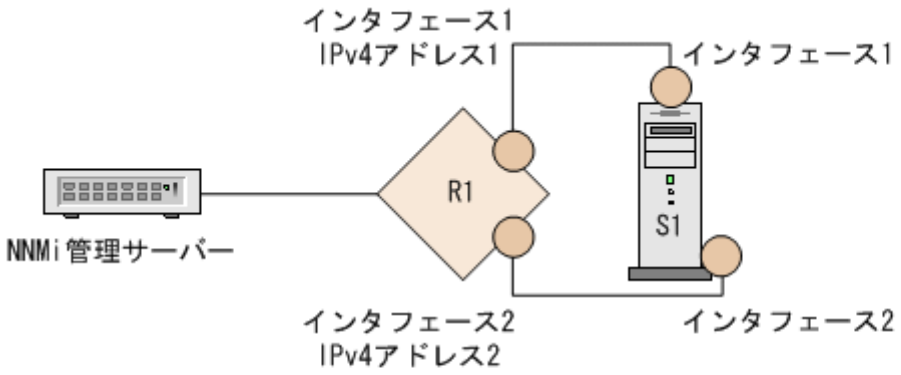
インシデント：発生なし。AddressNotResponding インシデントがクローズしました。

ステータス：IPv4 アドレスは正常な状態です。

結論：AddressResponding

結果：ノードステータスは正常域であり，ノードについての結論はResponsiveAddressesInNode です。

(5) インタフェースを操作できない



（説明）

R1	：ルーター1
R1のインタフェース1	：管理可能であるが操作できないように設定されています
R1のインタフェース1	：IPv4アドレス1に設定されています
R1のインタフェース2	：IPv4アドレス2に設定されています
S1	：サーバー1

シナリオ：R1 インタフェース 1 は操作できず（ifOperStatus=down），管理可能（ifAdminStatus=up）です。R1 はLinkDown トラップを送信します。R1 は到達可能です。幾つかの IPv4 アドレス（IPv4 アドレス 2 など）を ping できるためです。SNMP エージェントは稼働しています。IPv4 アドレス 1 はインタフェース 1 に関連づけられており，ICMP への応答を停止しました。

根本原因：インタフェースは停止しています。

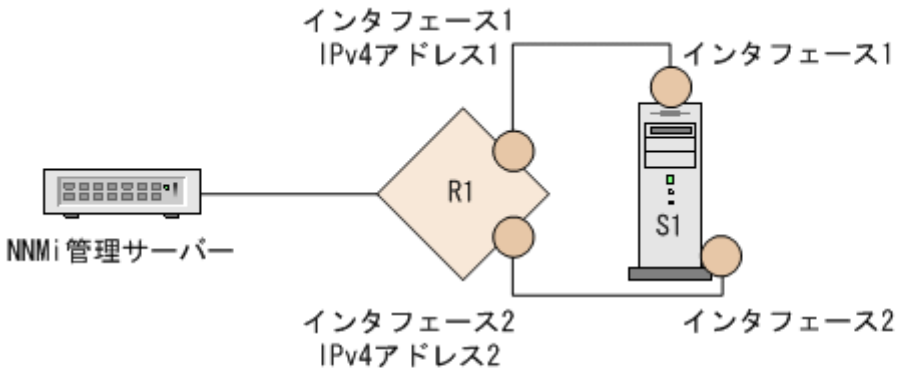
インシデント：InterfaceDown インシデントが発生しました。LinkDown インシデントがInterfaceDown インシデントの下に相関付けされています。

ステータス：インタフェースは危険な状態です。

結論：InterfaceDown

結果：ノードステータスは警戒域であり，ノードについての結論はInterfacesDownInNode です。AddressNotResponding インシデントが IPv4 アドレスに関連づけられていません。

(6)　インタフェースは操作可能である



（説明）

R1	：ルーター1
R1のインタフェース1	：管理可能であり操作可能であるように設定されています
R1のインタフェース1	：IPv4アドレス1に設定されています
R1のインタフェース2	：IPv4アドレス2に設定されています
S1	：サーバー1

シナリオ：このシナリオは，「(5)　インタフェースを操作できない」のシナリオに続いています。R1 インタフェース 1 が現在は操作可能であると想定します（ifOperStatus=up）。ノードは到達可能です。その IPv4 アドレスをすべて ping できます。SNMP エージェントは稼働しています。

根本原因：インタフェースは稼働しています。

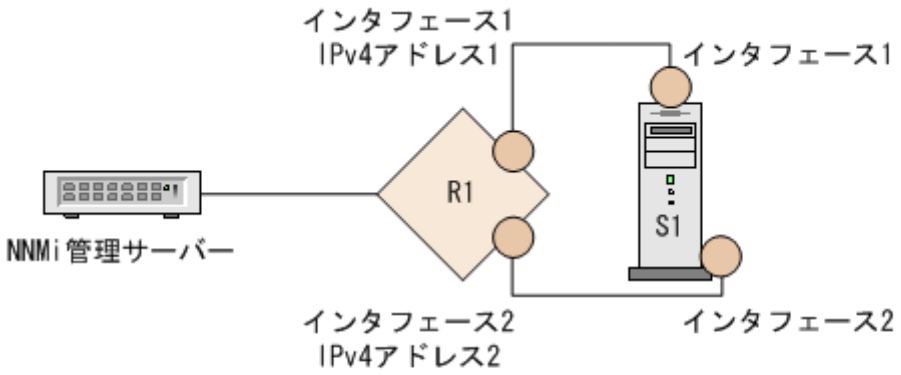
インシデント：発生なし。InterfaceDown インシデントがクローズしました。

ステータス：インタフェースは正常な状態です。

結論：InterfaceUp

結果：ノードステータスは正常域であり，ノードについての結論はInterfacesUpInNode です。

(7) インタフェースを管理できない



（説明）
R1　：ルーター1
R1のインタフェース1：管理不可能であり操作できないように設定されています
R1のインタフェース1：IPv4アドレス1に設定されています
S1　：サーバー1

シナリオ：R1 インタフェース 1 は管理できません（ifAdminStatus=down）が，ノードは到達可能です。例えば，インタフェース 2 を ping して SNMP エージェントが稼働していると想定します。R1 インタフェース 1 を無効にすると，そのインタフェースが操作できなくなります。このインタフェース IPv4 アドレス 1 に関連づけられた IPv4 アドレスが ICMP への応答を停止します。

根本原因：R1 インタフェース 1 は使用不可です。

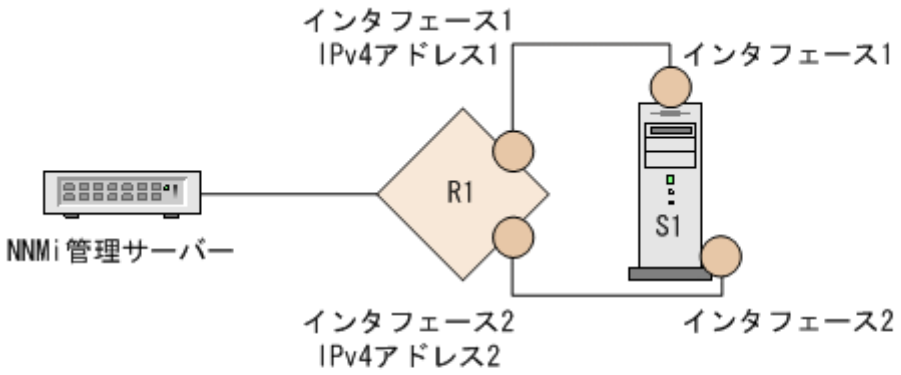
インシデント：発生なし。

ステータス：インタフェースは使用不可の状態です。

結論：InterfaceDisabled

結果：R1 インタフェース 1 に関連づけられた IPv4 アドレスはステータスが使用不可です。IPv4 アドレスについての結論はAddressDisabled です。

(8)　インタフェースを管理できる



（説明）
R1　：ルーター1
R1のインタフェース1　：管理可能であり操作可能であるように設定されています
R1のインタフェース1　：IPv4アドレス1に設定されています
S1　：サーバー1

シナリオ：このシナリオは，「(5)　インタフェースを操作できない」のシナリオに続いています。R1 インタフェース 1 が現在管理可能であり（ifAdminStatus=up），そのインタフェースの幾つかの IPv4 アドレスを ping することでこのノードに到達できると想定します。SNMP エージェントは稼働しています。R1 インタフェース 1 を有効にすることによって，操作可能になります。このインタフェースに関連づけられた IPv4 アドレスが ICMP への応答を開始します。

根本原因：インタフェースは有効です。

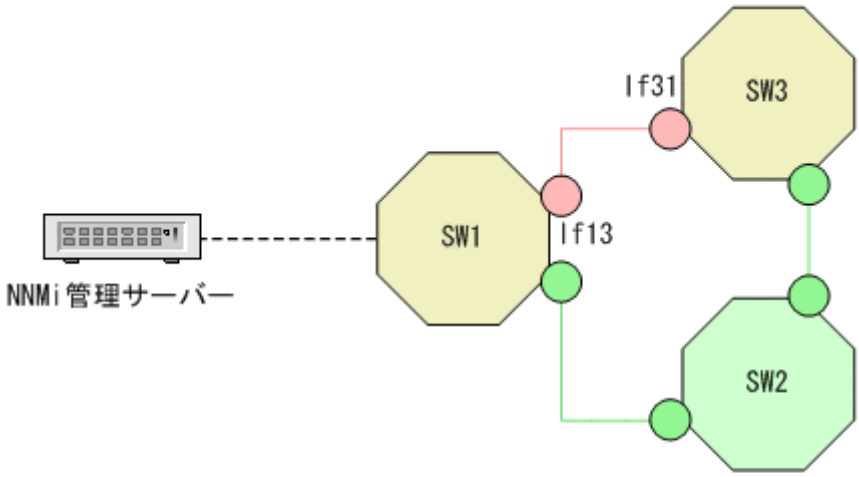
インシデント：発生なし。

ステータス：インタフェースは正常な状態です。

結論：InterfaceEnabled

結果：R1 インタフェース 1 に関連づけられた IPv4 アドレスはステータスが有効です。IPv4 アドレスについての結論はAddressEnabled です。

(9) 接続を操作できない



（説明）
SW1 ：スイッチ1
SW2 ：スイッチ2
SW3 ：スイッチ3
If31：スイッチ1に接続しているスイッチ3のインタフェース
If13：スイッチ3に接続しているスイッチ1のインタフェース

シナリオ：スイッチ 1（IF13）に接続しているスイッチ 3 のインタフェースと，スイッチ 3（IF31）に接続しているスイッチ 1 のインタフェースとの間の接続が停止しています。トラフィックは，管理サーバーからスイッチ 1（SW1）とスイッチ 2（SW2）を通って流れます。IF13 と IF31 の両方が停止とマークされます。

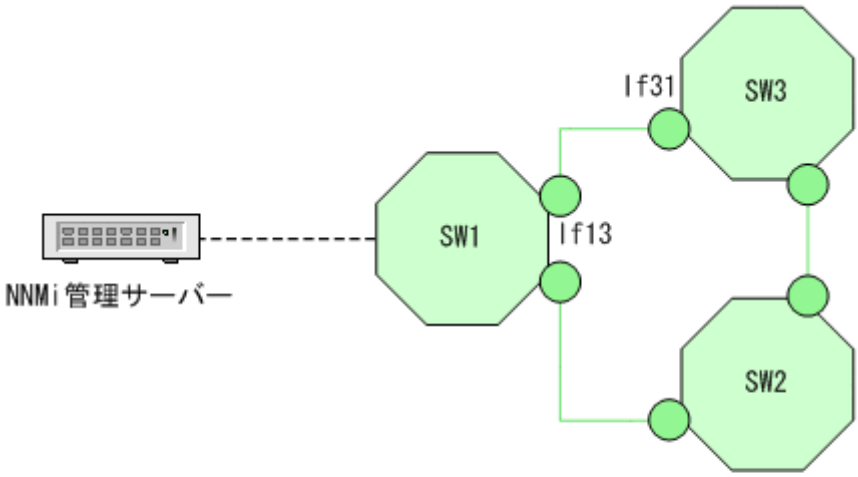
根本原因：IF13 と IF31 の間の接続が停止しています。

インシデント：ConnectionDown インシデントが発生します。IF13 と IF31 からのInterfaceDown インシデントはConnectionDown の下に相関付けされます。

ステータス：接続は危険な状態です。

結論：ConnectionDown

(10)　接続を操作できる



（説明）
SW1　：スイッチ1
SW2　：スイッチ2
SW3　：スイッチ3
If31：スイッチ1に接続しているスイッチ3のインタフェース
If13：スイッチ3に接続しているスイッチ1のインタフェース

シナリオ：このシナリオは，「(9)　接続を操作できない」のシナリオに続いています。IF13 と IF31 の間の接続が現在稼働していると想定します。

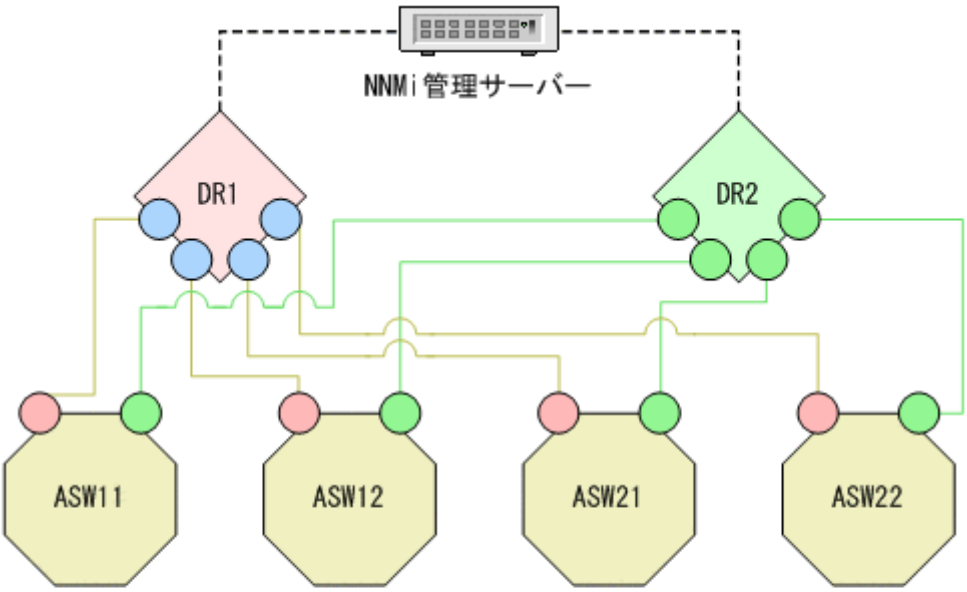
根本原因：IF13 と IF31 の間の接続が稼働しています。

インシデント：発生なし。ConnectionDown インシデントがクローズしました。

ステータス：接続は正常な状態です。

結論：ConnectionUp

(11) 直接接続しているノードが停止している



（説明）
DR1　:分散ルーター1
DR2　:分散ルーター2
ASW11:アクセススイッチ11
ASW12:アクセススイッチ12
ASW21:アクセススイッチ21
ASW22:アクセススイッチ22

シナリオ：アクセススイッチ ASW11，ASW12，ASW21，および ASW22 は，上で示すように分散ルーターに重複して接続されていると想定します。分散ルーター DR1 と DR2 は相互に直接接続しています。分散ルーター DR1 が停止します。

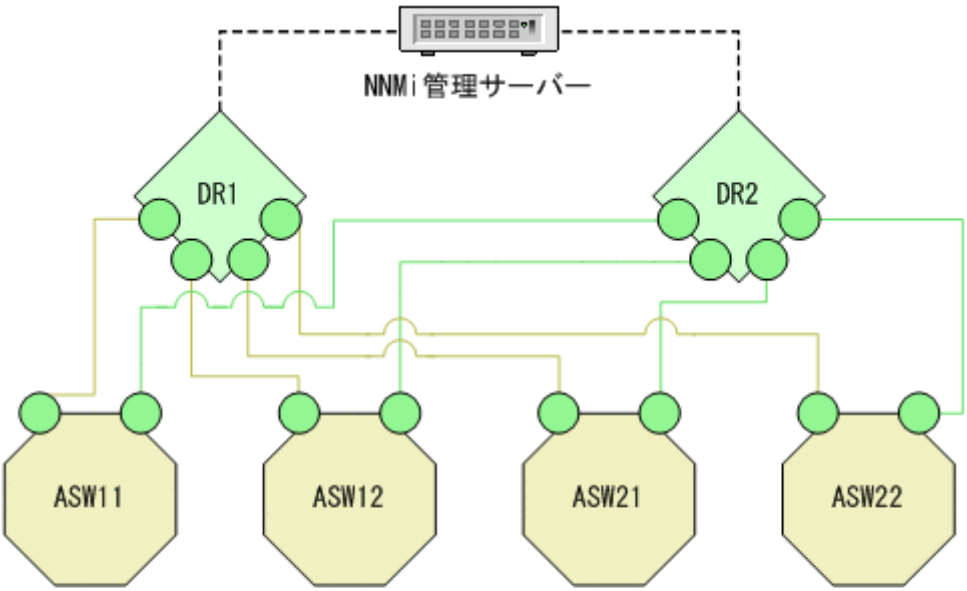
根本原因：ノード DR1 が隣接解析に従って停止しています。

インシデント：NodeDown インシデントが発生しました。1 ホップネイバーからのInterfaceDown インシデントがNodeDown インシデントの下に相関付けされます。

ステータス：ノードは危険な状態です。

結論：NodeDown

(12)　直接接続されたノードは稼働している



（説明）
DR1　：分散ルーター1
DR2　：分散ルーター2
ASW11：アクセススイッチ11
ASW12：アクセススイッチ12
ASW21：アクセススイッチ21
ASW22：アクセススイッチ22

シナリオ：このシナリオは，「(11)　直接接続しているノードが停止している」のシナリオに続いています。分散ルーター DR1 が復帰していると想定します。

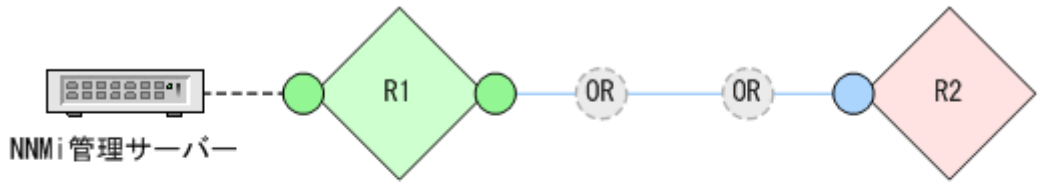
根本原因：ノード DR1 は稼働しています。

インシデント：発生なし。NodeDown インシデントがクローズしています。

ステータス：ノードは正常な状態です。

結論：NodeUp

(13) 間接接続されたノードは停止している



（説明）
R1：ルーター1
OR：光中継器（NNMiによって検出されていない）
R2：ルーター2

> **メモ**
>
> 上記図は概念図です。実際の NNMi トポロジマップまたはワークスペースビューを示していません。

シナリオ：このシナリオは，間接接続で NNMi が媒介デバイスを検出できない場合に発生します。この例では，ルーター R1 とルーター R2 は NNMi トポロジマップで直接接続しているように見えますが，実際は，これらの 2 つのルーターは光中継器経由で間接的に接続しています（光中継器は SNMP または ICMP のクエリーに応答しないため，NNMi によって検出されません）。

ルーター R2 は到達できません。原因は，接続されたインタフェースが停止しているか，または光中継器との接続が切断されているかのどちらかです。間接的にルーター R2 に接続しているルーター R1 のインタフェースは，光中継器がまだ稼働中であるため，稼働中です。

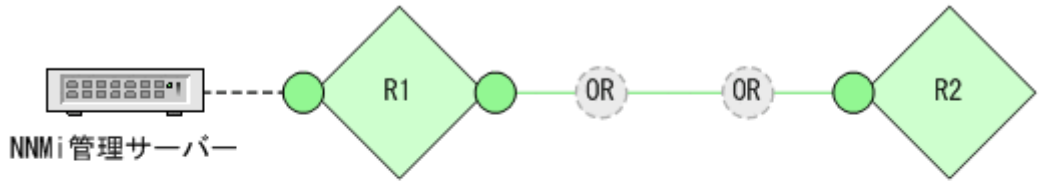
根本原因：ルーター R2 が隣接解析に従って停止しています。

インシデント：NodeDown インシデントが発生しました。

ステータス：ノード R2 は危険な状態です。

結論：NodeDown

(14) 間接接続されたノードは稼働している



（説明）
R1：ルーター1
OR：光中継器（NNMiによって検出されていない）
R2：ルーター2

メモ

上記図は概念図です。実際の NNMi トポロジマップまたはワークスペースビューを示していません。

シナリオ：このシナリオは，「(13)　間接接続されたノードは停止している」のシナリオに続いています。失敗した接続がバックアップされて，ルーター R2 が到達可能になったと想定します。

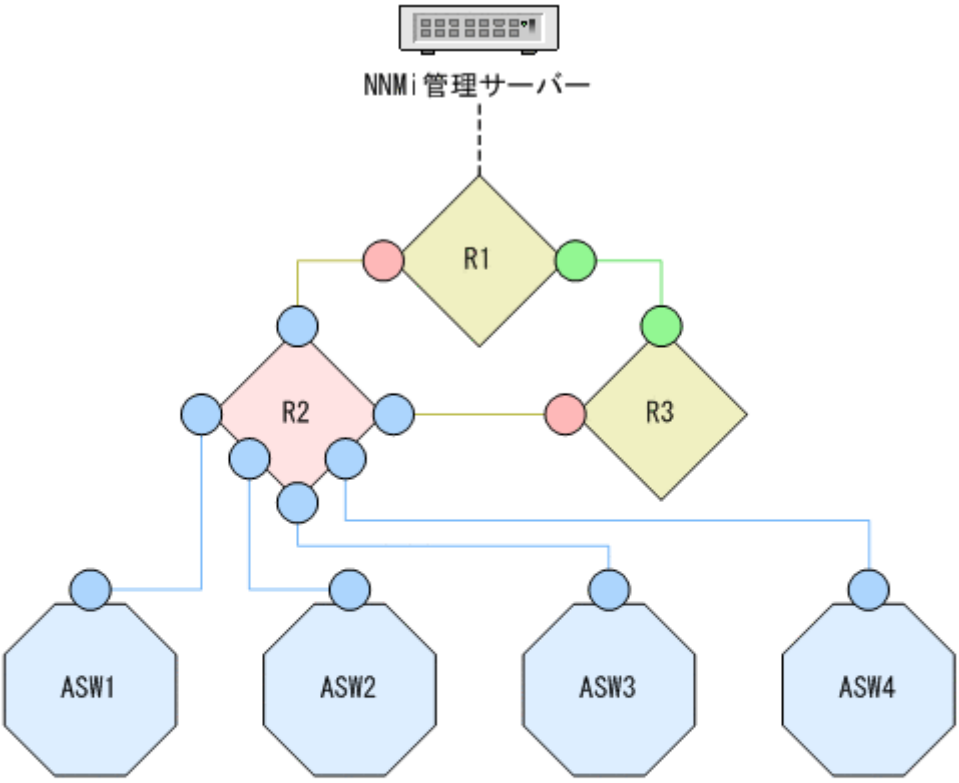
根本原因：R1 と R2 の間の接続が稼働しています。

インシデント：発生なし。NodeDown インシデントがクローズしました。

ステータス：ルーター R2 のステータスは正常域です。接続ステータスは正常域です。

結論：NodeUp

(15)　直接接続されたノードが停止しており，シャドウを作成する



（説明）
R1　：ルーター1
R2　：ルーター2
R3　：ルーター3
ASW1：アクセススイッチ1
ASW2：アクセススイッチ2
ASW3：アクセススイッチ3
ASW4：アクセススイッチ4

シナリオ：ルーター 2（R2）が上で示すように停止します。

根本原因：ノード R2 が NNMi の隣接解析に従って停止しています。

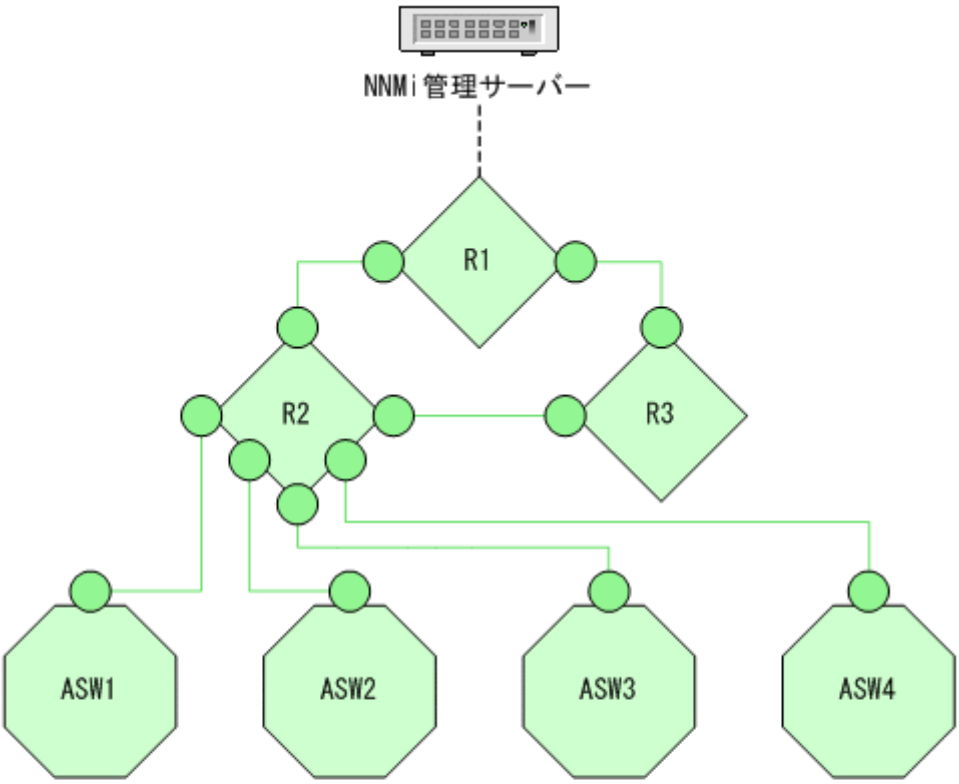
インシデント：NodeDown インシデントが発生しました。1 ホップネイバーからのInterfaceDown インシデントがNodeDown インシデントの下に相関付けされます。

ステータス：ノードは危険な状態です。

結論：NodeDown

結果：すべてのアクセススイッチが到達できません。シャドウ内のすべてのノードのステータスが不明であり，各ノードについての結論がNodeUnmanageable です。

(16)　直接接続されたノードが稼働しており，シャドウを除去している



（説明）
R1　：ルーター1
R2　：ルーター2
R3　：ルーター3
ASW1：アクセススイッチ1
ASW2：アクセススイッチ2
ASW3：アクセススイッチ3
ASW4：アクセススイッチ4

シナリオ：このシナリオは，「(15)　直接接続されたノードが停止しており，シャドウを作成する」のシナリオに続いています。図で示すように R2 が復帰していると想定します。

根本原因：ノード R2 は稼働しています。

インシデント：発生なし。NodeDown インシデントがNodeUp インシデントによってクローズしています。

ステータス：ノードは正常な状態です。

結論：NodeUp

結果：すべてのアクセススイッチが到達できるようになっています。シャドウ内のすべてのノードのステータスは正常です。

(17)　重要ノードが到達できない

シナリオ：あるノードは重要ノードグループの一部ですが，このノードが到達できなくなっています。

> **メモ**
>
> NmsApa サービスがノードを解析する前にノードを重要ノードグループに，追加する必要があります。ノードを重要ノードグループに追加する前に到達できなくなると，NmsApa サービスは NodeDown インシデントを発生しません。

根本原因：ノードは停止しています。NmsApa サービスは隣接解析を行いませんが，ノードが停止している理由は重要とマークされているためだけだと結論づけます。

インシデント：NodeDown インシデントが発生しました。相関インシデントは発生しません。

ステータス：ノードは危険な状態です。

結論：NodeDown

(18)　重要ノードが到達可能である

シナリオ：このシナリオは，「(17)　重要ノードが到達できない」のシナリオに続いています。重要ノードが復帰しており，到達できるようになったと想定します。

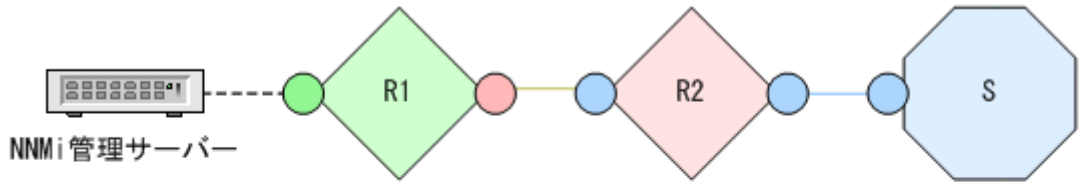
根本原因：ノードは稼働しています。

インシデント：発生なし。NodeDown インシデントがNodeUp インシデントによってクローズしています。

ステータス：ノードは正常な状態です。

結論：NodeUp

(19)　ノードまたは接続が停止している



（説明）
R1：ルーター1
R2：ルーター2
S　：アクセススイッチ

シナリオ：ルーター 2（R2）に対して冗長性がありません。R2 が停止しているか，ルーター 1（R1）と R2 の間の接続が停止しています。

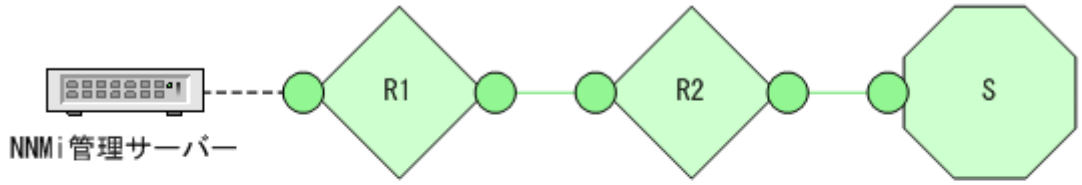
根本原因：ノードまたは接続は停止しています。

インシデント：NodeOrConnectionDown インシデントが発生しました。このシナリオのソースノードは R2 です。

ステータス：ノードは危険な状態です。接続は警戒域の状態です。

結論：NodeOrConnectionDown

(20)　ノードまたは接続が稼働している



（説明）
R1：ルーター1
R2：ルーター2
S　：アクセススイッチ

シナリオ：このシナリオは，「(19)　ノードまたは接続が停止している」のシナリオに続いています。R2 が稼働状態になったと想定します。

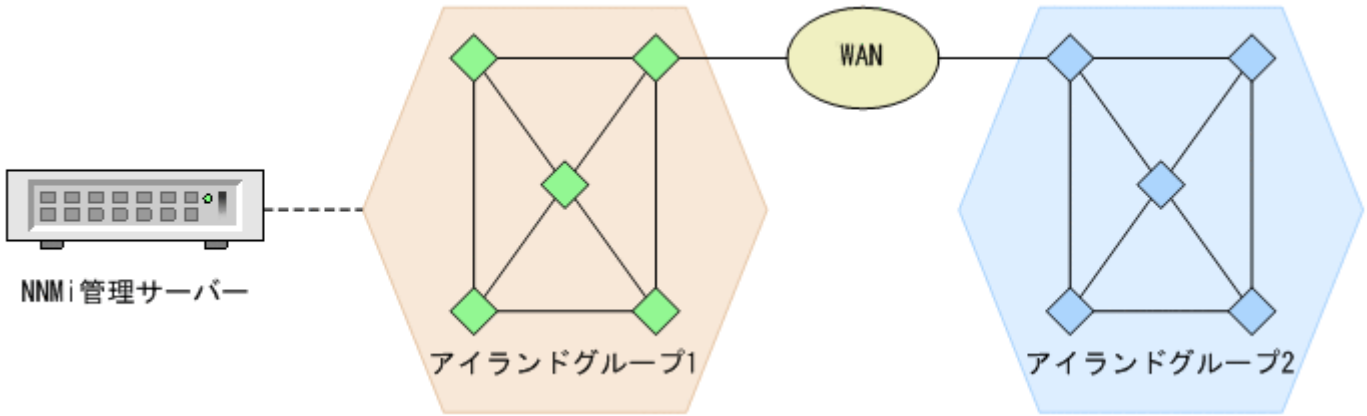
根本原因：NodeUp

インシデント：発生なし。NodeOrConnectionDown インシデントがクローズしました。

ステータス：ノードは正常な状態です。接続は正常な状態です。

結論：NodeUp

(21) アイランドグループが停止している



メモ

上記図は概念図です。実際の NNMi トポロジマップまたはワークスペースビューを示していません。

シナリオ：NNMi はネットワークを 2 つのアイランドグループに分割しました。NNMi 管理サーバーは，アイランドグループ 1 のノードに接続されます。アイランドグループ 2 は，サービスプロバイダの WAN に問題が発生したため，到達できなくなっています。

メモ

アイランドグループには，そのほかのネットワークに接続されていないか，または最低限接続しているノードの高度に接続されたセットが含まれています。例えば，NNMi は，WAN によって接続された地理的に分散されたサイトでエンタープライズネットワークの複数のアイランドグループを識別できます。アイランドグループは NNMi によって作成され，ユーザーは変更できません。アイランドグループに関する詳細については，NNMi ヘルプの NNMi コンソールを参照してください。

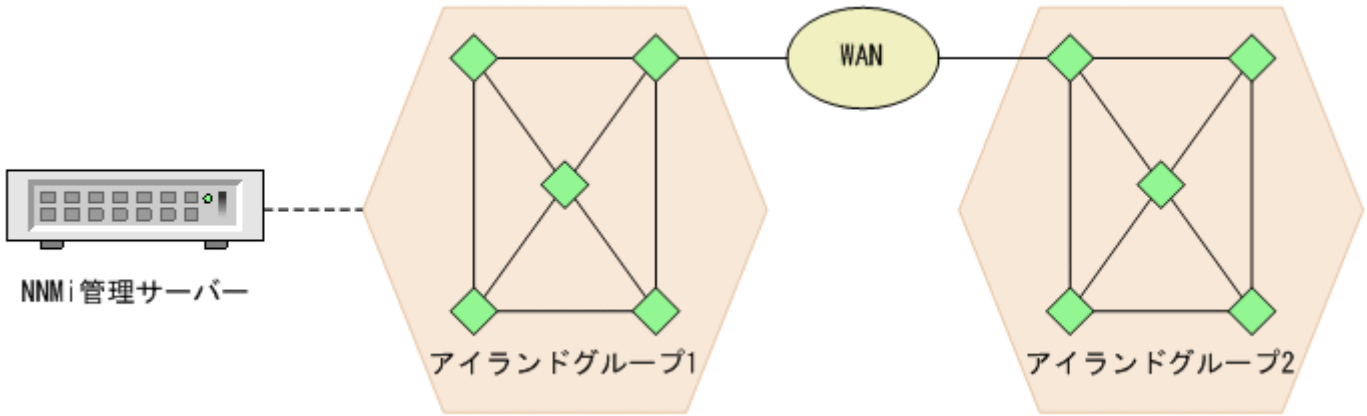
根本原因：アイランドグループ 2 が隣接解析に従って停止しています。

インシデント：`IslandGroupDown` インシデントが発生しました。NNMi はインシデントのソースノードとしてアイランドグループ 2 から代表ノードを使用します。

ステータス：アイランドグループ 2 のステータスは［不明］に設定されています。アイランドグループ 2 のオブジェクトは不明ステータスを持っています。アイランドグループ 1 の接続インタフェースは，稼働 WAN への接続がまだ稼働しているため，稼働しています。

結論：アイランドグループへの適用不可

(22)　アイランドグループが稼働している



メモ

上記図は概念図です。実際の NNMi トポロジマップまたはワークスペースビューを示していません。

シナリオ：このシナリオは，「(21)　アイランドグループが停止している」のシナリオに続いています。サービスプロバイダの WAN 問題が修正され，アイランドグループ 2 が到達可能になったと想定します。

根本原因：アイランドグループ 2 への WAN 接続はバックアップです。

インシデント：発生なし。`IslandGroupDown` インシデントがクローズしました。

ステータス：アイランドグループ 2 のステータスは［正常域］に設定されています。アイランドグループ 2 のオブジェクトは正常域ステータスに戻ります。

結論：アイランドグループへの適用不可

(23) リンク集約ポート（NNMi Advanced）

アグリゲーターが動作中

（説明）
SW1：スイッチ1
SW2：スイッチ2
If11, If12, If13：SW1上の集約ポート
If21, If22, If23：SW2上の集約ポート
If11 および If21：マスタインタフェース

次の三つの接続によって，1本のリンク集約接続が構成されています。
・-If11, If21
・-If12, If22
・-If13, If23

シナリオ：ポートアグリゲーター内のすべてのポートが運用上および管理上，動作中です。

根本原因：すべての操作および管理の状態が動作中です。

インシデント：インシデントは生成されません。

ステータス：アグリゲーターのステータスは［正常域］に設定されています。

結論：AggregatorUp

アグリゲーターの性能が低下している

（説明）
SW1：スイッチ1
SW2：スイッチ2
If11, If12, If13：SW1上の集約ポート
If21, If22, If23：SW2上の集約ポート
If11 および If21：マスタインタフェース

次の三つの接続によって，1本のリンク集約接続が構成されています。
・-If11, If21
・-If12, If22
・-If13, If23

シナリオ：ポートアグリゲーター内の一部（すべてではない）のポートが運用上停止しています。

根本原因：一部のポートの運用状態が停止中です。

インシデント：AggregatorDegraded インシデントが生成されます。

ステータス：アグリゲーターのステータスは［警戒域］に設定されています。

結論：AggregatorDegraded

アグリゲーターが機能を停止している

（説明）
SW1：スイッチ1
SW2：スイッチ2
If11, If12, If13：SW1上の集約ポート
If21, If22, If23：SW2上の集約ポート
If11 および If21：マスタインタフェース

次の三つの接続によって，1本のリンク集約接続が構成されています。
・-If11, If21
・-If12, If22
・-If13, If23

シナリオ：ポートアグリゲーター内のすべてのポートが運用上停止しています。

根本原因：すべてのポートの運用状態が停止中です。

インシデント：AggregatorDown インシデントが生成されます。

ステータス：アグリゲーターのステータスは［**危険域**］に設定されています。

結論：AggregatorDown

(24) リンク集約接続（NNMi Advanced）

リンク集約接続は動作中

（説明）
SW1：スイッチ1
SW2：スイッチ2
If11, If12, If13：SW1上の集約ポート
If21, If22, If23：SW2上の集約ポート
If11 および If21：マスタインタフェース

次の三つの接続によって，1本のリンク集約接続が構成されています。
・-If11, If21
・-If12, If22
・-If13, If23

シナリオ：接続のすべてのポートアグリゲーターメンバーが動作中です。

根本原因：接続のすべてのメンバーでアグリゲーターが動作中です。

インシデント：インシデントは生成されません。

ステータス：集約接続のステータスは［**正常域**］に設定されています。

結論：AggregatorLinkUp

リンク集約接続の性能が低下している

（説明）
SW1：スイッチ1
SW2：スイッチ2
If11, If12, If13：SW1上の集約ポート
If21, If22, If23：SW2上の集約ポート
If11 および If21：マスタインタフェース

次の三つの接続によって，1本のリンク集約接続が構成されています。
・-If11, If21
・-If12, If22
・-If13, If23

シナリオ：接続の一部（すべてではない）のポートアグリゲーターメンバーが停止中です。

根本原因：接続の一部のメンバーでアグリゲーターが停止中です。

インシデント：AggregatorLinkDegraded インシデントが生成されます。

ステータス：集約接続のステータスは［警戒域］に設定されています。

結論：AggregatorLinkDegraded

リンク集約接続が機能を停止している

（説明）
SW1：スイッチ1
SW2：スイッチ2
If11, If12, If13：SW1上の集約ポート
If21, If22, If23：SW2上の集約ポート
If11 および If21：マスタインタフェース

次の三つの接続によって，1本のリンク集約接続が構成されています。
・-If11, If21
・-If12, If22
・-If13, If23

シナリオ：接続のすべてのポートアグリゲーターメンバーが停止中です。

根本原因：接続のすべてのメンバーでアグリゲーターが停止中です。

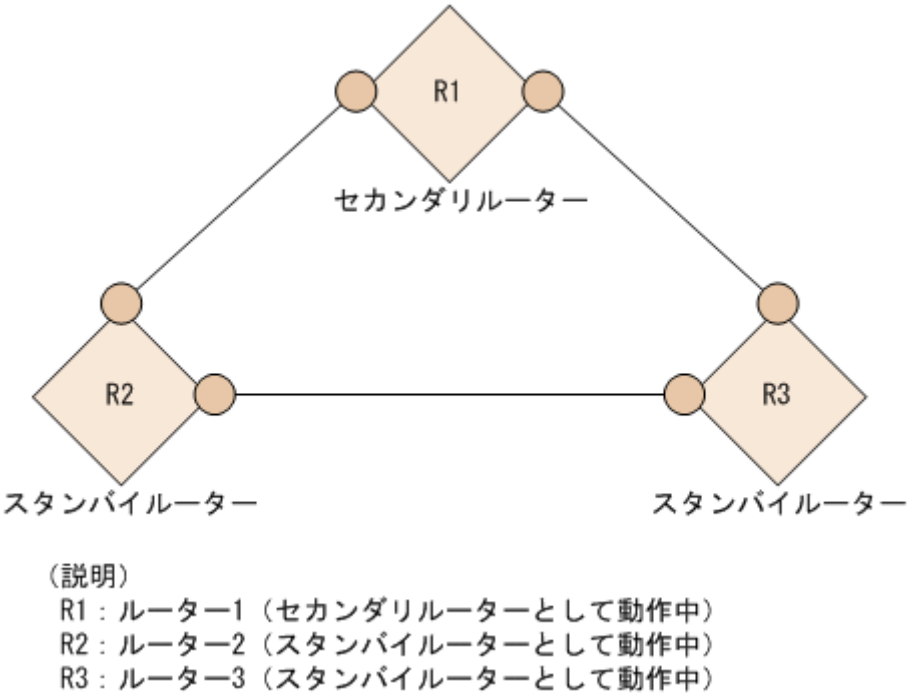
インシデント：AggregatorLinkDown インシデントが生成されます。

ステータス：集約接続のステータスは［**危険域**］に設定されています。

結論：AggregatorLinkDown

(25) ルーター冗長グループ：HSRP および VRRP（NNMi Advanced）

ルーター冗長グループにプライマリがない



シナリオ：ルーター冗長グループにプライマリメンバーが存在しません。正常に機能している HSRP または VRRP ルーターグループには，動作しているプライマリルーターとセカンダリルーターが 1 台ずつなければなりません。

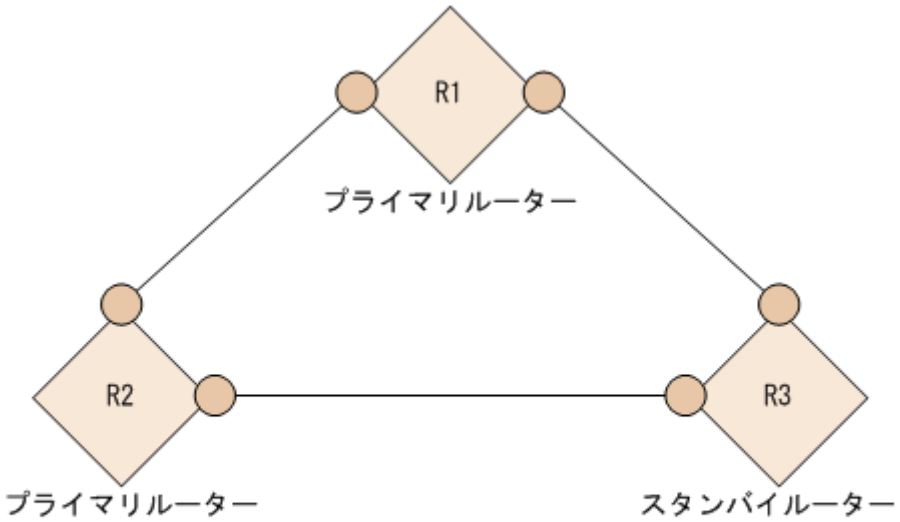
根本原因：このシナリオは，セカンダリルーターがアクティブでない場合にプライマリルーターのインタフェースに障害が発生していたか，ルーター冗長グループの設定に誤りがあったことが原因である可能性があります。

インシデント：RrgNoPrimary インシデントが生成されます。RrgNoPrimary がインパクトを受けます。InterfaceDown のような判明している根本原因がある場合は，RrgNoPrimary とInterfaceDown の間にインパクトの相関関係が生成されます。

ステータス：ルーター冗長グループのステータスは［**危険域**］に設定されています。

結論：RrgNoPrimary

ルーター冗長グループに複数のプライマリがある



（説明）
R1：ルーター1（プライマリルーターとして動作中）
R2：ルーター2（プライマリルーターとして動作中）
R3：ルーター3（スタンバイルーターとして動作中）

シナリオ：ルーター冗長グループに自身をプライマリルーターとして報告している複数のルーターが存在します。正常に機能している HSRP または VRRP ルーターグループは，動作中のプライマリルーターを 1 台だけ持っている必要があります。

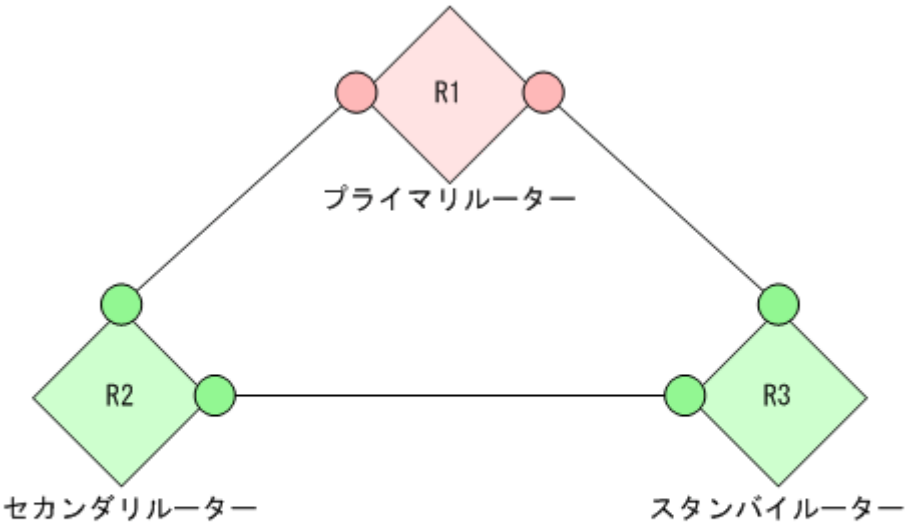
根本原因：このシナリオは，ルーター冗長グループの設定の誤りが原因である可能性があります。

インシデント：`RrgMultiplePrimary` インシデントが生成されます。`RrgMultiplePrimary` がインパクトを受けます。

ステータス：ルーター冗長グループのステータスは［**重要警戒域**］に設定されています。

結論：`RrgMultiplePrimary`

ルーター冗長グループでフェイルオーバーが起こった



（説明）
R1：最初のプライマリルーター1（障害発生中）
R2：セカンダリルーター2（プライマリルーターとして動作中）
R3：スタンバイルーター3（セカンダリルーターとして動作中）

シナリオ：ルーター冗長グループのプライマリルーターに障害が発生し，セカンダリルーターがプライマリルーターの役割を引き継ぎました。通常，スタンバイがセカンダリになり，それ自体は問題ではありません（グループは正しく機能しています）。このシナリオに対して生成されるインシデントは，グループでフェイルオーバーが発生したことを報告するためのものです。

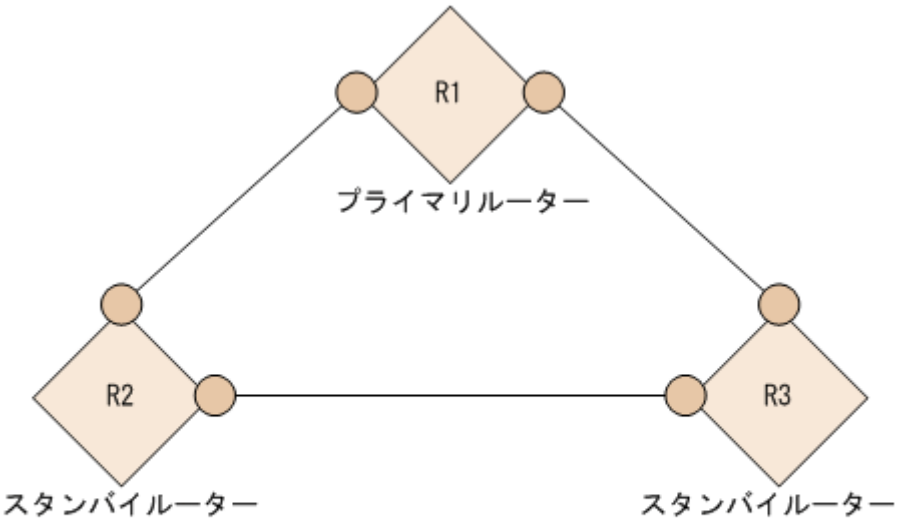
根本原因：このシナリオはプライマリルーターの障害が原因である可能性が最も高いです。

インシデント：RrgFailover インシデントが生成されます。RrgFailover の相関処理特性がインパクトを受け，InterfaceDown のような判明している根本原因がある場合は，RrgFailover インシデントと InterfaceDown インシデントとの間の相関関係がインパクトを受けます。

ステータス：この場合，ステータスは生成されません。

結論：RrgFailover

ルーター冗長グループにセカンダリがない



（説明）
R1：プライマリルーター1
R2：セカンダリルーター2（障害発生中）
R3：スタンバイルーター3（セカンダリルーターに遷移しない）

シナリオ：ルーター冗長グループのセカンダリルーターに障害が発生しました。スタンバイが存在しないか，スタンバイがセカンダリの役割を引き継ぎませんでした。

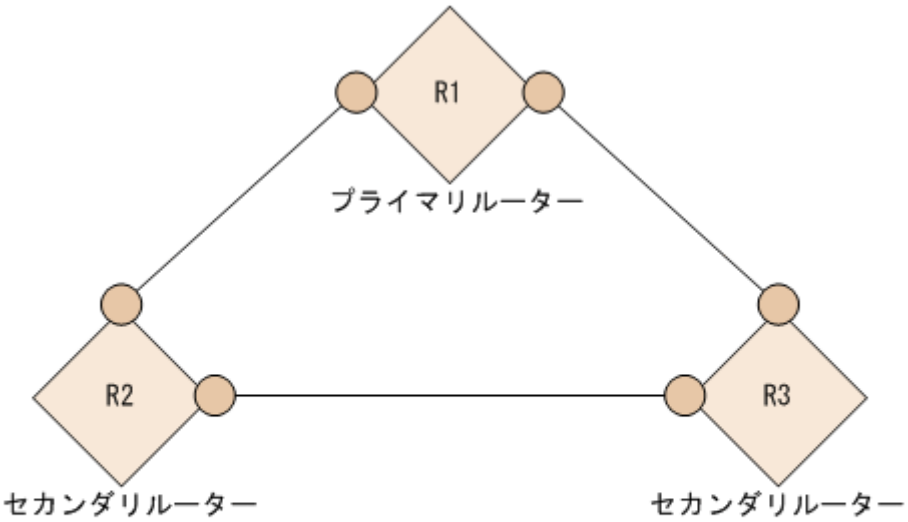
根本原因：このシナリオは，ルーターのインタフェースの障害か，ルーターグループの何らかの設定ミスが原因である可能性があります。

インシデント：RrgNoSecondary インシデントが生成されます。RrgNoSecondary の性質がインパクトを受け，InterfaceDown のような判明している根本原因がある場合は，RrgNoSecondary インタフェースと InterfaceDown インタフェースとの間の相関関係がインパクトを受けます。

ステータス：ルーター冗長グループのステータスは［警戒域］に設定されています。

結論：RrgNoSecondary

ルーター冗長グループに複数のセカンダリがある



（説明）
R1：プライマリルーター1
R2：セカンダリルーター2
R3：スタンバイルーター3（セカンダリルーターとして動作中）

シナリオ：ルーター冗長グループに自身をセカンダリルーターとして報告している複数のルーターが存在します。正常に機能している HSRP または VRRP ルーターグループは，動作しているセカンダリルーターを 1 台だけ持っていなければいけません。

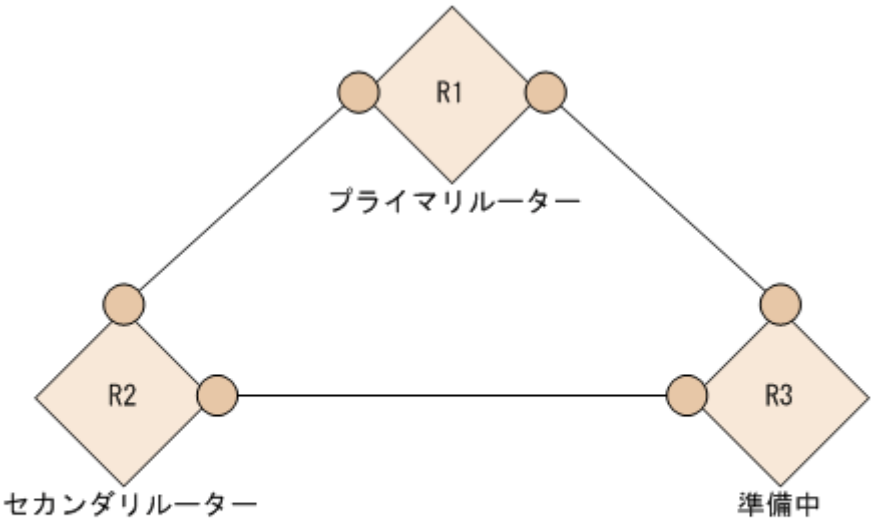
根本原因：このシナリオは，ルーター冗長グループの設定ミスが原因である可能性があります。

インシデント：`RrgMultipleSecondary` インシデントが生成されます。`RrgMultipleSecondary` の性質がインパクトを受けます。

ステータス：ルーター冗長グループのステータスは［**警戒域**］に設定されています。

結論：`RrgMultipleSecondary`

ルーター冗長グループの性能が低下した



（説明）
R1：プライマリルーター1
R2：セカンダリルーター2
R3：スタンバイルーター3

シナリオ：ルーター冗長グループに何らかの変更がありました。グループは機能しており，1 台のプライマリルーターと 1 台のセカンダリルーターがありますが，問題となりかねない何らかの異常な状態が存在します。例えば，幾つかのルーターが動作可能状態になっていない可能性があります。

根本原因：このシナリオは，ルーターグループの何らかの設定ミスが原因である可能性があります。

インシデント：RrgDegraded インシデントが生成されます。RrgDegraded の性質がインパクトを受けます。

ステータス：ルーター冗長グループのステータスは［注意域］に設定されています。

結論：RrgDegraded

(26)　コンポーネントヘルスに関するシナリオ

ファンの故障または誤動作

シナリオ：ファンセンサーがシャーシ内のファンの故障を検出しました。

インシデント：FanOutOfRangeOrMalfunctioning インシデントが生成されます。

ステータス：ファンセンサーノードコンポーネントのステータスは［危険域］です。［重要警戒域］というステータスがノードに伝えられます。

結論：FanOutOfRangeOrMalfunctioning

電源の故障または誤動作

シナリオ：電源センサーがシャーシ内の電源の故障を検出しました。

インシデント：PowerSupplyOutOfRangeOrMalfunctioning インシデントが生成されます。

ステータス：電源ノードコンポーネントのステータスは［危険域］です。［重要警戒域］というステータスがノードに伝えられます。

結論：PowerSupplyOutOfRangeOrMalfunctioning

温度の超過または誤動作

シナリオ：温度センサーがシャーシ内の高温を検出しました。

インシデント：TemperatureOutOfRangeOrMalfunctioning インシデントが生成されます。

ステータス：温度センサーノードコンポーネントのステータスは［危険域］です。ノードのステータスは変化しません。

結論：TemperatureOutOfRangeOrMalfunctioning

電圧の逸脱または誤動作

シナリオ：電圧センサーがシャーシ内の電圧の問題を検出しました。

インシデント：VoltageOutOfRangeOrMalfunctioning インシデントが生成されます。

ステータス：電圧センサーノードコンポーネントのステータスは［危険域］です。ノードのステータスは変化しません。

結論：VoltageOutOfRangeOrMalfunctioning

付録 D.7　ネットワーク設定の変更

NNMi オペレータは設定変更を 1 日のうちで，何度か行うことがあります。次のシナリオは，共通ネットワーク設定の変更について説明し，NNMi がこれらの変更に対してどう対応するかを示しています。

(1)　ノード更新中

例えば故障したインタフェースボードを，ネットワークオペレータが正常な代替品と交換して，ノードを変更する場合を想定します。NNMi がこの変更を認識すると，検出プロセスはNmsApa サービスに通知を送信します。NmsApa サービスはこの通知を使用して，次のタスクを完了します。

- ノードのステータスを再計算します。

- ノード上の削除した IPv4 アドレスおよびインタフェースのすべての登録済インシデントをクローズします。

(2) インタフェースが接続に加入および離脱する

ネットワークオペレータがネットワークデバイスの接続方法を変更する場合を想定します。インタフェースが接続に加入したり 1 つの接続を離れて別の接続に加入したりすると，NNMi 検出プロセスはNmsApa サービスに通知を送信します。NmsApa サービスはこの通知を使用して，接続のステータスを再計算します。

(3) デバイスがトラップを発生した場合

ColdStart トラップと WarmStart トラップ－NmsApa サービスは，ColdStart トラップとWarmStart トラップのイベントシステムからの通知を登録します。これらの通知が行われると，NmsApa サービスはそのトラップを発生したノードからのデバイス情報の再検出を開始します。

LinkUp トラップと LinkDown トラップ－NmsApa サービスは，LinkUp トラップおよびLinkDown トラップのイベントシステムからだけでなく，ベンダー固有のリンクトラップからの通知も登録します。これらの通知が行われると，NmsApa サービスはそのトラップを発生したノードからのデバイス情報の再検出を開始します。

メモ

NNMi が提供するトラップインシデント設定の一覧は，NNMi ヘルプを参照するか，［**設定**］ワークスペースの［**インシデント**］から［**SNMP トラップの設定**］を選択してください。

付録 D.8 NNMi 管理設定の変更

NNMi ツール管理者は NNMi 設定変更を，1 日のうちで何度か行うことがあります。次のシナリオは，共通 NNMi 管理設定の変更を説明し，NNMi がこれらの変更に対してどう対応するかを示しています。

- **NNMi 管理者は IPv4 アドレスの管理を解除するか，サービス停止にする**

  NmsApa サービスは，StatePoller からの通知を，pingState がポーリングなしに設定されたあとで受け取ります。NmsApa サービスはこの通知に反応して，IPv4 アドレスのステータスをステータスなしに設定します。

- **NNMi 管理者は IPv4 アドレスを管理するか，サービス状態に戻す**

  NmsApa サービスは，StatePoller からの通知を，pingState が測定された値に設定されたあとで受け取ります。NmsApa サービスはこの通知に反応して，IPv4 アドレスのステータスを，測定された値に基づいて計算します。

- **NNMi 管理者はインタフェースの管理を解除するか，サービス停止にする**

NmsApa サービスは，StatePoller からの通知を，operState がポーリングなしに設定されたあとで受け取ります。NmsApa サービスはこの通知に反応して，インタフェースのステータスをステータスなしに設定します。

- NNMi 管理者はインタフェースを管理するか，サービス状態に戻す

  NmsApa サービスは，StatePoller からの通知を，operState が測定された値に設定されたあとで受け取ります。NmsApa サービスはこの通知に反応して，インタフェースのステータスを，測定された値に基づいて計算します。

- NNMi 管理者はノードの管理を解除するか，サービス停止にする

  NmsApa サービスは，StatePoller からの通知を，agentState がポーリングなしに設定されたあとで受け取ります。すべてのインタフェースでoperState がポーリングなしに設定され，すべての IPv4 アドレスでpingState がポーリングなしに設定されます。NmsApa サービスはこの通知に反応して，ノードのステータスをステータスなしに設定します。

- NNMi 管理者はノードを管理するか，サービス状態に戻す

  NmsApa サービスは，StatePoller からの通知を，agentState が測定された値に設定されたあとで受け取ります。すべてのインタフェースでoperState が測定された値に設定され，すべての IPv4 アドレスでpingState が測定された値に設定されます。NmsApa サービスはこの通知に反応して，ノードのステータスを計算します。

付録 E　NNMi が使用するポートの一覧

次の表は，NNMi が管理サーバーで使用するポートを一覧で示しています。NNMi はこれらのポートをリスニングします。ポートの衝突が発生した場合，こうしたポート番号の多くは「設定の変更」欄で示した方法によって変更できます。

重要

アプリケーションフェイルオーバーが正しく機能するには，次のように設定してください。

- TCP ポート 7800-7810 をオープンにしてください。
- アクティブ NNMi 管理サーバーとスタンバイ NNMi 管理サーバーは相互のネットワークアクセスに制限のないことが必要です。

NNMi を HA 構成にしてクラスタシステムで運用する場合は，プライマリクラスタノードとセカンダリクラスタノードで使用するポート番号の設定を同じにしてください。nms-local.properties ファイルでポートを変更する場合，ノードごとに設定する必要があります（HA 構成のファイルレプリケーションでは複製されません）。

表 E-1　NNMi 管理サーバーで使用されるポート

ポート	タイプ	名称	目的	設定の変更
162	UDP	trapPort	SNMP トラップポート。	nnmtrapconfig.ovpl Perl スクリプトを使用して変更します。
1098	TCP	nmsas.server.port.naming.rmi	• NNMi コマンドラインツールで使用され，NNMi で使用されるさまざまなサービスと通信します。 • システムのファイアウォールを設定して，これらのポートへのアクセスをローカルホストだけに制限することをお勧めします。	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
1099	TCP	nmsas.server.port.naming.port	• NNMi コマンドラインツールで使用され，NNMi で使用されるさまざまなサービスと通信します。 • システムのファイアウォールを設定して，これらのポートへのアクセスをローカルホストだけに制限することをお勧めします。	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
3873	TCP	nmsas.server.port.remoting.ejb3	• NNMi コマンドラインツールで使用され，NNMi で	nms-local.properties ファイルを変更します。 • Windows：

ポート	タイプ	名称	目的	設定の変更
3873	TCP	nmsas.server.port.remoting.ejb3	使用されるさまざまなサービスと通信します。 • システムのファイアウォールを設定して，これらのポートへのアクセスをローカルホストだけに制限することをお勧めします。	%NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
4444	TCP	nmsas.server.port.jmx.jrmp	• NNMi コマンドラインツールで使用され，NNMi で使用されるさまざまなサービスと通信します。 • システムのファイアウォールを設定して，これらのポートへのアクセスをローカルホストだけに制限することをお勧めします。	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
4445	TCP	nmsas.server.port.jmx.rmi	• NNMi コマンドラインツールで使用され，NNMi で使用されるさまざまなサービスと通信します。 • システムのファイアウォールを設定して，これらのポートへのアクセスをローカルホストだけに制限することをお勧めします。	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
4446	TCP	nmsas.server.port.invoker.unified	• NNMi コマンドラインツールで使用され，NNMi で使用されるさまざまなサービスと通信します。 • システムのファイアウォールを設定して，これらのポートへのアクセスをローカルホストだけに制限することをお勧めします。	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
4457	TCP	nmsas.server.port.hq	• グローバルネットワーク管理の非暗号化トラフィックで使用します。 • メッセージングでは，グローバルマネージャーからリージョナルマネージャーへ通信が行われます。 • このポートが開くと，双方向となります。	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties

ポート	タイプ	名称	目的	設定の変更
4459	TCP	nmsas.server.port.hq.ssl	• グローバルネットワーク管理の暗号化トラフィックで使用します。 • メッセージングでは，グローバルマネージャーからリージョナルマネージャーへ通信が行われます。 • このポートが開くと，双方向となります。	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
4712	TCP	nmsas.server.port.ts.recovery	内部トランザクションサービスのポート	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
4713	TCP	nmsas.server.port.ts.status	内部トランザクションサービスのポート	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
4714	TCP	nmsas.server.port.ts.id	内部トランザクションサービスのポート	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
5432	TCP	com.hp.ov.nms.postgres.port	この PostgreSQL ポートは，この NNMi 管理サーバーに対して組み込みデータベースが待機するポートです。	nms-local.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties

ポート	タイプ	名称	目的	設定の変更
5447	TCP	trapReceiverNettyPort	TrapReceiver で JBoss からの接続を待機するポート。	nnmtrapconfig.ovpl Perl スクリプトを使用して変更します。
7500	UDP	nnmcluster	nnmcluster が使用するポート。	設定変更できません。
7800-7810	TCP	－	• アプリケーションのフェイルオーバーで使用する JGroups ポート。 • アプリケーションフェイルオーバーを使用していない場合，システムのファイアウォールを設定して，これらのポートへのアクセスを制限することをお勧めします。	nms-cluster.properties ファイルを変更します。 • Windows： %NNM_CONF%¥nnm¥props¥nms-cluster.properties • Linux： $NNM_CONF/nnm/props/nms-cluster.properties
8886	TCP	OVsPMD_MGMT	NNMi ovspmd（プロセスマネージャー）管理ポート。	1. ovstop コマンドを実行し，NNMi サービスを停止します。 2. services ファイルを開きます。 • Windows： %Windir%¥system32¥drivers¥etc¥services • Linux： /etc/services 3. 次の行をファイルに追加します。 ovspmd_mgmt <ポート番号>/tcp 4. ovstart コマンドを実行し，NNMi サービスを開始します。
8887	TCP	OVsPMD_REQ	NNMi ovsmpd（プロセスマネージャー）リクエストポート。	1. ovstop コマンドを実行し，NNMi サービスを停止します。 2. services ファイルを開きます。 • Windows： %Windir%¥system32¥drivers¥etc¥services • Linux： /etc/services 3. 次の行をファイルに追加します。 ovspmd_req <ポート番号>/tcp 4. ovstart コマンドを実行し，NNMi サービスを開始します。
8989	TCP	com.hp.ov.nms.events.action.server.port	アクションサーバーポートを有効化して設定可能にします。	nnmaction.properties ファイルを変更します。 • Windows：

ポート	タイプ	名称	目的	設定の変更
8989	TCP	com.hp.ov.nms.events.action.server.port	アクションサーバーポートを有効化して設定可能にします。	%NnmDataDir%shared¥nnm¥conf¥props¥nnmaction.properties • Linux： $NnmDataDir/shared/nnm/conf/props/nnmaction.properties
20480	TCP	nmsas.server.port.web.http	デフォルト HTTP ポート • Web UI および Web サービスに使用 • GNM 設定では，NNMi はこのポートを使用してグローバルマネージャーからリージョナルマネージャーへの通信を確立します。 • このポートが開くと，双方向となります。	nms-local.properties ファイルを変更します。インストール作業中に変更することもできます。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties
20481	TCP	nmsas.server.port.web.https	デフォルトのセキュアー HTTPS ポート（SSL） • Web UI と Web サービスに使用	nms-local.properties ファイルを変更します。インストール作業中に変更することもできます。 • Windows： %NNM_CONF%¥nnm¥props¥nms-local.properties • Linux： $NNM_CONF/nnm/props/nms-local.properties

（凡例）　－：名称はありません。

次の表は，NNMi がほかのシステムとの通信に使用するポートの一部を一覧で示しています。NNMi がファイアウォールによってこれらのシステムと分離されている場合は，ファイアウォールでこれらのポートの多くを開く必要があります。実際にどのポートを開くかは，NNMi と連携するシステムおよびそのシステムの設定によって異なります。

表 E-2　ファイアウォールの通過方向

目的	ポート番号 （ポート／タイプ）	ファイアウォールの通過方向
SNMP リクエスト	161／udp	NNMi→監視対象ノード
SNMP レスポンス	ANY／udp	NNMi←監視対象ノード※1
SNMP トラップ／SNMP Inform リクエスト	162／udp	NNMi←監視対象ノード
SNMP Inform リクエストのレスポンス	ANY／udp	NNMi→監視対象ノード※2

目的	ポート番号 （ポート／タイプ）	ファイアウォールの通過方向
SNMP トラップ転送	162／udp	NNMi→SNMP マネージャー NNMi→Northbound アプリケーション
LDAP	389／tcp	NNMi→LDAP サーバー
SSL 接続による LDAP	636／tcp	NNMi→LDAP サーバー
メッセージング bisocket コネクタ	4457／tcp	NNMi（グローバルマネージャー）→NNMi（リージョナルマネージャー）
SSL 接続によるメッセージング bisocket コネクタ	4459／tcp	NNMi（グローバルマネージャー）→NNMi（リージョナルマネージャー）
アプリケーションフェイルオーバー	7800-7810／tcp	NNMi（アクティブ）← →NNMi（スタンバイ）
NNMi コンソール	20480／tcp	• NNMi←Web ブラウザ • NNMi（グローバルマネージャー）→NNMi（リージョナルマネージャー）
SSL 接続による NNMi コンソール	20481／tcp	• NNMi←Web ブラウザ • NNMi（グローバルマネージャー）→NNMi（リージョナルマネージャー）

（凡例）

← →：

tcp の場合，コネクションを張る方向を示します。

udp の場合，パケットを送る方向を示します。

注※1　SNMP レスポンスは SNMP リクエストの送信先ポートを送信元とし，SNMP リクエストの送信元ポートを送信先とする通信です。

注※2　SNMP Inform リクエストのレスポンスは SNMP Inform リクエストの送信先ポートを送信元とし，SNMP Inform リクエストの送信元ポートを送信先とする通信です。

注 1　NNMi と監視ノード間で，ICMP についても通過させる必要があります。

注 2　ポート番号はデフォルト設定の場合です。

注 3　アプリケーションフェイルオーバーの場合の設定については，「18.　アプリケーションフェイルオーバー構成の NNMi を設定する」を参照してください。

ICMP 障害ポーリングを使用する，またはノード検出用のために Ping スイープを使用するように NNMi を設定する場合は，ICMP パケットの通過を許可するようにファイアウォールを設定する必要があります。

グローバルネットワーク管理機能を使用する場合は，グローバル NNMi 管理サーバーからリージョナル NNMi 管理サーバーに対して，表 E-3 に示すポートがアクセス可能になっている必要があります。グロー

バルネットワーク管理機能では，グローバル NNMi 管理サーバーからリージョナル NNMi 管理サーバーのこれらの TCP ポートへ通信できる必要があります。リージョナル NNMi 管理サーバーからは，グローバル NNMi 管理サーバーのこれらのポートに対して接続しません。

表 E-3　グローバルネットワーク管理で必須のアクセス可能ソケット

セキュリティ	パラメーター	TCP ポート
非 SSL	nmsas.server.port.web.http	20480
	nmsas.server.port.hq	4457
SSL	nmsas.server.port.web.https	20481
	nmsas.server.port.hq.ssl	4459

注

ポート番号はデフォルト設定の場合です。

付録 F　リファレンスページ（User Commands）

付録 F.1　nnm.envvars

NNMi の共通パスの環境変数を定義するスクリプト

SYNOPSIS

Windows オペレーティングシステム：

```
nnm.envvars.bat
```

Linux オペレーティングシステム：

```
nnm.envvars.sh
nnm.envvars.csh
```

DESCRIPTION

nnm.envvars は，共通パスの NNMi の環境変数を定義するスクリプトです。共通パスは，すべての OS プラットフォームに共通のパス名およびファイル名を提供することによって，NNMi の使用を簡素化します。共通パスは，Windows コマンドインタープリターまたは Linux シェルに応じて提供されます。

Linux では，システムにログオンするたびにシェルスクリプトが起動されるように，.profile または.login あるいは，現在の環境に記述することで，個別の端末，ユーザー，またはセッションにファイル内容を適用できます。

定義されている共通パスを参照する場合は，%NnmInstallDir%bin（Windows の場合）にある nnm.envvars.bat ファイル，または，$NnmInstallDir/bin（Linux の場合）にあるnnm.envvars.sh ファイルを参照してください。

EXAMPLES

.profile または.login ファイルを変更する場合は，次の例から適切な行をファイルに追加してください。

nnm.envvars スクリプトを有効にするには，次のコマンドを実行します。

- Windows のコマンドラインから
  %NnmInstallDir%bin¥nnm.envvars.bat
  説明：$NnmInstallDir は NNMi のインストールディレクトリです。NNMi インストーラが環境変数として登録します。
- Linux 形式のシェルがインストールされた Windows から

sh，ksh，またはbash を使用する場合：

$NnmInstallDir/bin/nnm.envvars.sh

説明：$NnmInstallDir は NNMi のインストールディレクトリです。

csh を使用する場合：

source $NnmInstallDir/bin/nnm.envvars.csh

説明：$NnmInstallDir は NNMi のインストールディレクトリです。

- Linux オペレーティングシステムから

sh，ksh，またはbash を使用する場合：

. /opt/OV/bin/nnm.envvars.sh

csh を使用する場合：

source /opt/OV/bin/nnm.envvars.csh

AUTHOR

nnm.envvars was developed by Micro Focus.

FILES

Windows オペレーティングシステム：

%NnmInstallDir%bin¥nnm.envvars.bat（Windows のコマンドラインの場合）

%NnmInstallDir%bin¥nnm.envvars.sh（sh，ksh または bash の場合）

%NnmInstallDir%bin¥nnm.envvars.csh（csh の場合）

Linux オペレーティングシステム：

$NnmInstallDir/bin/nnm.envvars.sh（sh，ksh，または bash の場合）

$NnmInstallDir/bin/nnm.envvars.csh（csh の場合）

EXTERNAL INFLUENCES

International Code Set Support：1 バイトまたはマルチバイト文字コードセットをサポートします。

付録 F.2　nnmfindattachedswport.ovpl

エンドノードが接続されているスイッチポートを見つけます。

SYNOPSIS

nnmfindattachedswport.ovpl [-u <*user name*>] [-p <*password*>] { -i <*end node file*> | -n <*end node*> } [-o <*output file*>]

DESCRIPTION

NNMi コマンドラインツールを頻繁に実行する場合は，nnm.properties ファイルを作成しておくことを推奨します。このファイルには，コマンドラインオプションの-u および-p に代わって使用されるユーザー名およびパスワードが格納されます。nnm.properties ファイルを使用することで，パスワードを入力せずに多数のコマンドを実行できます。詳細は nnm.properties リファレンスページを参照してください。

nnmfindattachedswport.ovpl コマンドは，エンドノードに接続されているスイッチポートを表示します。nnmfindattachedswport.ovpl コマンドを使用する場合，エンドノードを MAC アドレス，IP アドレス，またはホスト名として指定します。MAC アドレスは，先頭に0x や0X を付けずに大文字で指定します。また，シードファイル内で一つのエントリごとに 1 行を使用して，入力をシードファイルとして指定することができます。

次が表示されます。

- エンドノード
- スイッチのホスト名
- エンドノードに接続されているスイッチポートのインタフェース名
- エンドノードが属する VLAN 名
- エンドノードが属する VLAN ID
- インタフェースのステータスコード

ステータスコードは，NNMi がエンドノード情報を正しく取得した場合は Success（成功）を示し，そうでない場合はエラーコードを示します。表示はカンマ区切り（CSV）形式です。存在しない値に対しては，-1 が代わりに表示されます。NNMi は，すべての値が-1 の行を表示することにより，コマンドの完了を示します。

NNMi は，最初に値のそれぞれを指定するヘッダーを CSV に出力します。出力をファイルに転送するオプションがあります。その後，ファイルは Microsoft™Excel にインポートすることができます。

Parameters

nnmfindattachedswport.ovpl コマンドは，次のパラメータをサポートします。

-u <*user name*>

コマンドの実行に必要な NNMi のユーザー名を指定します。ユーザーは，システム，管理者，またはレベル 2 のオペレータが指定できます。nnm.properties ファイルが存在していない限り，これが必要になります。

-p <*password*>

コマンドの実行に必要な NNMi のユーザーパスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。

-i <*end node file*>

エンドノードのリストを含む入力ファイルを指定するには，-i を使用します。NNMi は各エンドノードに接続されているスイッチポートを検索します。ファイルは，1 行当たり 1 エントリだけを持つ必要があります。各行の値は，MAC アドレス，IP アドレス，またはホスト名を指定します。MAC アドレスは，先頭に0x や0X を付けずに大文字で指定します。-n を指定しない場合，このパラメータが必要です。

-n <*end node*>

エンドノードを指定するには，-n を使用します。nnmfindattachedswport.ovpl コマンドは，このエンドノードに接続されているスイッチポートを見つけます。エンドノードは，MAC アドレス，IP アドレス，またはホスト名を指定します。MAC アドレスは，先頭に0x や0X を付けずに大文字で指定します。

-o <*output file*>

nnmfindattachedswport.ovpl コマンドの出力が転送されるファイル名を指定するには，-o を使用します。

EXAMPLES

入力ファイルの一例を次に示します。

```
10.45.130.2
# これはコメント行です。
 con5.acme.com
000087D064CB
10.12.149.4
laserj.acme.com
```

表示例は次のようになります。

```
Input,SwitchName,IfName,VLANName,VLANId,StatusCode
10.45.130.2,-1,-1,-1,-1,UNABLE_TO_LOCATE_ENTRY_IN_FDB
con5.acme.com,-1,-1,-1,-1,UNABLE_TO_LOCATE_ENTRY_IN_ARP_CACHE
000087D064CB,10.45.130.143,2/1,Network_B_IPv4,4,SUCCESS
10.12.149.4,sw1-loop0.acme.com,Fa2/21,VLAN0490,490,SUCCESS
laserj.acme.com,sw1-loop0.acme.com,Fa2/12,mpls-intercon,169,SUCCESS
-1,-1,-1,-1,-1,-1
```

AUTHOR

nnmfindattachedswport.ovpl was developed by Micro Focus.

付録 F.3　nnmprops

NNMi プロパティの値の問い合わせ

SYNOPSIS

nnmprops [-l] [-q *prop*] [-m *match*] [-e *expand*]

DESCRIPTION

nnmprops は NNMi プロセスを実行するために使用されるプロパティ値を問い合わせるために使用されます。nnmprops コマンドは，ファイルシステム内の複数の場所に格納された，これらのプロパティを統合して表示します。このコマンドは，NNMi システムプロパティの値を問い合わせて，操作する必要のあるほかのスクリプトで使用できます。

Parameters

nnmprops コマンドは，次のオプションをサポートします。

-l

すべてのプロパティを一覧表示します。

-q *PROP*

指定したプロパティを問い合わせます。複数のプロパティを問い合わせるときは，オプションを何度も指定できます。

-m *STRING*

プレフィックスが STRING であるすべてのプロパティを問い合わせます。

-e *STRING*

STRING にあるプロパティを該当する値で展開します。

RETURN VALUE

エラーが発生しなかった場合は常にステータス0（ゼロ），それ以外の場合は1 で終了します。

EXAMPLES

```
nnmprops -l
```

すべてのプロパティとその値を一覧表示します。

```
nnmprops -q com.hp.ov.nms.trapd.blocking -q com.hp.ov.nms.trapd.udpPort
```

"com.hp.ov.nms.trapd.blocking"と"com.hp.ov.nms.trapd.udpPort"プロパティの値を問い合わせます。

```
nnmprops -m com.hp.nms.trapd
```

"com.hp.nms.trapd"で始まるすべてのプロパティを問い合わせます。例えば，".notifySourcesPeriod"，".updateSourcesPeriod"，".overallUnblockTrapRate"，".unblockTrapRate"の値が返されます。

```
nnmprops -e ”The values for com.hp.ov.nms.trapd.blocking are ${com.hp.ov.nms.trapd.blocking}.”
```

${com.hp.ov.nms.trapd.blocking}の値が展開されて，指定した文字列が表示されます。プロパティ名を識別するために，"${"と"}"が必要となることに注意してください。

AUTHOR

nnmprops was developed by Micro Focus.

FILES

nnmprops プログラムは，いくつかのプロパティファイルを使用します。これらのファイルは二つのカテゴリに分類されます。インストール時に設定されるデフォルト値，またユーザーが修正した優先される値です。インストール時に設定されるデフォルト値は，将来のバージョンで変更する場合があります。しかしながら，ユーザーが変更した値は，デフォルト値より常に優先されます。

%NnmInstallDir%misc¥nnm¥props（Windows）および$NnmInstallDir/misc/nnm/props（Linux）ディレクトリ階層の配下にあるファイルは，インストール時のデフォルト値を定義しています。

Note

NNMi の将来のバージョンで変更が上書きされる可能性があるため，この場所のファイルを編集しないでください。

%NnmDataDir%shared¥nnm¥conf¥props（Windows），%NnmDataDir%Conf¥nnm¥props（Windows），$NnmDataDir/shared/nnm/conf/props（Linux），および$NnmDataDir/conf/nnm/props（Linux）ディレクトリ配下にあるファイルは，ユーザーが変更した値や，インストール時や実行時にプログラムで変更した値を定義しています。これらの値は，最初はインストール時に提供される値のコピーをコメントアウトしたものです。これらのファイルを編集するには，コメントを削除して，値を変更します。この新しい値は，インストール時に提供されるデフォルト値より優先されます。

二つのディレクトリの違いは次のとおりです。

- %NnmDataDir%shared¥nnm¥conf¥props（Windows）および$NnmDataDir/shared/nnm/conf/props（Linux）は，クラスタ（例えば，HA クラスタあるいは NNMi アプリケーションフェイルオーバークラスタ）で共有されるプロパティを含みます。
- %NnmDataDir%Conf¥nnm¥props（Windows）および$NnmDataDir/conf/nnm/props（Linux）は，共有されない値を定義します。例えば，クラスタ内の各ノードは，同じプロパティに対して異なる値を持っている場合があります。

付録 F.4　nnmsetcmduserpw.ovpl

スクリプトを実行するときに，実行ユーザーとして -u オプションまたは-p オプションの代わりに使用されるアカウントクレデンシャルを設定します。

SYNOPSIS

nnmsetcmduserpw.ovpl

DESCRIPTION

nnmsetcmduserpw.ovpl は，通常-u オプションまたは-p オプションの指定が必要なスクリプトを実行する際に使用される NNMi アカウントクレデンシャルを設定するときに使用できます。ユーザー名とパスワードの値は，コマンドラインに-u オプションまたは-p オプションを指定せずにスクリプトが実行された場合に使用されます。

注意：Windows システムの管理者または Linux システムの root ユーザーはこのコマンドを実行しないでください。管理者または root ユーザーは，デフォルトで -u/-p を指定する必要はありません。

このコマンドを実行するには，コマンドラインスクリプトを実行するユーザーとしてシステムにログインする必要があります。ユーザー名とパスワードの値は，ユーザー単位で設定します。

Parameters

サポートするパラメータはありません。

EXAMPLES

-u オプションまたは -p オプションを通常指定するコマンドラインスクリプトを実行するときに使用するアカウントのユーザー名とパスワードを設定しておくと，コマンドラインにパスワードを指定することを回避できます。

使用方法：

```
# nnmsetcmduserpw.ovpl

警告:この変更は、このユーザーがこれらを必要とするスクリプトを実行するたびに

        -u/-pコマンドラインオプションの代わりに使用される

        資格証明に影響します。このスクリプトを実行する前に、

        希望のユーザーとしてログインしていることを確認してください。

        このスクリプトを実行すると、.nnm/nnm.propertiesファイルが

        ユーザーのホームディレクトリに作成され、編集されます。

続行しますか? [Y/N]:
Y
ユーザーコマンドパスワードの変更を続行します

ユーザー名を入力してください:
myuser
パスワードを入力してください:
mypass
```

```
パスワードを再入力してください:
mypass
ユーザー/パスワード値は /home/user/.nnm/nnm.properties に正常に保存されました
```

AUTHOR

nnmsetcmduserpw.ovpl was developed by Micro Focus.

FILES

nnmsetcmduserpw.ovpl は，次のディレクトリにあります。

- Windows：%NNM_BIN%
- Linux：$NNM_BIN

SEE ALSO

nnm.properties.

付録 F.5　nnmsnmpnotify.ovpl

SNMP 通知（Trap リクエストまたは Inform リクエスト）を発行します。

SYNOPSIS

nnmsnmpnotify.ovpl [-v *version*] [-c *community*] [-p *port(default:162)*] [-A] [-t *timeout*] [-r *retries*] [-d] [-T] [-a *agent_addr*] [-e *enterprise*] *node trap-oid variable type value* [*variable type value*]...

DESCRIPTION

nnmsnmpnotify.ovpl コマンドは，ローカルシステム上のイベントを別のシステムに知らせるため，SNMP 通知リクエストを送信します。ユーザーは，通知の肯定応答（SNMPv2 Inform）または否定応答（SNMPv1 または SNMPv2 Trap）を実行することができます。ただし，SNMP Version 1 しかサポートしていないシステムに対しては，肯定応答の通知を送信することができません。

デフォルトでは，通知が否定応答されます。nnmsnmpnotify.ovpl コマンドは，指定されたプロトコルバージョンに基づいて，SNMP Version 1 または SNMP Version 2 の Trap を送信します。デフォルトバージョンを使用する場合，SNMP Trap リクエストの送信直後にnnmsnmpnotify.ovpl コマンドが終了します。通知内容が相手側システムに実際に到着したという確認はありません。

肯定応答の通知を送信するには-A オプションを使用します。nnmsnmpnotify.ovpl コマンドは，相手側システムに対して SNMP Version 2 の Inform リクエストを送信します。このコマンドは，対応する肯定応答を待ち，必要に応じて再送信を実施します。再送信が必要な場合，nnmsnmpnotify.ovpl コマンドは，コマンドライン上で指定された *timeout* および *retries* を使用します。指定した時間および再試行内に肯定応

答が表示された場合，通知が相手側システムに到着したことが分かります。指定した時間および再試行内に肯定応答が表示されなかった場合，通知は相手側システムに到着していません。

node は，IP アドレスを持つ SNMP をサポートしているシステムです。IP アドレスまたはホスト名により指定できます。*node* に空の文字列（""）が指定された場合，送信先は localhost になります。

トラップタイプは，コマンドラインの *trap-oid* 引数にオブジェクト識別子として指定します。ユーザーは，オブジェクト識別子の書式ですべての通知を識別する必要があります。SNMPv2 MIB で定義された通知を指定することができますが，ベンダー固有の SNMPv1 MIB を`nnmsnmpnotify.ovpl` に直接指定することも可能です。しかし，`nnmsnmpnotify.ovpl` への入力として指定する前に，ベンダー固有の SNMPv1 MIB で定義されているトラップをオブジェクト識別子の書式で変換しなければなりません。SNMP Version 1 トラップで *trap-oid* の代わりに空の文字列（""）が指定された場合，汎用トラップタイプの値は 6，特定のトラップタイプの値は 0 になります。SNMP Version 2 通知の場合，*trap-oid* の varBind は設定されません。

`nnmsnmpnotify.ovpl` にトラップオブジェクト識別子を指定するときは次の目安に従ってください。

1. 6 種類の一般 SNMP トラップ（`coldStart`, `warmStart`, `linkDown`, `linkUp`, `authenticationFailure`, `egpNeighborLoss`）を生成するには，該当トラップに対応するオブジェクト識別子（RFC 1907 で定義されています）を使用します（例えば，`coldStart` トラップイベントのオブジェクト識別子は `1.3.6.1.6.3.1.1.5.1` です）。
2. SNMP 一般ではないが SNMPv2 の書式で定義されているトラップを生成するには，SNMPv2 対応の MIB から`NOTIFICATION-TYPE` 識別子を使用します。
3. SNMP 一般ではないが SNMPv1 の書式で定義されているトラップを生成するには，SNMPv1 対応の MIB からトラップエンタープライズおよび特定ナンバーを使用します。次に，書式 *enterprise.0.specific field* 欄でオブジェクト識別子を作成します。 例えば，デバイステストに対するベンダー固有の MIB を考慮してください。MIB では，enterprise `1.3.6.1.4.1.11.2.17.1` および特定トラップフィールド 4 でトラップを定義します。この結果，トラップのオブジェクト識別子は `1.3.6.1.4.1.11.2.17.1.0.4` になります。

`nnmsnmpnotify.ovpl` によりリモートノードに渡されるデータは，*variable,type,value* の三つ一組で指定されます。ユーザーは，この組を少なくとも一つコマンドラインの引数で指定する必要があります。

各変数は，10 進ドット形式またはニモニック文字列のオブジェクトインスタンス識別子です。例えば，`.1.3.6.1.4.1.11.2.17.2.1.0` または `openViewSourceId.0` 形式のいずれかを使用できます。

各 *type* は，次のタイプのいずれかとします。

INTEGER
INTEGER32
IPADDRESS
COUNTER
COUNTER32

COUNTER64（SNMPv2c に対応しているリモートノード用）
GAUGE
GAUGE32
OBJECTIDENTIFIER
OCTETSTRING
OCTETSTRINGASCII
OCTETSTRINGHEX
OCTETSTRINGOCTAL
OPAQUE
OPAQUEASCII
OPAQUEHEX
OPAQUEOCTAL
TIMETICKS
UNSIGNED32

各 *type* の詳細説明は，*RFC 1155* および *RFC 1902* を参照してください。

このときの *value* パラメータは，指定されたタイプで有効ある必要があります。16 進数または 8 進数の値が必要なタイプを使用する場合，ユーザーがこの値の各バイトを完全に定義する必要があります。例えば，`fff`（または`17377`）を指定すると，1 バイト足りないため動作しません。代わりに`0fff`（または`017377`）を使用してください。*value* は，コマンドラインで指定する必要があります。この`value`は，512 バイトを超えてはなりません。

Parameters

`-v` *`version`*

リモートノードとの通信に使用する SNMP のバージョンを指定します。*`version`* に有効な値は，`1`または`2c`です。

`-c` *`community`*

リモートノード上で認証に使用するコミュニティ文字列を指定します。

注記：シェルに影響する文字がコミュニティ文字列に含まれている場合は，必要に応じて一つ以上のエスケープ文字または引用符を使用してください。

`-p` *`port`*

リモートノードと通信するときに使用するポートを指定します。

`-t` *`timeout`*

SNMP Version 2 Inform リクエストの応答を待つタイムアウト期間を 10 分の 1 秒単位で指定します。このオプションは`-A`オプションを使用する場合に有効です。

-r *retries*

SNMP Version 2 Inform リクエストの応答が受信されない場合に試みるリトライ数を指定します。このオプションは-A オプションを使用する場合に有効です。

-d

ASN.1 パケットトレースをダンプします。

-T

OID を 10 進ドット形式で出力します。

-a *agent_addr*

ローカルホストを指定エージェントアドレスの通知ソースとみなして無視します。*agent_addr* は，IP アドレスまたはホスト名でなければなりません。

-e *enterprise*

指定した*enterprise* 値の通知に対してエンタープライズオブジェクト識別子のデフォルトを無視します。

EXAMPLES

次のコマンドは，ノードv2c_node に対して SNMP リンクダウン Inform リクエストを送信します。

```
nnmsnmpnotify.ovpl -A -v2c v2c_node .1.3.6.1.6.3.1.1.5.3
```

次のコマンドは，エージェントアドレスをagent に設定して，ノードv1_node に対し SNMP リンクダウン Trap リクエストを送信します。

```
nnmsnmpnotify.ovpl -a agent v1_node .1.3.6.1.6.3.1.1.5.3
```

AUTHOR

nnmsnmpnotify.ovpl was developed by Micro Focus.

FILES

次の環境変数は，ユーザー自身のシェルおよびプラットフォームの要求条件に従って確立される一般的なパスです。

- Windows：%NNM_BIN%¥nnmsnmpnotify.ovpl
- Linux：$NNM_BIN/nnmsnmpnotify.ovpl

SEE ALSO

nnmsnmpwalk.ovpl，nnmsnmpset.ovpl，nnmsnmpbulk.ovpl.

RFC 1155, 1157, 1212: SNMP Version 1.

RFC 1901 - 1908, 2576, 2578, 3416 - 3418: SNMP Version 2.

RFC 3411 - 3415: SNMP Version 3.

EXTERNAL INFLUENCES

Environmental Variables

$LANG は，メッセージを表示するときの言語を決定します。$LANG が指定されていない場合，または空の文字列に設定された場合，$LANG ではなくC がデフォルトに使用されます。国際化変数のどれかに無効な設定値がある場合，nnmsnmpnotify.ovpl は，国際化変数のすべてがC に設定されているように処理します。

International Code Set Support

シングルバイトまたはマルチバイトの文字コードセットをサポートします。

注記：タイプoctetstringascii の SNMP MIB 値は VT-ASCII に限定されています。

付録 F.6　ovstatus

NNMi 管理対象プロセスの状態の報告

SYNOPSIS

ovstatus [[-c] [-d] [-v] [*managed_process_names*...]]

DESCRIPTION

ovstatus は NNMi 管理対象プロセスの現在の状態を報告します。ovstatus は状態要求（OVS_REQ_STATUS）をプロセス管理プロセス（Linux オペレーティングシステム）またはサービス（Windows オペレーティングシステム）である ovspmd に送信します。一つ以上のmanaged_process_name 引数で呼び出された場合，ovstatus は指定された管理対象プロセスの状態を報告します。引数なしで呼び出された場合，ovstatus は NNMi 起動ファイル（SUF）に追加された，（ovspmd 自身を含む）すべての管理対象プロセスの状態を報告します。

ovstart とは異なり，ovstatus は ovspmd が既に起動中ではなくてもこれを起動しません。

管理対象プロセスは ovaddobj によってローカル登録ファイルの情報から構成されます。管理対象プロセスは，その管理対象プロセスを記述している LRF の第 1 フィールドによって命名されます。

Parameters

ovstatus コマンドは，次のオプションをサポートします。オプションではない第 1 引数と，以降のすべての引数は状態を報告する管理対象プロセスの名前として解釈され，状態要求で ovspmd に渡されます。

-c

　各管理対象プロセスの状態行を 1 行出力します。

-d

ovspmd への連絡と状態要求の送信，および通信チャネルの閉鎖を含む処理の重要な段階を報告します。

-v

管理対象プロセスからの冗長メッセージを印刷します。特に，このオプションは現在のすべてのovw セッションを記述しているovuispmd からの冗長メッセージを表示します。

RETURN VALUE

通常，ovstatus はステータス0（ゼロ）で終了します。ゼロ以外の状態を返すのは，ovspmd が起動していないなど，システムに問題がある場合だけです。

DIAGNOSTICS

ovstatus は特定のコマンドラインエラー（特に引数過多）およびシステムエラーを報告します。メッセージの先頭には「ovstatus:」が付与され，内容が一目瞭然であることを目的としています。ovstatus は ovspmd から受信したエラーメッセージも出力します。これらのメッセージの先頭には「ovspmd:」が付与されます。ovstatus はサポートしていないオプションは無視します。

ovstatus は，すべてのOVs_WELL_BEHAVED プロセスおよびOVs_NON_WELL_BEHAVED プロセスの既知の状態を報告します。OVs_DAEMON プロセスはovspmd の制御外で動作します。ovspmd はこれらのプロセスを把握できないので，該当プロセスはPID，unknown 状態，および最終メッセージDoes not communicate with ovspmd を報告します。

ovspmd は複数の要求（ovstart，ovstop，ovstatus）を一度に処理できることに注意してください。これらのコマンドのどれかが処理中の場合，新規要求は前回のコマンドが完了するまでタイプ別に待ち行列に入れられます。

AUTHOR

ovstatus was developed by the Micro Focus.

FILES

次に示す環境変数は，お使いのシェルおよびプラットフォーム要件に応じて確立された汎用パス名を表します。次に示すファイルの環境変数の使用については，nnm.envvars リファレンスページを参照してください。

Windows：

```
%NNM_BIN%¥ovstatus
%NNM_BIN%¥ovspmd
```

Linux：

```
$NNM_BIN/ovstatus
$NNM_BIN/ovspmd
```

EXTERNAL INFLUENCES

Environmental Variables

$LANG は，国際化変数 LC_ALL，LC_CTYPE，LC_MESSAGES が未設定，NULL，または無効な場合にデフォルト値を提供します。

$LANG が未設定，NULL，または無効な場合，デフォルト値の C（Windows の場合 English_UnitedStates.1252）が使用されます。

LC_ALL（または $LANG）はovspmd が起動したほかのすべてのプロセスのロケールを決定します。

LC_CTYPE は，テキストをシングルバイト文字，マルチバイト文字，あるいはその両方として解釈すること，文字の分類は印刷可能とすること，正規表現の文字クラス表現によって文字を一致させることを決定します。

LC_MESSAGES は，メッセージを表示する言語を決定します。

SEE ALSO

ovstart, ovstop, ovspmd, nnmcluster.

付録 G　リファレンスページ（Administrator Commands）

付録 G.1　jp1nnmiinitconfig.ovpl

設定値変更簡易化コマンド（本コマンドにより，一部の設定を簡単に設定できます。）

SYNOPSIS

```
jp1nnmiinitconfig.ovpl help
```

```
jp1nnmiinitconfig.ovpl setUndefTrap [-template (once|multiple) | [-severity <string>] [-nature <string>] [-multiple (true|false)]]
```

```
jp1nnmiinitconfig.ovpl setImportantNodeGroup [-force]
```

DESCRIPTION

本コマンドでは，UndefinedSNMPTrap インシデントに関する設定, および，"重要なノード"にすべてのノードを登録する設定が可能です。

本コマンドを実行すると，既存の設定を上書きしますので，本機能に関連する設定を実施していない状態で使用してください。

1. UndefinedSNMPTrap インシデントについての説明

   本設定を行うことで，NNMi が"SNMP トラップの設定" に定義がない OID の SNMP トラップを受け取った際に"UndefinedSNMPTrap"インシデントを生成することができます。

   UndefinedSNMPTrap インシデントには，破棄した SNMP トラップの OID 情報が含まれます。

   NNMi は"SNMP トラップの設定" に定義のない OID の SNMP トラップを破棄しますが，本設定を有効にすると，破棄した SNMP トラップの OID がわかるため，"SNMP トラップの設定"に追加すべきトラップ定義を把握するのに役立ちます。

   本コマンドによる設定では，UndefinedSNMPTrap インシデントの重大度，原因のほか，各 OID につき 1 度だけ UndefinedSNMPTrap を生成するか，受信するたびに毎回 UndefinedSNMPTrap を生成するかを選択できます。

   ※注：本設定実施時の注意事項

   - 本コマンドで設定を実施した場合，OID が完全一致していなくても，前方一致している SNMP トラップインシデント定義が存在していれば，その定義が使用されるようになります。
   - 本コマンドは，設定ファイル nms-jboss.properties を変更します。手動で変更する場合は，「8.6.1　未定義のトラップのインシデントを有効化にする」に従って，設定ファイルを変更してください。

2. 重要なノードノードグループについての説明

本設定を行うことで，"重要なノード"ノードグループにすべてのノードが所属するようになります。すべてのノードについて，無応答時に，根本原因ではなくてもインシデントを通知したい場合は，本設定を行ってください。

※注：重要なノードについて

"重要なノード"ノードグループに含めたノードは，NNMi の根本原因解析機能で特別扱いされるようになり，ノードが無応答になった際，ノードが停止したことを示すインシデント（NodeDown など）が必ず発生するようになります。

NNMi は，障害発生時，発生した事象のすべてをそのまま通知するのではなく，監視結果を解析し，根本原因に絞って通知します。この根本原因解析は，問題に迅速に対応でき，運用負担を低減する機能ですが，運用によっては弊害が生じるケースもあります。

例えばサーバー A が，ネットワーク経路上のルーター障害によって通信できなくなった場合，根本原因としてルーター障害がインシデントが通知されます。その際，サーバー A と通信ができない現象は，ルーター障害の影響による副次的な事象（根本原因ではない）と判定され，サーバー A が停止したことを示すインシデントが通知されないことがあります。

そのため，無応答になったノードすべてについてインシデントを発行したい場合，本設定を使用し，"重要なノード"にすべてのノードを登録してください。

なお，本設定を実施後，設定を元に戻す場合や，一部のノードを除外する場合は，GUI の下記から，設定を編集してください。（本コマンドを実施した場合 [追加のフィルター]に"hostname is not null"が設定されています。）

- [設定]-[オブジェクトグループ]-[ノードグループ]-[重要なノード]ノードグループ-[追加のフィルター]タブ

Commands

`help`

コマンドの使用方法を表示します。

`setUndefTrap`

UndefinedSNMPTrap インシデントに関する設定を変更します。

本コマンドを使用することで，UndefinedSNMPTrap インシデントを有効化します。

オプションを指定することで，指定したパラメータに従って UndefinedSNMPTrap インシデントが生成されるようになります。オプションは，`-template` によるテンプレート設定，もしくは`-severity`，`-nature`，`-multiple` による個別設定が選択できます。

本コマンドの設定後，設定を有効化するためには，NNMi を再起動してください。

※注：オプション省略時の動作について

- 本コマンドは，オプションを省略した場合，省略した部分の設定は変更しません（すべてのオプションを省略した場合，UndefinedSNMPTrap インシデントの有効化のみを行います）。
- オプションを省略した場合，指定しなかった設定は未定義になり，undefined と表示されます。undefined となった設定は，以下の内容で動作します。

- 重大度：NORMAL（正常域），原因：INFO（情報），-multiple false 相当（UndefinedSNMPTrap インシデントを，各 OID につき 1 回だけ生成する）

`setImportantNodeGroup`

NNMi に重要なノード ノードグループにすべてのノードを登録する設定を行います。

本設定を行うことで，"重要なノード"ノードグループにすべてのノードが所属するようになります。すべてのノードについて，無応答時に，根本原因ではなくてもインシデントを通知したい場合は，本コマンドを使用してください。

本コマンドを使用する際は，NNMi を起動する必要があります。NNMi 停止時に実行する場合は，`-force` オプションを使用することで，NNMi を起動できます。

Parameters

`-severity <string>`

UndefinedSNMPTrap インシデントの重大度を指定します。下記のいずれかが指定可能です。

NORMAL, WARNING, MINOR, MAJOR, CRITICAL

`-nature <string>`

UndefinedSNMPTrap インシデントの原因に設定される値を指定します。下記のいずれかが指定可能です。

ROOTCAUSE，SECONDARYROOTCAUSE，SYMPTOM，SERVICEIMPACT，NONE，INFO

`-multiple (true|false)`

UndefinedSNMPTrap インシデントを，各 OID につき 1 回だけ生成するか，受信するたびに毎回生成するかを選択します。

- `true`：同じ OID の SNMP トラップを複数回受け取った場合，毎回インシデントを生成します。
- `false`：SNMP トラップの各 OID につき 1 度だけ UndefinedSNMPTrap インシデントを生成します。

`-template (once|multiple)`

設定テンプレート名（`once`，`multiple`）を指定します。他のサブオプションとは同時に指定できません。

- `once` を実行した場合，下記と同じ設定となります。

  `-severity WARNING -nature INFO -multiple false`

- `multiple` を実行した場合，下記と同じ設定になります。

  `-severity NORMAL -nature INFO -multiple true`

`-force`

NNMi 停止時に指定した場合，NNMi を起動します。

EXAMPLES

1. setUndefTrap の設定例

   未定義の OID の SNMP トラップ受信時，各 OID につき 1 度だけ UndefinedSNMPTrap インシデントを生成する場合は，下記を実行します（テンプレートを使用した設定例）。

   ```
   jp1nnmiinitconfig.ovpl setUndefTrap -template once
   ```

   未定義の OID の SNMP トラップ受信時，毎回 UndefinedSNMPTrap インシデントを生成する場合は，下記を実行します（テンプレートを使用した設定例）。

   ```
   jp1nnmiinitconfig.ovpl setUndefTrap -template multiple
   ```

   重大度や原因の値を細かく指定したい場合は，-severity，-nature，-multiple オプションで設定します。

   以下は，UndefinedSNMPTrap インシデントの重大度を「MINOR(警戒域)」，原因を「SYMPTOM(症状)」にし，未定義のトラップ受信時，毎回インシデントを生成する例です。

   ```
   jp1nnmiinitconfig.ovpl setUndefTrap -severity MINOR -nature SYMPTOM -multiple true
   ```

2. setImportantNodeGroup の設定例

   NNMi 停止時に，NNMi を起動し，すべてのノードを"重要なノード" ノードグループに登録する場合，下記を実行します。

   ```
   jp1nnmiinitconfig.ovpl setImportantNodeGroup -force
   ```

NOTES

本コマンドにおいて，setImportantNodeGroup を設定する際は，NNMi を起動する必要があります。

また，setUndefTrap を設定する場合，設定を反映するためには NNMi を再起動する必要があります。

両方のメインオプションを設定する場合は，下記手順で設定できます。

1. NNMi 停止時に設定する場合

   ```
   jp1nnmiinitconfig.ovpl setUndefTrap [オプション]
   jp1nnmiinitconfig.ovpl setImportantNodeGroup -force
   ```

   ※完了後，NNMi が起動した状態になりますので，必要に応じて NNMi を停止してください。

2. NNMi 起動時に設定する場合

   ```
   jp1nnmiinitconfig.ovpl setImportantNodeGroup
   jp1nnmiinitconfig.ovpl setUndefTrap [オプション]
   ovstop
   ovstart
   ```

AUTHOR

jp1nnmiinitconfig.ovpl was developed by Hitachi Ltd.

FILES

$NNM_BIN/jp1nnmiinitconfig.ovpl

付録 G.2　nmsdbmgr

このコマンドは，データベース接続性の定期テストを含め，NNMi の組み込みデータベースを制御します。

SYNOPSIS

nmsdbmgr [-ovspmd] [-start] [-test] [-initnmsdb] [-stop] [-status] [-kill]

DESCRIPTION

nmsdbmgr は，ovspmd プロセスで NNMi の組み込みデータベースを制御するための，インターフェイスを提供するプログラムです。このプログラムを使用すると，ovspmd プロセスで，組み込みデータベースの起動，停止，およびステータスの検査ができます。

実行中，このプログラムは，データベースの接続性を 5 分間隔でテストし，接続性テストの結果に応じて ovspmd プロセスに報告するステータスメッセージを更新します。データベースのテストが成功したときに報告されるメッセージはDatabase Available です。データベースのテストが失敗したときに報告されるメッセージはData Warehouse inaccessible です。失敗のメッセージが表示された場合，NNMi にデータベース関連の問題が発生している可能性があります。例えば，NNMi コンソールでノードインベントリの取得・表示ができないなどです。

このコマンドをovspmd プロセスと独立して実行することも技術的に可能ですが，予期しない結果になるおそれがあるので，この方法は推奨されません。

注意：nmsdbmgr プログラムの実行中にプロセステーブルを確認すると，このプログラムから，プラットフォームに応じてpostgres またはpostgres.exe という名前の子プロセスが多数生成されていることが分かります。これらの子プロセスは，組み込みデータベースそれ自体を表しています。これらの子プロセスのインスタンスが複数存在しても問題はありません。

EXAMPLES

このコマンドの通常の使用法は，ovspmd プロセスを使用した間接的な使用です。

典型的な使用法は，同プロセスを次のように起動または停止することです。

```
# ovstart -c nmsdbmgr
# ovstop -c nmsdbmgr
# ovstatus -c nmsdbmgr
```

AUTHOR

nmsdbmgr was developed by Micro Focus.

FILES

Windows：

%NNM_DB%¥Postgres

%NnmInstallDir%nonOV¥Postgres

Linux：

$NNM_DB/Postgres

$NnmInstallDir/nonOV/Postgres

SEE ALSO

ovspmd，ovstart，ovstop，ovstatus.

付録 G.3　nnmaction

NNMi Action Server（アクションサーバー）のラッパープロセス

SYNOPSIS

nnmaction

DESCRIPTION

nnmaction はアクションサーバーと呼ばれるプロセスで，ovspmd プロセスによって管理されます。nnmaction.properties ファイルにエントリを追加することで，アクションサーバーに引数が渡されます。

nnmaction コマンドは絶対に手動で実行しないでください。nnmaction プロセスの起動および管理は，ovspmd プロセスが行います。nnmaction プロセスを再起動するには，ovstop nnmaction コマンド，ovstart nnmaction コマンドの順に実行します。nnmaction プロセスのステータスを確認するには，ovstatus nnmaction コマンドを実行します。

ovstart コマンドまたはovstop コマンドを実行するには，Windows システムの管理者または Linux システムの root としてログオンする必要があります。

AUTHOR

nnmaction was developed by Micro Focus.

FILES

- Windows：%NNM_SHARED_CONF%¥props¥nnmaction.properties
- Linux：$NNM_SHARED_CONF/props/nnmaction.properties
  アクションサーバーが使用するパラメータファイルです。

SEE ALSO

ovspmd，ovstart，ovstop，ovstatus.

付録 G.4　nnmbackup.ovpl

NNMi のデータおよびファイルのバックアップに使用するコマンドです。

SYNOPSIS

nnmbackup.ovpl [-?|-h|-help] [-type (online|offline)] [-scope (config|topology|events|all)] [-force] [-archive] [-noTimeStamp] -target <*directory*>

DESCRIPTION

nnmbackup.ovpl は，NNMi の主要なバックアップコマンドです。このコマンドは，バックアップ動作に対してどのディレクトリとテーブルを考慮するかを決定するために%NNM_DATA%¥shared¥nnm¥backup.properties ファイル（Windows）または$NNM_DATA/shared/nnm/backup.properties ファイル（Linux）を使用します。backup.properties ファイルは，リストア時に特別な処理が必要なファイルやディレクトリも定義しています。nnmbackup.ovpl コマンドは，バックアップ領域（config, topology, events, all），バックアップロケーション，バックアップタイプ（online またはoffline）などの項目を決定するために，種々の引数を指定します。

nnmbackup.ovpl コマンドを使用して NNMi のバックアップを作成し，次にnnmrestore.ovpl コマンドを使用してデータベースレコードを別の NNMi 管理サーバーに配置する場合は，どちらの NNMi 管理サーバーも同種のオペレーティングシステムと同一バージョンの NNMi がインストールされ，同一のパッチレベルが適用されている必要があります。バックアップデータをある NNMi 管理サーバーから別の NNMi 管理サーバーに配置するということは，どちらのサーバーも同一のデータベース UUID を持つことを意味します。グローバルネットワーク管理機能を使用する場合は，2 台目の NNMi 管理サーバーに NNMi をリストアした場合，元の NNMi 管理サーバーから NNMi をアンインストールしてください。

このコマンドを実行する前に，ターゲットディレクトリに十分な保存領域があることを確認してください。ほとんどの NNMi インストール環境において，NNMi インストール環境（Windows では%NNM_DATA%ディレクトリ，Linux では$NNM_DATA ディレクトリ）のコンテンツを保存するだけの十分な空き容量があれば，保存領域も十分に取れているはずです。次のディレクトリで，使用可能な保存領域を確認してください。

- Windows：%NnmInstallDir%
- Linux：$NnmInstallDir

組み込みデータベースのデータは，%NNM_DATA%¥shared¥nnm¥databases¥Postgres ディレクトリ（Windows）または$NNM_DATA/shared/nnm/databases/Postgres ディレクトリ（Linux）に格納されます。

ターゲットディレクトリを構成するのは，指定されたバックアップオプションに適用できるファイルのすべてか，-archive オプションを使用する場合は単独の tar ファイルです。各バックアップ動作では，指定

されたターゲットディレクトリの下の親ディレクトリnnm-bak-<*TIMESTAMP*>にファイルを保存します。バックアップの実行中に何かデータベース操作が行われた場合，その操作はバックアップ対象となります。バックアップ完了後にファイルを圧縮することができます。

リストア時に特別な処理が必要なファイルは，ターゲットディレクトリ/nnm-bak-<*TIMESTAMP*>/special_filesの下に保存されます。リストア時は，除外，リストア，またはマージ対象のファイルがNNMiによって選択されます。詳細については，nnmrestore.ovplリファレンスページを参照してください。

nnmbackup.ovplコマンドには，リストア操作の実行に必要なデータが含まれています。このコマンドを実行するには，WindowsシステムのAdministrator権限またはLinuxシステムのrootでログインする必要があります。

Parameters

nnmbackup.ovplコマンドは，次のオプションをサポートします。

-type (online|offline)

このオプションでは，実行するバックアップのタイプを決定します。onlineオプションを指定する場合，nnmbackup.ovplコマンドを実行するときに，NNMiが稼働中である必要があります（所要のプロセスはnmsdbmgrのみ）。offlineオプションを指定する場合，NNMiを完全に停止してからnnmbackup.ovplコマンドを実行します。

-scope (config|topology|events|all)

このオプションでは，バックアップ操作の領域を指定します。バックアップ対象のデータには，ファイルシステム内のファイルとデータベース内のテーブルの2種類があります。ファイルシステム内のファイルに使用する-scopeオプションの値は，選択するバックアップタイプに関わらず常に適用されます（-typeオプションの項を参照）。ただし，データベース内のテーブルに使用する-scopeオプションの値は，-type onlineオプションを使用してオンラインバックアップを実行する場合にだけ適用されます。オフラインバックアップの場合は，要求した領域だけでなく，データベースの内容全体がバックアップされます。このため，オフラインバックアップを実行する場合は，領域を指定しないことを推奨します（デフォルトはallです）。指定可能な領域は，config，topology，events，およびallです。各領域には，下位の領域のデータとファイルが含まれます（all→events→topology→config）。%NNM_DATA%¥shared¥nnm¥backup.propertiesファイル（Windows）または$NNM_DATA/shared/nnm/backup.propertiesファイル（Linux）には，各領域でバックアップされるファイルおよびテーブルの一覧が指定されます。

-force

このオプションを指定すると，要求したバックアップのタイプに応じてて，nnmbackup.ovplコマンドがNNMiを起動および停止します。オンラインバックアップの場合，NNMiが実行中でないときは，nmsdbmgrプロセスが起動されます（バックアップに必要なため）。オフラインバックアップの場合，NNMiが実行中であれば，すべてのNNMiプロセスを停止します。

-archive

このオプションを指定すると，バックアップファイルがターゲットディレクトリにtarファイルとして保存されます。

-noTimeStamp

このオプションを指定すると，nnmbackup.ovpl コマンドは，タイムスタンプの入らない名前（単に nnm-bak というディレクトリ名または nnm-bak.tar というファイル名）でバックアップファイルを対象ディレクトリに保存します。同一のファイル名やディレクトリ名を持つ，以前のバックアップが存在する場合は，.previous というサフィックスが付けられてリネームされます。.previous の付いたバックアップが既に存在する場合，そのファイルまたはディレクトリは削除されます。このオプションの目的は，日常的なバックアップ作業において，実行したすべてのバックアップを保存する代わりに，成功した最新の二つのバックアップのみを維持することによって，ディスク使用量を削減することにあります。

-target <*directory*>

バックアップファイルを保存する出力ディレクトリを指定します。ターゲットディレクトリ内にnnm-bak-<*TIMESTAMP*>という親ディレクトリが作成され，すべてのバックアップファイルがこのディレクトリに保存されます。-archive オプションが指定されている場合，一時的にnnm-bak-<*TIMESTAMP*>ディレクトリが作成された後， このディレクトリが同じ名前の tar ファイルに置き換えられます。

-?|-h|-help

コマンドの使用方法を表示します。

EXAMPLES

ネットワークの検出結果は保存せずに，検出前の NNMi の設定を保存したい場合，online オプションと config オプションを使用して，次のようにバックアップを実行します。

```
#./nnmbackup.ovpl -type online -scope config -target /tmp/bak/config
```

アプリケーションを停止せずに，NNMi の設定，検出されたトポロジ，およびイベントデータを保存する場合，online，events，およびforce オプションを使用して，次のようにバックアップを実行します。

```
#./nnmbackup.ovpl -force -type online -scope events -target /tmp/bak/evt
```

計画されたバックアップを定期的に実行する場合，offline オプションとall オプションを使用して，次のようにバックアップを実行します。

```
#./nnmbackup.ovpl -type offline -scope all -target /tmp/bak/all
```

AUTHOR

nnmbackup.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_DATA%¥shared¥nnm¥backup.properties
- Linux：$NNM_DATA/shared/nnm/backup.properties

SEE ALSO

nnmrestore.ovpl.

付録 G.5　nnmbackupembdb.ovpl

NNMi 組み込みデータベースの完全なバックアップを取得します。

SYNOPSIS

nnmbackupembdb.ovpl [-?|-h|-help] [-noTimeStamp] -target <*target directory*> [-force]

DESCRIPTION

NNMi 組み込みデータベースの完全なバックアップを取得するときに，nnmbackupembdb.ovpl コマンドを使用します。バックアップの内容は，指定したファイルに圧縮されない状態で格納されます。このファイルは，nnmrestoreembdb.ovpl コマンドでリストアを行うときに必要となります。

nnmbackupembdb.ovpl コマンドを使用して NNMi 組み込みデータベースのバックアップを作成し，次に nnmrestoreembdb.ovpl コマンドを使用して組み込みデータベースレコードを別の NNMi 管理サーバーに配置する場合は，どちらの NNMi 管理サーバーも同種のオペレーティングシステムと同一バージョンの NNMi がインストールされ，同一のパッチレベルが適用されている必要があります。

nnmbackupembdb.ovpl コマンドを実行する前に，バックアップ先ディレクトリに十分な空き容量があることを確認してください。組み込みデータベースのデータは，%NNM_DATA%¥shared¥nnm¥databases¥Postgres ディレクトリ（Windows）または$NNM_DATA/shared/nnm/databases/Postgres ディレクトリ（Linux）に保存されます。バックアップ先ディレクトリにバックアップファイルを格納するために必要な空き容量があるか確認するために，このディレクトリのサイズをチェックしてください。必要に応じて，バックアップ後にファイルを圧縮してもかまいません。

バックアップ先ディレクトリに作成されるバックアップファイルには，バックアップ操作を開始したときのデータベースの内容が格納されます。バックアップ実行中にデータベース内で実行されるステートメントはバックアップ対象外です。

このコマンドは，NNMi の稼動中にも実行できますが，一時的にパフォーマンスが低下する場合があります。このコマンドが正常に実行されるには，最低限nmsdbmgr プロセスが起動している必要があります。-force オプションを指定すると，このコマンドによって（起動していない場合は）nmsdbmgr プロセスが起動し，対話式のメッセージは非表示となります。

このコマンドを実行するには，Windows システムの管理者または Linux システムの root としてログインする必要があります。

Parameters

-target <*target directory*>

バックアップファイルを保存するディレクトリの名前を指定します（存在しないディレクトリを指定した場合は作成されます）。

-force

このオプションを指定すると，nmsdbmgr プロセスが起動していない場合は自動的に起動します。

-noTimeStamp

このオプションを指定すると，nnmbackupembdb.ovpl コマンドは，タイムスタンプの入らない名前（単に nnm-bak.pgd というファイル名）でバックアップファイルを対象ディレクトリに保存します。同一のファイル名を持つ，以前のバックアップが存在する場合は，.previous というサフィックスが付けられてリネームされます。.previous の付いたバックアップが既に存在する場合，そのファイルは削除されます。このオプションの目的は，日常的なバックアップ作業において，実行したすべてのバックアップを保存する代わりに，成功した最新の二つのバックアップのみを維持することによって，ディスク使用量を削減することにあります。

--?|-h|-help

コマンドの使用方法を表示します。

EXAMPLES

このコマンドを使用すると，"オンデマンド"バックアップを実行できます。また，バックアップコマンドの実行を，定期的に実行されるタスクとして追加することもできます。次のように実行します。

```
# nnmbackupembdb.ovpl -target /backups/nnm

警告: 許可されていても、NNMの実行中にこのコマンドを実行すると
        一時的なパフォーマンスの問題が発生します。少なくとも、nmsdbmgrプロセスが
        実行している（ovstart nmsdbmgr）ことを確認してください。

フルデータベースバックアップの実行を今すぐ開始してもよいですか? [n]
y
組み込みデータベースのフルバックアップを実行しています...


NNM組み込みデータベース nnm は、/backups/nnm/nnm-bak-20070929064743.pgd に正常にバックアップされました。


#
```

AUTHOR

nnmbackupembdb.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmbackupembdb.ovpl
- Linux：$NNM_BIN/nnmbackupembdb.ovpl

SEE ALSO

ovstart, ovstop, ovstatus, nmsdbmgr, nnmrestoreembdb.ovpl.

付録 G.6　nnmcertmerge.ovpl

自動でキーストアとトラストストアを証明書ストアにマージします。

SYNOPSIS

nnmcertmerge.ovpl [-?|-h|-help] [-keystore <*file*> -truststore <*file*>]|[-directory <*directory*>]

DESCRIPTION

証明書ストアをnnm.keystoreおよびnnm.truststoreファイルに自動でマージするときに，nnmcertmerge.ovplを使用します。このコマンドは，グローバルネットワーク管理，HA構成，およびアプリケーションフェイルオーバーの機能を使用する場合に，すべての証明書のマージ作業を簡単にします。

このコマンドを実行するには，Windowsシステムの管理者またはLinuxシステムのrootとしてログインする必要があります。

Parameters

nnmcertmerge.ovplコマンドは，次のオプションをサポートします。

-keystore <*file*>

このオプションを使用すると，指定したファイルがnnm.keystoreファイルにマージされます。-truststoreオプションと同時に使用できます。

-truststore <*file*>

このオプションを使用すると，指定したファイルがnnm.truststoreファイルにマージされます。-keystoreオプションと同時に使用できます。

-directory <*directory*>

-directoryオプションは単独で使用する必要があります。このオプションを使用すると，指定したディレクトリにあるすべてのファイルが，次のように処理されます。
.keystoreで終わるすべてのファイルがnnm.keystoreファイルにマージされます。
.truststoreで終わるすべてのファイルがnnm.truststoreファイルにマージされます。

-?|-h|-help

コマンドの使用方法を表示します。

EXAMPLES

キーストアを NNMi にマージします。

```
nnmcertmerge.ovpl -keystore /tmp/hostA.keystore
```

トラストストアを NNMi にマージします。

```
nnmcertmerge.ovpl -truststore /tmp/hostA.truststore
```

キーストアとトラストストアを NNMi にマージします。

```
nnmcertmerge.ovpl -keystore /tmp/hostA.keystore -truststore /tmp/hostA.truststore
```

ディレクトリにあるすべてのキーストアとトラストストアを NNMi にマージします。

```
nnmcertmerge.ovpl -directory /tmp/AppFailoverHosts/
```

AUTHOR

nnmcertmerge.ovpl was developed by Micro Focus.

FILES

nnmcertmerge.ovpl は，次のディレクトリにあります。

- Windows：%NNM_BIN%
- Linux：$NNM_BIN

付録 G.7　nnmchangeembdbpw.ovpl

NNMi 組み込みデータベースの認証に使用されているパスワードを変更します。

SYNOPSIS

nnmchangeembdbpw.ovpl

DESCRIPTION

nnmchangeembdbpw.ovpl を使用すると，NNMi が組み込みデータベースに接続する際に使用するデータベースパスワードを変更できます。このコマンドは，組み込みデータベースオプションで NNMi をインストールした場合に有効です。このコマンドは，組み込みデータベース用に作成されたデフォルトパスワードを変更するときにだけお使いください。このコマンドを実行しなくても NNMi は正常に機能します。

nnmchangeembdbpw.ovpl コマンドを実行するには，nmsdbmgr だけが起動していて，ほかの NNMi プロセスは停止している必要があります。このコマンドは，必要に応じて自動的にシステムをこの状態にします。つまり，自動的にovstop を実行してからovstart nmsdbmgr を実行します。

このコマンドが完了すると，組み込みデータベースパスワードは対話式に入力された値に変更され，NNMiサーバーは新しいパスワードを使用してデータベースに接続するよう再設定されます。

このコマンドを実行するには，Windows システムの管理者または Linux システムの root としてログインする必要があります。

Parameters

サポートするパラメータはありません。

EXAMPLES

NNMi 組み込みデータベースのパスワードを頻繁に変更する必要がある場合，nnmchangeembdbpw.ovpl コマンドを使用します。

nnmchangeembdbpw.ovpl コマンドでは，入力したパスワードは表示されません。

nnmchangeembdbpw.ovpl コマンドを使用した場合，次のメッセージが表示されます。

```
# nnmchangeembdbpw.ovpl
警告：このツールを実行する前に、NNMが停止したことを確認してください。
          NNMを停止できない場合、予期しないデータベースエラーが発生する可能性があります。
          NNMを停止した後、ovstart nmsdbmgrを実行してデータベースのみを開始します。
NNMを停止しましたか (ovstop)? [Y/N]:
Y
組み込みデータベースのパスワードの変更を続行します
パスワードを入力してください:
mynewpw
パスワードを再入力してください:
mynewpw
ユーザー/パスワードが正常に変更されました!
#
```

AUTHOR

nnmchangeembdbpw.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmchangeembdbpw.ovpl
- Linux：$NNM_BIN/nnmchangeembdbpw.ovpl

SEE ALSO

ovstart, ovstop, ovstatus, nmsdbmgr.

付録 G.8　nnmchangesyspw.ovpl

NNMi のインストール時に通常設定されるシステムアカウントのパスワードを変更します。

SYNOPSIS

nnmchangesyspw.ovpl

DESCRIPTION

nnmchangesyspw.ovpl コマンドを使用すると，NNMi システムパスワードを変更できます。通常，NNMi システムパスワードはインストール時に設定され，復旧を行うときに使用されます。このコマンドは，インストール時に設定されたシステムパスワードをリセットする場合にだけ使用します。

nnmchangesyspw.ovpl コマンドを実行する前に，ovstop コマンドを実行して NNMi を停止してください。nnmchangesyspw.ovpl コマンドの実行後，ovstart コマンドを実行して NNMi を起動します。この操作により，新しいパスワード値が即座に有効になります。

このコマンドを実行するには，Windows システムの管理者または Linux システムの root としてログインする必要があります。

Parameters

サポートするパラメータはありません。

EXAMPLES

管理者権限が付与されたほかのユーザーアカウントをすべて削除してしまった上，NNMi のインストール時に設定したシステムパスワードを覚えていないような場合に，システムパスワードを変更します。

nnmchangesyspw.ovpl コマンドでは，入力したパスワードは表示されません。

nnmchangesyspw.ovpl コマンドを使用した場合，次のメッセージが表示されます。

```
# nnmchangesyspw.ovpl

警告: この変更は NNM が再起動されない限り直ちには反映
          されません。このスクリプトを実行する前に ovstop を実行し、
          実行後に ovstart を実行して変更が即時に反映されるようにしてください。
続行しますか?[Y/N]:
Y
システムのパスワードの変更を続行します
パスワードを入力してください:
mynewpw
パスワードを再入力してください:
mynewpw
システム パスワードが正常に変更されました
#
```

AUTHOR

nnmchangesyspw.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmchangesyspw.ovpl
- Linux：$NNM_BIN/nnmchangesyspw.ovpl

SEE ALSO

ovstart, ovstop, ovstatus.

付録 G.9　nnmcluster

NNMi クラスタサービスを開始します。

SYNOPSIS

nnmcluster [-disable|-enable] [-display] [-interfaces] [-startnnm|-stopnnm] [-acquire|-relinquish] [-shutdown] [-dbsync] [-halt] [-node *nodename*] [-daemon]

DESCRIPTION

nnmcluster は，NNMi クラスタプロセスを開始します。NNMi クラスタは，二つのシステムを使用することによって，一方のシステムに障害が発生した場合の NNMi サービスの運用を保証します。両方のノードでnnmcluster コマンドを実行すると，両ノードが互いに相手を検出し，クラスタを形成します。先にクラスタに参加したノードは"アクティブ"状態となり，（ovstart コマンドにより）NNMi サービスを開始します。後でクラスタに参加したノードは，既にアクティブなノードが存在していることを検出し，"スタンバイ"状態となります。システムのシャットダウンや障害などで，スタンバイノードとアクティブノードとの接続が切断されると，スタンバイノードはアクティブ状態となり，NNMi サービスを開始します。

nnmcluster をコマンドラインパラメータなしで呼び出すと，クラスタは対話モードで起動します。対話モードでは，クラスタの設定を対話式の操作で表示したり変更したりできます。対話モードで表示したり変更したりできる設定には，自動フェイルオーバーの有効化/無効化の切り替え，クラスタ内ノードのシャットダウン，NNMi サービスのアクティブノードからスタンバイノードへの移行などがあります。

nnmcluster が -daemon パラメータを付けて呼び出された場合，NNMi クラスタはバックグラウンドデーモンプロセス，または，Windows サービスとして起動します。

ほかのパラメータを指定してnnmcluster コマンドを呼び出した場合，コマンドラインに指定されたアクションが開始されます。これらのアクションは通常，ローカルノードの NNMi クラスタデーモンプロセスを対象としますが，-node *nodename* オプションを使用すると，指定したノードの NNMi クラスタデーモンプロセスが対象となります。

コマンドラインに指定できるオプションのほとんどは対話モードでも使用できます。例えば，コマンドラインでの"-shutdown"オプションの指定は，対話モードでの”shutdown”コマンドに対応します。対話モードだけで使用できるコマンドもいくつかあります。例えば，使用できるコマンドを一覧表示する"help"コマンドや，対話モードを終了する"quit"コマンドがその一例です。"-node *nodename*"コマンドも対話モードに用意されています。

なお，NNMi サービスを開始できるのは，NNMi クラスタデーモンプロセスだけです。対話モードおよびコマンドラインのアクション指定は，クラスタ内ノードの一方のデーモンプロセスの動作を実行するための方法です。例えば，"-acquire"を指定した場合は，ローカルノード（-node オプションでノードを指定した場合は指定されたノード）上のデーモンプロセスが，アクティブな状態を取得し，NNMi サービスを開始します。いったん開始された NNMi クラスタデーモンプロセスとのやりとりは，コマンドラインまたは対話モード設定を通じて行います。例えば，開始された NNMi クラスタデーモンプロセスを終了したい場合は，nnmcluster -shutdown コマンドを使用します。

NNMi クラスタアプリケーションは，アクティブノードとスタンバイノードとの間で組み込みデータベースを同期します。この動作は，データベースの完全なバックアップをスタンバイノードに送信した後，定期的に増分データベーストランザクションログを送信することで実現されます。完全バックアップとトランザクションログを送信する頻度（間隔）は，ほかのクラスタパラメータと一緒にnms-cluster.properties ファイルに定義します。

NNMi クラスタアプリケーションにはスタートアップ期間があり，この期間を経てからデータベースをアクティブノードからスタンバイノードへ送信できるようになります。 このスタートアップ期間の間は，アクティブな状態をスタンバイノードへ移動させるコマンドオプションが無効になります。このようなオプションとしては，shutdown，acquire，relinquish，その他の使用可能オプションがあります。これらのオプションにより，スタンバイノードのデータベースが不完全な状態となり，NNMi を実行できない状態になるため，これらは無効にされます。しかしながら，スタンバイノードがデータベース全体を受信すると，その時点以降に問題となる期間はなく，両方のシステムが動作を継続します（再起動を行うと，スタンバイノードがアクティブノードと同期状態であるかどうかを検証し直します）。

Parameters

nnmcluster コマンドは，次のオプションをサポートします。サポートされないオプションを指定した場合，使用方法のメッセージが表示されます。オプションは，常に指定した順序で処理されます。例えば，"-display -disable"と指定する場合と，"-disable -display"と指定する場合は，結果が異なります。

-node *nodename*

オプションを指定時以外は，すべてのパラメータがローカルノード上の NNMi クラスタデーモンプロセスを参照します。

-disable

自動フェイルオーバー機能を無効にします（自動フェイルオーバー機能はデフォルトでは有効となります）。システム管理者は，管理作業を行うため短時間，アクティブノードをシャットダウンする場合があります。-disable パラメータを指定すると，スタンバイノードをアクティブにして NNMi サービス

を起動することなく，アクティブノードをシャットダウンできます。-acquire オプションに続けて-enable オプションを指定すると，同じノードをアクティブ状態のまま再起動できます。

-enable

上述の方法で無効化された自動フェイルオーバー機能を再び有効化します。

-interfaces

システムに搭載されているネットワークインタフェース（NIC）の一覧を作成し，システムの命名規則と Java の命名規則を表示します。Linux プラットフォームでは，"eth0"，"lan1"，"bge3"など，両者は同じ値になります。Windows では，"Network Interface 1"が"eth3"にマッピングされるなど，名称が異なります。 このオプションの目的は，データ NIC の代わりに管理 NIC を選択するなど，NNMi クラスタの通信で使用する NIC を制御することです。上記の Windows の例では，NNMi クラスタは"eth3"などの Java 名を知っている必要があります。

-display

クラスタに接続し，現在のクラスタ状態を問い合わせ，状態を管理者に対して表示します。

-startnnm

-stopnnm オプションで NNMi サービスを停止した場合など，NNMi サービスがアクティブノードで稼働しているとは限りません。-startnnm オプションはこれらの NNMi サービスをアクティブノードで起動します。

-stopnnm

アクティブノードの NNMi サービスをシャットダウンしますが，"アクティブ"状態は解除されません。このオプションを指定した場合，フェイルオーバーイベントは発生しません。つまり，スタンバイノードにアクティブ状態が移行するわけではありません。

-acquire

システム管理者は，現在"アクティブ"状態のノードから"スタンバイ"状態のほかのノードに NNMi サービスを移行する場合があります。-node オプションでノードを指定しない場合は，ローカルシステムが新しいアクティブなノードとなります。-node *nodename* オプションでノードを指定した場合は，指定したノードが新しいアクティブなノードとなります。

-relinquish

ローカル（アクティブ）ノードのアクティブ状態を放棄し，NNMi サービスをスタンバイノードに移行するためのオプションです。アクティブ状態を放棄するノードは NNMi サービスを停止し，スタンバイ状態となります。

-dbsync

アクティブなノード上でデータベースをバックアップして，スタンバイなノードのデータベースへ同期させます。

-shutdown

ローカルノード上の NNMi クラスタデーモンプロセスをシャットダウンします。-node オプションでノードを指定した場合は，指定したノード上の NNMi クラスタデーモンプロセスをシャットダウンします。NNMi クラスタプログラムは，クラスタのシャットダウンによってスタンバイノードのデータ完全性が損なわれる可能性がある場合は，クラスタのシャットダウンを禁止します。例えば，スタンバ

イノードがアクティブノードからデータベースの完全なバックアップを受信しているときに，フェイルオーバーイベントが生じるのは好ましくありません。シャットダウンを行ったときに，クラスタが移行状態にあることを通知するメッセージを受信することがあります。つまり，スタンバイノードが重要なデータを受信しているため，それが完了してからシャットダウンを行う必要があるということです。また，アクティブノードが存在しない状態でスタンバイノードをシャットダウンすることはできませんので注意してください。アクティブノードとスタンバイノードの両方を停止させる場合は，-halt オプションの使用を推奨します。このオプションでは，スタンバイノードが停止してからアクティブノードが停止します。

-halt

クラスタ内のすべてのノードの NNMi クラスタデーモンプロセスをシャットダウンします。このオプションを指定した場合，フェイルオーバーを無効にしてからすべてのスタンバイノードをシャットダウンし，最後にアクティブノードをシャットダウンします。

-node *nodename*

コマンドラインで指定された一つまたは複数のアクションが，指定したノード上の NNMi クラスタデーモンプロセスに実行されます。このオプションを指定しない場合，コマンドラインで指定されたアクションは，ローカルノード上の NNMi クラスタデーモンに適用されます。

-daemon

NNMi クラスタをデーモンとして起動します。コマンドはバックグラウンドで即時実行されます。デーモンモードでは，コマンドラインにほかのパラメータを指定できません。

RETURN VALUE

コマンドラインオプションを指定したnnmcluster コマンドが成功した場合，コマンドはステータス 0（ゼロ）で終了します（エラーなし）。一方，コマンドラインオプションを指定したnnmcluster コマンドが失敗した場合，コマンドはステータス 1 で終了します（エラー発生）。対話モードでは，終了ステータスは常に 0 となります。

デーモンモードでは，nnmcluster コマンドはバックグラウンドプロセスとして動作し，シェルプロンプトが即座に返されます。デーモンプロセスを開始した場合，クラスタの状態はnnmcluster -display コマンドまたはovstatus コマンドで確認できます。つまり，（クラスタ内のほかのノードとの兼ね合いで，ノードのアクティブ/スタンバイ状態により）NNMi クラスタは NNMi サービスを開始するタイミングを判定します。ovstatus コマンドは，スタンバイノードの"稼働していない"状態を通知しますが，nnmcluster -display コマンドはノードがスタンバイ状態にあることを通知します。

DIAGNOSTICS

nnmcluster は，NNMi ログディレクトリ（Windows の場合は%NNM_DATA%¥log¥nnm，Linux の場合は $NNM_DATA/log/nnm）にログを出力します。デーモンの動作モードが対話モードであってもコマンドラインであっても，動作中のnnmcluster プロセスのアクティブなインスタンスごとに個別のログファイルが生成されます。最新の実行スレッドは常に"nnmcluster.0.*.log"となります。NNMi クラスタは内部的に"JGroups"と呼ばれるオープンソースの技術を使用します。JGroups のログファイルは，"jgroups.log"という名前で上記のディレクトリに保存されます。

EXAMPLES

```
nnmcluster -daemon
```

```
nnmcluster -display
```

最初のコマンドを実行すると，NNMi クラスタがデーモンプロセスとして起動し，シェルプロンプトが即座に返されます。デーモンプロセスはバックグラウンドで実行されます。対話モードまたはコマンドラインモードを使用すると，このデーモンプロセスに対して照会，シャットダウン，またはその他のアクションの適用を行うことができます。2 番目は繰り返して使用するコマンドで，クラスタ（特にローカルデーモンプロセス）の状態を監視し，起動状態がアクティブかスタンバイかを確認できます。

```
nnmcluster -shutdown -node xyz.mycompany.com
```

指定したノード上の NNMi クラスタデーモンプロセスをシャットダウンします。 指定したノードがアクティブノードの場合，自動フェイルオーバーが有効のときは，NNMi サービスはスタンバイノードに移行します。

```
nnmcluster
```

NNMi クラスタコマンドの対話モードを開始し，クラスタパラメータを表示または変更します。このプログラムを終了するには，"exit"コマンドまたは"quit"コマンドを使用します。

```
nnmcluster -acquire
```

デーモンモードで動作している，ローカルシステムの NNMi クラスタプロセスをアクティブノードに切り替えます。現在アクティブなノードがスタンバイモードに設定され，NNMi サービスがローカルノードで開始されます。

一般的なシステム管理手順を次に示します。管理者は，アクティブノードの NNMi クラスタを一時的にシャットダウンし，スタンバイノードへのフェイルオーバーイベントを発生させずに，アクティブノードをアクティブノードのまま後で再起動します。

手順 1：nnmcluster -disable -shutdown コマンドを実行します。

手順 2：必要なシステム管理タスクを実行します。

手順 3：nnmcluster -daemon コマンドを実行します。

手順 4：nnmcluster -display コマンドを実行します。このコマンドを使用して，デーモンがいつ起動するかを確認します。

手順 5：nnmcluster -enable コマンドを実行します。

手順 1 に示すコマンドを実行すると，まずフェイルオーバーが無効になり，ローカルデーモンプロセスがシャットダウンされます。手順 2 では，フェイルオーバーをトリガーするおそれがない状態で，システム管理者が管理タスクを実行します。手順 3 に示すコマンドにより，デーモンモードの NNMi クラスタプロセスが再起動します。手順 4 に示すコマンドを繰り返し使用すると，ローカルデーモンプロセスが起動し

て NNMi が実行されるタイミングを確認できます。手順 5 に示すコマンドで，アクティブノードの NNMi が起動した後に自動フェイルオーバーを再度有効にします。

AUTHOR

nnmcluster was developed by Micro Focus.

FILES

- Windows：%NNM_PROPS%¥nms-cluster.properties
- Linux： $NNM_PROPS/nms-cluster.properties

このファイルはクラスタパラメータを定義します。具体的には，クラスタに一意な名前を付け，同じネットワークに存在するほかの NNMi クラスタと区別する必要があります。オプションで，タイムアウトなどのほかのパラメータを設定できます。

SEE ALSO

ovstart, ovstop.

付録 G.10　nnmcommconf.ovpl

通信の構成情報を表示します。

SYNOPSIS

nnmcommconf.ovpl [-u *username*] [-p *password*] -proto <*icmp* | *snmp*> -host <*hostname*>

DESCRIPTION

nnmcommconf.ovpl は，特定のプロトコルを使用して，特定のホストと NNMi がどのように通信を行うかについて，NNMi から情報を読み取り，表示するコマンドです。nnmcommconf.ovpl は，SNMP または ICMP プロトコルに基づいて情報を表示します。

Parameters

nnmcommconf.ovpl コマンドは，次のオプションをサポートします。

-proto <*protocol*>

　プロトコル：SNMP または ICMP

-host <*hostname*>

　情報の取得元となるホストの名前を指定します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

```
nnmcommconf.ovpl -u foo -p bar -proto icmp -host baz
```

このコマンドを実行すると，次のような情報を取得できます。

```
       address              = 10.2.1.2
      timeout              = 2000
      retries              = 1
      enabled              = true
      region name          = default
nnmcommconf.ovpl -u foo -p bar -proto snmp -host baz
```

このコマンドを実行すると，次のような情報を取得できます。

```
       name                  = baz
      management address     = 10.2.1.1
      addressForced          = false
      preferredVersion       = null
      minimum security level = コミュニティのみ（SNMPv1 またはv2c）
      readCommunity          = public
      writeCommunity         = *******
      timeout                = 5000
      retries                = 1
      port                   = 161
      proxyPort              = null
      proxyAddress           = null
      SNMPv3 user names      = null
      SNMPv3 context names   = null
      enabled                = true
      address disco enabled  = false
      get bulk enabled       = true
      region name            = region
      node setting description = null
      active address         = 10.2.1.1
      active readCommunity   = public
      active SNMPv3 user     = null
      active SNMPv3 context  = null
```

AUTHOR

nnmcommconf.ovpl was developed by Micro Focus.

FILES

nnmcommconf.ovpl は，次のディレクトリにあります。

- Windows：%NNM_BIN%
- Linux：$NNM_BIN

SEE ALSO

nnm.properties, nnmcommload.ovpl.

付録 G.11　nnmcommload.ovpl

CSV ファイルから通信設定を読み込みます。

SYNOPSIS

nnmcommload.ovpl [-u *username*] [-p *password*] -file <*filepath* | *filename*>

DESCRIPTION

nnmcommload.ovpl コマンドを使用すると，複数のデバイスの通信設定を一括してインポートできます。このコマンドは，コミュニティ文字列が変更制御機構で管理されているときに有用です。NNMi に一括して指定項目を挿入できます。設定ファイルに入力したデータの形式に応じて，各指定項目が個別エントリとして，NNMi コンソールの［通信の設定］フォームの［領域］タブまたは［特定ノードの設定］タブに表示されます。

IP アドレスとしてホスト名を指定した場合，nnmcommload.ovpl コマンドは，IP アドレスを完全修飾名へと解決しません。実際のホスト名を指定すると，nnmcommload.ovpl コマンドは，DNS を使用してホスト名を完全修飾名へと解決します。サイズが大きいインポートファイルを処理するには，時間がかかる場合があります。500 行を超えるファイルについては，500 行のバッチに分けてエントリをデータベースに保存します。500 行をインポートファイルから読み込んだ後，各行の SNMP 構成エントリは既存の SNMP 領域またはデフォルト設定に基づいて解決され，データベースに保存されます。

インポートを実行するには，次に示す項目が，示されたとおりの順序で指定されたテキストファイルを作成します。デバイス一つにつき 1 行とします。1 行に指定する各項目はカンマで区切ります。「#」で始まる行はコメント行となります。行に指定された項目は厳密に指定位置によって解釈されるため，値のない項目についてもカンマを打つ必要があります。カンマを含む値を指定する場合は，二重引用符で囲みます（例："comm,string"）。

- ターゲットホスト名または IP アドレス（必須：［特定ノードの設定］の設定）
  ";;"で区切られた一つ以上のホスト名フィルター（任意：［領域］の設定）
- 一つの読み取りコミュニティ文字列（任意：［特定ノードの設定］の設定）
  ";;"で区切られた一つ以上の読み取りコミュニティ文字列（任意：［領域］の設定）

コミュニティ文字列に順序を設定したい場合は，"#PRI#"という文字列に続いて，各領域のコミュニティ文字列に割り当てる優先順位を指定します。例えば，"public#PRI#5"は，コミュニティ文字列"public"に順位番号 5 が割り当てられます。

- 管理アドレス（任意：[特定ノードの設定］の設定）
  ";;"で区切られた一つ以上のアドレス範囲（任意：[領域］の設定）
- 書き込みコミュニティ文字列（任意）
- ミリ秒単位のタイムアウト値（任意）
- リトライ数（任意）
- ポート（任意）
- プロキシアドレス（任意）
- プロキシポート（任意）
- ユーザー名（SNMP v3 任意：[特定ノードの設定］の設定）
  ";;"で区切られた一つ以上のユーザー名（SNMP v3 任意：[領域］の設定）
- コンテキスト名（SNMP v3 任意：[特定ノードの設定］の設定）
  ";;"で区切られた一つ以上のコンテキスト名（SNMP v3 任意：[領域］の設定）
- 認証プロトコル（SNMP v3 任意：[特定ノードの設定］の設定- MD5|SHA）
  ";;"で区切られた一つ以上の認証プロトコル（SNMP v3 任意：[領域］の設定）
- 認証パスフレーズ（SNMP v3 任意：[特定ノードの設定］の設定）
  ";;"で区切られた一つ以上の認証パスフレーズ（SNMP v3 任意：[領域］の設定）
- プライバシプロトコル（SNMP v3 任意：[特定ノードの設定］の設定- DES|3DES|AES|AES192|AES256）
  ";;"で区切られた一つ以上のプライバシプロトコル（SNMP v3 任意：[領域］の設定）
- プライバシパスフレーズ（SNMP v3 任意：[特定ノードの設定］の設定）
  ";;"で区切られた一つ以上のプライバシパスフレーズ（SNMP v3 任意：[領域］の設定）
- SNMP 優先バージョン（任意- 1|2|3：[特定ノードの設定］の設定限定）
- 「SNMP の通信を有効にする」フラグ（任意- true|false）
- 「SNMP アドレス再検出を有効にする」フラグ（任意- true|false）
- 「SNMP GetBulk を有効にする」フラグ（任意- true|false）
- 説明（任意：[特定ノードの設定］の設定または［領域］の設定）
- 「ICMP の通信を有効にする」フラグ（任意- true|false）
- ミリ秒単位の ICMP のタイムアウト値（任意）
- ICMP のリトライ数（任意）
- デバイスの資格証明のユーザー名（任意）
- デバイスの資格証明のパスワード（任意）

- デバイスの資格証明のタイプ（任意）
  現在サポートされているタイプは，"Shell"だけです。
- 領域の名前（任意：[領域] の設定 - 指定しない場合，Region15 のようにデフォルトで "Region" ＋順序が設定されます）
- 領域の順序（任意：[領域] の設定 - 指定しない場合，既存の領域の順序における最大値 ＋ 1 が設定されます）
- SNMP 最小セキュリティレベル（任意：[領域] の設定 - V1-ONLY|V1V2-ONLY|COMMUNITY|NOAUTH-NOPRIV|AUTH-NOPRIV|AUTH-PRIV）指定しない場合，「コミュニティ」が設定されます。

例えば，特定ノードの設定を読み込むための次のエントリはすべて有効と考えられます。

```
hostname
hostname,
hostname,,
hostname,public
hostname,,10.2.2.3,,1000,2,161
node1,community,10.3.7.96,writecommunity
node2, community,10.3.7.95（community の前のスペースは削除されます）
10.2.23.34,community,10.2.23.8
10.2.23.34,community,10.2.23.88,writecommunity,2000,2,161,10.56.22.199,162
```

次は，複数のホスト名フィルター，アドレス範囲，読み取りコミュニティ文字列，SNMPv3 設定，およびデバイスの資格証明を持つ領域の設定を読み込む例です。コミュニティ文字列の順序も割り当てられています。

```
testv3*;;cisco*.fc.usa.hp.com, region
description,true,3,3,user1,password1,Shell,myregion,10,Community
```

```
public;;readcommunity#PRI#1,3.3.3.3;;4.4.4.4,writecommunity,3330,3,161,7.7.7.7,777,v3User1;;v3User2,v3Context1;;v3Context2,MD5;;SHA,authPass1;;authPass2,AES;;DES,privPass1;;privPass2,3,true,true,true,my region description,true,3,3,user1,password1,Shell,myregion,10,Community
```

次は，複数の SNMPv3 設定を持つ領域の設定を読み込む例です。v3 と領域のパラメータだけが指定されています。

- v3User1 は認証なし，プライバシなしのユーザーです
- v3User2 は MD5 認証，プライバシなしのユーザーです
- v3User3 は SHA 認証，3DES プライバシを持つユーザーです

";;"を区切り文字として使用しているため，必要に応じて認証とプライバシ情報の指定が省略されていることに注意してください。

```
,,,,,,,,,v3User1;;v3User2;;v3User3,v3Context1;;;;v3Context3,;;MD5;;SHA,;;authPass2;;authPass3,;;;;3DES,;;;;privPass3,3,,,,,,,,,,myregion,10,Community
```

Parameters

nnmcommload.ovpl コマンドは，次のオプションをサポートします。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-file <*filepath* / *filename*>（必須）

読み込むデータを上述の形式で指定したファイルのファイル名またはフルパスを指定します。

EXAMPLES

```
nnmcommload.ovpl -u joe -p secret -file import.txt
```

```
nnmcommload.ovpl -u joe -p secret -file C:¥temp¥import.txt
```

```
nnmcommload.ovpl -u joe -p secret -file /tmp/import.txt
```

AUTHOR

nnmcommload.ovpl was developed by Micro Focus.

FILES

nnmcommload.ovpl は，次のディレクトリにあります。

- Windows：%NNM_BIN%
- Linux：$NNM_BIN

SEE ALSO

nnm.properties, nnmcommconf.ovpl.

付録 G.12　nnmcommunication.ovpl

通信設定の管理を可能にします。

SYNOPSIS

```
nnmcommunication.ovpl -h|-help
```

nnmcommunication.ovpl addCertificate (-default|-region (<*name*>|<*uuid*>)| -nodeSetting (<*name*>|<*uuid*>))(-cert <*cert_file*>)

nnmcommunication.ovpl addCommunity (-default|-region <*region*>) -community <*value*> [-ordering <*order*>]

nnmcommunication.ovpl addCredential (-default|-region <*region*>| -nodeSetting <*nodeName*>) -type (SHELL|VMWARE|CISCOACI) -username <*username*> -password <*password*>

nnmcommunication.ovpl addSnmpV3Setting (-default | -region (<*name*>|<*uuid*>)) -snmpV3Setting (<*name*>|<*uuid*>)

nnmcommunication.ovpl createAddressRange [-uuid <*uuid*>] -region <*region*> -range <*ip_range*>

nnmcommunication.ovpl createHostnameFilter -hostname <*pattern*> -region <*region*> [-uuid <*uuid*>]

nnmcommunication.ovpl createNodeSettings -name <*name*> [-addressDiscovery (true|false)] [-description <*description*>] [-icmpEnabled (true|false)] [-icmpRetries <*number*>] [-icmpTimeout <*timeout*>] [-managementAddress <*ip*>] [-shellCredential <*credential*>] [-shellUser <*username*>] [-snmpCommunity <*string*>] [-snmpEnabled (true|false)] [-snmpGetBulk (true|false)] [-snmpPort <*port*>] [-snmpPreferredVersion (V1|V2C|V3)] [-snmpProxyAddress <*ip*>] [-snmpProxyPort <*port*>] [-snmpRetries <*number*>] [-snmpTimeout <*timeout*>] [-snmpV3Setting (<*name*>|<*uuid*>)] [-snmpWriteCommunity <*community_string*>] [-uuid <*uuid*>]

nnmcommunication.ovpl createRegionSettings -name <*name*> -ordering <*order*> [-addressDiscovery (true|false)] [-addressFilter <*filter*>] [-description <*description*>] [-hostnameFilter <*filter*>] [-icmpEnabled (true|false)] [-icmpRetries <*number*>] [-icmpTimeout <*timeout*>] [-shellCredential <*credential*>] [-shellUser <*username*>] [-snmpCommunities <*strings*>] [-snmpEnabled (true|false)] [-snmpGetBulk (true|false)] [-snmpPort <*port*>] [-snmpProxyAddress <*ip*>] [-snmpProxyPort <*port*>] [-snmpRetries <*number*>] [-snmpSecurityLevel (COMMUNITY_ONLY|COMMUNITY|NO_AUTH_NO_PRIV|AUTH_NO_PRIV|AUTH_PRIV|COMMUNITY_ONLY_V1)] [-snmpTimeout <*timeout*>] [-snmpV3Settings (<*name*>|<*uuid*>)] [-snmpWriteCommunity <*community_string*>] [-uuid <*uuid*>]

nnmcommunication.ovpl createSnmpV3Settings -name <*name*> -username <*name*> [-authPass <*string*>] [-authProtocol (HMAC_MD5_96|HMAC_SHA_1|HMAC128_SHA_224|HMAC192_SHA256|HMAC256_SHA_384|HMAC384_SHA_512)] [-contextName <*string*>] [-privPass <*string*>] [-privProtocol (DES_CBC|TripleDES|AES_128|AES_192|AES_256)] [-uuid <*uuid*>]

nnmcommunication.ovpl delete (-region <*region*>| -nodeSetting <*nodeSetting*>| -snmpV3Setting <*v3Setting*>)

nnmcommunication.ovpl deleteAddressRange (-region (<*name*>|<*uuid*>) | -range <*ip_range*> | -uuid <*uuid*>)

nnmcommunication.ovpl deleteHostnameFilter (-hostname <*pattern*> -region <*region*> | -uuid <*uuid*>)

nnmcommunication.ovpl listAddressRanges [-range <*ip_range*>] [-region (<*name*>|<*uuid*>)] [-uuid <*uuid*>]

nnmcommunication.ovpl listCertificates (-default|-region (<*name*>|<*uuid*>)| -nodeSetting (<*name*>|<*uuid*>))

nnmcommunication.ovpl listCommunities (-default|-region <*region*>)

nnmcommunication.ovpl listDefaults

nnmcommunication.ovpl listCredentials (-default|-region <*region*>| -nodeSetting <*nodeName*>)[-type(SHELL|VMWARE|CISCOACI)]

nnmcommunication.ovpl listEffective-node (<*name*>|<*uuid*>)

nnmcommunication.ovpl listHostnameFilters (-hostname <*pattern*> -region <*region*>) |(-uuid <*uuid*>)

nnmcommunication.ovpl listNodeSettings -name <*name*>

nnmcommunication.ovpl listRegionSettings [-name <*name*>]

nnmcommunication.ovpl listSnmpAgentSettings [-node (<*name*>|<*uuid*>)]

nnmcommunication.ovpl listSnmpV3Settings [(-name <*name*>|-uuid <*uuid*>)]

nnmcommunication.ovpl listWebAgentSettings [-node (<*name*>|<*uuid*>)]

nnmcommunication.ovpl removeCertificate (-default|-region (<*name*>|<*uuid*>)| -nodeSetting (<*name*>|<*uuid*>)) (-uuid <*uuid*>|-fingerPrint <*fingerPrint*>| -serialNumber <*serialNumber*>)

nnmcommunication.ovpl removeCommunity (-default|-region <*region*>) (-community <*value*>|-ordering <*order*>) -uuid <*uuid*>: The object unique identifier.

nnmcommunication.ovpl removeCredential (-default|-region <*region*>| -nodeSetting <*nodeName*>) -type (SHELL|VMWARE|CISCOACI) [-username <*username*>] [-password <*password*>]

nnmcommunication.ovpl removeSnmpV3Setting (-default | -region (<*name*>|<*uuid*>)) -snmpV3Setting (<*name*>|<*uuid*>)

nnmcommunication.ovpl updateCredential (-default | -region <*region*>| -nodeSetting <*nodeName*>) -type (VMWARE|CISCOACI) (-username <*username*> | -password <*password*>)

nnmcommunication.ovpl updateDefaults [-addressDiscovery (true|false)] [-default] [-icmpRetries <*number*>] [-icmpTimeout <*timeout*>] [-interfaceMatcher <*value*>] [-managementAddressSelection <*ALG1,ALG2,ALG3*>] [-preferIPVersion (IPv4|IPv6|IPAny)] [-shellCredential <*credential*>] [-shellUser <*username*>] [-snmpCommunities <*strings*>] [-snmpGetBulk (true|false)] [-snmpPort <*port*>] [-snmpProxyAddress <*ip*>] [-snmpProxyPort <*port*>] [-snmpRetries <*number*>] [-

snmpSecurityLevel (COMMUNITY_ONLY|COMMUNITY|NO_AUTH_NO_PRIV|AUTH_NO_PRIV|AUTH_PRIV|COMMUNITY_ONLY_V1)] [-snmpTimeout <*timeout*>] [-snmpV3Settings (<*name*>|<*uuid*>)] [-snmpWriteCommunity <*community_string*>]

nnmcommunication.ovpl updateSnmpAgentSettings -node (<*name*> | <*uuid*>) [-address <*ip*>] [-community <*string*>] [-enabled (true|false)] [-mode (AUTO|LOCKED)] [-port <*port*>] [-retries <*integer*>] [-snmpProxyAddress <*ip*>] [-snmpProxyPort <*port*>] [-timeout <*duration*>] [-version (V1|V2C|V3)] [-writeCommunity <*string*>]

nnmcommunication.ovpl updateNodeSettings -nodeSetting (<*name*>|<*uuid*>) [-addressDiscovery (true|false)] [-description <*description*>] [-icmpEnabled (true|false)] [-icmpRetries <*number*>] [-icmpTimeout <*timeout*>] [-managementAddress <*ip*>] [-name <*newName*>] [-shellCredential <*credential*>] [-shellUser <*username*>] [-snmpCommunity <*string*>] [-snmpEnabled (true|false)] [-snmpGetBulk (true|false)] [-snmpPort <*port*>] [-snmpPreferredVersion (V1|V2C|V3)] [-snmpProxyAddress <*ip*>] [-snmpProxyPort <*port*>] [-snmpRetries <*number*>] [-snmpTimeout <*timeout*>] [-snmpV3Setting (<*name*>|<*uuid*>)] [-snmpWriteCommunity <*community_string*>]

nnmcommunication.ovpl updateRegionSettings -region (<*name*>|<*uuid*>) [-addressDiscovery (true|false)] [-addressFilter <*filter*>] [-description <*description*>] [-hostnameFilter <*filter*>] [-icmpEnabled (true|false)] [-icmpRetries <*number*>] [-icmpTimeout <*timeout*>] [-name <*newName*>] [-ordering <*integer*>] [-shellCredential <*credential*>] [-shellUser <*username*>] [-snmpCommunities <*strings*>] [-snmpEnabled (true|false)] [-snmpGetBulk (true|false)] [-snmpPort <*port*>] [-snmpProxyAddress <*ip*>] [-snmpProxyPort <*port*>] [-snmpRetries <*number*>] [-snmpSecurityLevel (COMMUNITY_ONLY|COMMUNITY|NO_AUTH_NO_PRIV|AUTH_NO_PRIV|AUTH_PRIV|COMMUNITY_ONLY_V1)] [-snmpTimeout <*timeout*>] [-snmpV3Settings (<*name*>|<*uuid*>)] [-snmpWriteCommunity <*community_string*>]

nnmcommunication.ovpl updateSnmpV3Setting -snmpV3Setting (<*name*>|<*uuid*>) [-authPass <*string*>] [-authProtocol (HMAC_MD5_96|HMAC_SHA_1|HMAC128_SHA_224|HMAC192_SHA256|HMAC256_SHA_384|HMAC384_SHA_512)] [-contextName <*string*>] [-name <*newName*>] [-privPass <*string*>] [-privProtocol (DES_CBC|TripleDES|AES_128|AES_192|AES_256)] [-username <*name*>]

nnmcommunication.ovpl updateWebAgentSettings -node (<*name*>|<*uuid*>) [-mode <*AUTO*>|<*LOCKED*>] [-agentEnabled (true|false)] [-username <*string*>] [-password <*string*>] [-port <*port*>] [-scheme (HTTP|HTTPS)] [-timeout <*duration*>] [-cert <*cert_file*>]

DESCRIPTION

nnmcommunication.ovpl コマンドラインの一般的なフォーマットは次のとおりです。

nnmcommunication.ovpl <*command*> <*options*>

下記の「Commands」項には，使用可能なコマンドの選択肢が一覧表示されています。同様に「Options」項には，各コマンドで使用可能なオプションが一覧表示されています。多くのコマンドで類似のオプションを使用できます。各コマンドで使用できる正しいオプションについては，上記の「SYNOPSIS」項を参照してください。

Commands

`-h | -help`

コマンドの使用方法を表示します。

`addCertificate`

デバイスの信頼済み証明書を通信の設定のデフォルト，特定のノード，または領域の設定に追加します。

`addCommunity`

デフォルトまたは領域の設定にコミュニティ文字列を追加します。順序が設定されていない場合，指定したコミュニティ文字列は最後に試行されます。

`addCredential`

デバイスの資格証明を通信の設定のデフォルト，特定ノード，または領域の設定に追加します。

`addSnmpV3Setting`

デフォルト設定または領域の設定に指定された SNMPv3 の設定を割り当てます。

`createAddressRange`

領域<*region*>に新しいアドレス範囲のエントリを作成します。領域には複数のアドレス範囲を設定できます。

`createHostnameFilter`

新しいホスト名フィルターエントリを作成します。

`createNodeSettings`

新しい特定ノードの設定を作成します。

`createRegionSettings`

新しい領域の設定を作成します。

`createSnmpV3Settings`

新しい SNMPv3 設定を作成します。

`delete`

領域の設定，特定ノードの設定，または SNMPv3 の設定を削除します。

`deleteAddressRange`

一つ以上のアドレス範囲のエントリを削除します。

`deleteHostnameFilter`

一つ以上のホスト名フィルターのエントリを削除します。

`listAddressRanges`

アドレス範囲のエントリを一覧表示します。

`listCertificates`

通信の設定のデフォルト，特定ノード，または領域の設定に設定された，デバイスの信頼済み証明書のリストを表示します。

`listCommunities`

デフォルトまたは領域の設定のコミュニティ文字列を一覧表示します。

`listCredentials`

通信の設定のデフォルト，特定のノード，または領域の設定に設定された，デバイスの資格証明リストを表示します。

`listDefaults`

デフォルトの通信の設定を一覧表示します。

`listEffective`

ノードに設定されている有効な通信の設定を一覧表示します。

`listHostnameFilters`

ホスト名フィルターのエントリを一覧表示します。

`listNodeSettings`

特定ノードの設定を一覧表示します。

`listRegionSettings`

設定済みの領域を一覧表示します。領域名が指定されている場合は，その領域だけを一覧表示します。領域名が指定されていない場合は，すべての領域を一覧表示します。

`listSnmpAgentSettings`

ローカル NNMi サーバーで管理している SNMP エージェントのアクティブな設定を一覧表示します。

`listWebAgentSettings`

ローカルの NNMi 管理サーバーによって管理されるすべての Web Agent の設定を表示します。特定ノードで実行されている Web Agent のリストを表示するには，ノードのホスト名または UUID をコマンドで指定します。

`listSnmpV3Settings`

SNMPv3 設定を一覧表示します。固有名または UUID が指定されている場合は，その値に一致する SNMPv3 設定だけを一覧表示します。固有名または UUID が指定されていない場合は，すべての SNMPv3 設定を一覧表示します。

`removeCertificate`

通信の設定のデフォルト，特定ノード，または領域に設定されているデバイスの信頼済み証明書を削除します。

`removeCommunity`

デフォルトまたは領域の設定から，コミュニティ文字列を削除します。

`removeCredential`

通信の設定のデフォルト，特定ノード，または領域に設定されているデバイスの資格証明を削除します。

`removeSnmpV3Setting`

デフォルトまたは領域の設定から，指定した SNMPv3 設定を削除します。

`updateSnmpAgentSettings`

ローカルの NNMi サーバーが管理するノードのエージェントの設定を直接更新します。エージェントの設定を直接更新すると，コマンドに-mode AUTO 引数を指定しない限り，エージェントのモードが LOCKED になります。

`updateCredential`

通信の設定のデフォルト，特定ノード，または領域に設定されているデバイスの資格証明を更新します。

`updateDefaults`

デフォルトの通信の設定フィールドを更新します。

`updateNodeSettings`

特定ノードの設定のフィールドを更新します。

`updateRegionSettings`

領域設定のフィールドを更新します。

`updateSnmpV3Setting`

SNMPv3 設定を変更します。

`updateWebAgentSettings`

ローカルの NNMi 管理サーバーによって管理されるノードでホストされる Web Agent の設定を直接更新します。エージェントの設定を直接更新する場合，コマンドで-mode AUTO 引数を指定しないときは，エージェントモードが LOCKED になります。値を指定しないで"-cert"オプションを指定すると，Web Agent 設定から信頼済み証明書を削除できます。

Options

このセクションでは，上記のコマンドに共通のオプションを示します。

`-address <ip>`

ノードに使用する管理アドレスです。

`-addressDiscovery (true|false)`

true の場合，NNMi は古い管理アドレスが応答しなくなったときに，新しい管理アドレスを検出しようとします。

`-addressFilter <filter>`

領域に含まれるノードをアドレスの範囲で指定します。例えば，10.1.0.0/16, 192.168.1-20.* または fc00::/7 です。複数のエントリはセミコロンで区切ります。

`-authPass <string>`

SNMPv3 認証パスフレーズです。

`-authProtocol (HMAC_MD5_96|HMAC_SHA_1|HMAC128_SHA_224|HMAC192_SHA256|HMAC256_SHA_384|HMAC384_SHA_512)`

SNMPv3 認証プロトコルです。

-cert <*cert_file*>

信頼済み証明書ファイルに対する完全修飾パスです。サポートされる信頼済み証明書ファイルの拡張子は，.pem，.crt，.cer，および.der です。

-community <*string*>

コミュニティ文字列です。

-contextName <*string*>

SNMPv3 コンテキスト名です。

-default

add/update/remove オプションを指定した場合にデフォルト設定の値を変更します。list オプションの場合は，デフォルト設定を列挙します。

-description <*description*>

設定の説明です。

-enabled (true|false)

SNMP エージェントの有効，無効を設定します。

-fingerPrint <*fingerPrint*>

信頼済み証明書の公開鍵のフィンガープリントです。

-hostnameFilter <*filter*>

含まれるノードのホスト名のパターンです。例えば，*.usa.myco.com です。

-icmpEnabled (true|false)

true の場合は ICMP 通信を有効にし，false の場合は無効にします。

-icmpRetries <*number*>

ICMP のリトライ数です。

-icmpTimeout <*timeout*>

ミリ秒単位での ICMP のタイムアウトです。

-interfaceMatcher <*value*>

優先管理アドレスとして使用するインタフェースに一致する式です。

-managementAddress <*ip*>

ノードに使用する管理アドレスです。

-managementAddressSelection <*ALG1,ALG2,ALG3*>

管理アドレス選択アルゴリズムを実行する順番です。有効な値は，LOW_LOOPBACK, HIGH_LOOPBACK, SEED, または INTERFACE です。INTERFACE を指定する場合は，<*interfaceMatcher*>パラメータも指定する必要があります。

-mode (AUTO|LOCKED)

エージェントのモードです。AUTO は設定からの適用を意味します。LOCKED はユーザーによる明示的な設定を意味します。

-name <*name*>

名前で出力をフィルタリングします; ワイルドカードとして * と ? がサポートされています。

-name <*name*>

設定名です。

-name <*newName*>

更新後の名前です。

-node (<*name*>|<*uuid*>)

ノード名で出力をフィルタリングします; FQDN，短縮名，IP アドレス，および UUID がサポートされています。

-nodeSetting (<*name*>|<*uuid*>)

特定ノードの設定を指定します; 有効な入力はターゲットホスト名または UUID です。

-ordering <*integer*>

設定の優先順序です。設定は，1 から大きい数値へと，優先順序に従って試行されます。同じ優先順序を持つ 2 つの項目があるとエラーになります。

-password <*password*>

デバイス通信のパスワードです。

-port <*port*>

SNMP エージェントの UDP ポートです。

-preferIPVersion (IPv4|IPv6|IPAny)

（NNMi Advanced だけ）優先する IP バージョン設定です。

-privPass <*string*>

SNMPv3 プライバシパスフレーズです。

-privProtocol (DES_CBC|TripleDES|AES_128|AES_192|AES_256)

SNMPv3 プライバシプロトコルです。

-range <*ip_range*>

IP アドレス範囲です。例えば，192.168.1.0/24 や 192.168.1-10.*です。

-retries <*integer*>

SNMP のリトライ数です。

-region (<*name*>|<*uuid*>)

領域を指定します。有効な値は領域名または UUID です。

-serialNumber <*serialNumber*>

信頼済み証明書のシリアル番号です。

-shellCredential <*credential*>

デバイスの資格証明のパスワードです。

-shellUser <*username*>

デバイスの資格証明のユーザー名です。

-snmpCommunities <*strings*>

セミコロンで区切られた SNMP 読み取りコミュニティ文字列のリストです。Linux プラットフォーム上のシェルは，セミコロンを特殊文字として扱います。 したがって，セミコロンが特殊文字として解釈されないように，リストを引用符で囲む必要があります。

-snmpCommunity <*string*>

ノードに使用する SNMP コミュニティです。

-snmpEnabled (true|false)

SNMP 通信を有効または無効にします。

-snmpGetBulk (true|false)

true の場合，NNMi は SNMPv2c GetBulk コマンドを使用します。

-snmpPort <*port*>

SNMP 通信に使用するポートです。

-snmpPreferredVersion (V1|V2C|V3)

使用する SNMP 優先バージョンです。

-snmpProxyAddress <*ip*>

使用する SNMP プロキシの IP アドレスです。SNMP プロキシを利用できるようにするには，SNMP プロキシポートも指定する必要があります。

-snmpProxyPort <*ip*>

SNMP プロキシによる通信に使用するポートです。SNMP プロキシを利用できるようにするには，SNMP プロキシアドレスも指定する必要があります。

-snmpRetries <*number*>

SNMP のリトライ数です。

-snmpSecurityLevel (COMMUNITY_ONLY|COMMUNITY|NO_AUTH_NO_PRIV|AUTH_NO_PRIV|AUTH_PRIV|COMMUNITY_ONLY_V1)

SNMP 最小セキュリティレベルです。

-snmpTimeout <*timeout*>

ミリ秒単位での SNMP のタイムアウトです。

-snmpV3Setting (<*name*>|<*uuid*>)

名前または UUID により SNMPv3 設定を指定します。

-snmpV3Settings (<*name*>|<*uuid*>)

名前または UUID により SNMPv3 設定を指定します。

-snmpWriteCommunity <*community_string*>

SNMP 書き込みコミュニティ文字列です。

-timeout <*duration*>

SNMP Agent または Web Agent のタイムアウトです。SNMP Agent の場合，1 秒を 1000 または PT1S，Web Agent の場合は 1 秒を PT1S の形式で指定します。

-type <*SHELL|VMWARE|CISCOACI*>

資格情報のタイプ。有効な値は SHELL，VMWARE, CISCOACI です。

-username <*name*>

この設定のユーザー名です。

-uuid <*uuid*>

オブジェクトの一意の識別子です。

-version (V1|V2C|V3)

このエージェントが使用する SNMP バージョンです。有効な値は V1，V2C または V3 です。

-writeCommunity <*string*>

SNMP 書き込みコミュニティ文字列です。

-agentEnabled (true|false)

Web Agent の通信を有効または無効にします。

-scheme (HTTP|HTTPS)

Web Agent の通信に使用されるスキームです。

Additional Parameters

-fields <*fields*>

出力するフィールドを選択します。

-format <*style*>

テーブルデータの出力形式を設定します; 有効な値は TEXT, LIST, CSV または XML です。

-http.host <*host*>

サーバーホストです;デフォルトは localhost です。

-http.port <*port*>

サーバーポートです;デフォルトは 80 です。

-p <*password*>

ユーザーのパスワードです。

-quiet

通常の出力を抑制し，エラーだけを出力します。

-u <*username*>

このコマンドを実行するためのユーザー名です。

EXAMPLES

米国内（ホスト名によって判定）のすべてのノードについて地域の設定を作成し，順序を 10 に設定します。

```
nnmcommunication.ovpl createRegionSettings -name UsaNodes -ordering 10 -hostnameFilter
*.usa.myco.com.
```

一部のフィールドを設定するために上記で作成した領域設定を更新します。

```
nnmcommunication.ovpl updateRegionSettings -region UsaNodes -addressDiscovery true -
snmpTimeout 200
```

上記で作成した領域設定を一覧表示します。

```
nnmcommunication.ovpl listRegionSettings -name UsaNodes
```

SNMPv3 設定を作成します。

```
nnmcommunication.ovpl createSnmpV3Settings -name xyzSettings -username xyzUser -contextName
xyzContext -authProtocol HMAC_MD5_96 -authPass myPass -privProtocol AES_192 -privPass
myPrivPass
```

上記で作成した SNMPv3 設定を一覧表示します。

```
nnmcommunication.ovpl listSnmpV3Settings -name xyzSettings
```

上記で作成した SNMPv3 設定を使用して，ノード xyz.myco.com の特定ノードの設定を作成します。

```
nnmcommunication.ovpl createNodeSettings -name xyz.myco.com -snmpPreferredVersion V3 -
snmpV3Setting xyzSettings
```

xyz.myco.com の有効なノードの設定を一覧表示します。

```
nnmcommunication.ovpl listEffective -node xyz.myco.com
```

xyz.myco.com のノードの設定を更新し，ICMP を無効にします。

```
nnmcommunication.ovpl updateNodeSettings -nodeSetting xyz.myco.com -icmpEnabled false
```

管理アドレスを更新し，ノード mynode のエージェント設定をロックします。

```
nnmcommunication.ovpl updateSnmpAgentSettings -node mynode -address 192.168.1.1
```

mynode のエージェント設定をロックせずに管理アドレスを更新します。

```
nnmcommunication.ovpl updateSnmpAgentSettings -node mynode -address 192.168.1.1 -mode AUTO
```

mynode でホストされる Web Agent を更新し，ポートを 80 に設定し，HTTP を使用し，3 分でタイムアウトします。

```
nnmcommunication.ovpl updateWebAgentSettings -node mynode -port 80 -scheme HTTP -timeout PT3M
```

mynode でホストされる Web Agent を更新して，指定した信頼済み証明書を設定します。

```
nnmcommunication.ovpl updateWebAgentSettings -node mynode -cert /tmp/trustedCert-host1.pem
```

デフォルトの資格情報を新たに追加して，VMware ハイパーバイザーと通信します

```
nnmcommunication.ovpl addCredential -default -type VMWARE -username vmwareuser -password
vmwarepass
```

デフォルトの資格情報をすべてリスト表示します。

```
nnmcommunication.ovpl listCredentials -default
```

デフォルトの VMWARE 資格情報を削除します。

```
nnmcommunication.ovpl removeCredential -default -type VMWARE
```

デフォルトの VMWARE 資格情報を更新します。

```
nnmcommunication.ovpl updateCredential -default -type VMWARE -username username -password
password
```

通信の設定のデフォルトの設定で，信頼済み証明書を追加します。

```
nnmcommunication.ovpl addCertificate -default -cert /tmp/trustedCert-host1.pem
```

通信の設定で「アメリカ」領域に設定された，すべての信頼済み証明書を表示します。

```
nnmcommunication.ovpl listCertificates -region Americas
```

デフォルトに設定された信頼済み証明書を対応するシリアル番号とともに削除します。

```
nnmcommunication.ovpl removeCertificate -default -serialNumber ”1111111111”
```

AUTHOR

`nnmcommunication.ovpl` was developed by Micro Focus.

FILES

- Windows：`%NNM_BIN%¥nnmcommunication.ovpl`
- Linux：`$NNM_BIN/nnmcommunication.ovpl`

付録 G.13　nnmconfigexport.ovpl

構成情報をファイルにエクスポートします。このファイルから，ほかのシステムに構成情報をインポートすることができます。

SYNOPSIS

nnmconfigexport.ovpl -? | -c <*configuration*>[,configuration...] [-a <*author_key*>] [-u <*username*> -p <*password*>] [-x <*file_prefix*>] [-f <*output file or directory*>]

DESCRIPTION

nnmconfigexport.ovpl はカスタム構成情報を標準出力またはファイルにエクスポートする Perl スクリプトです。

Parameters

nnmconfigexport.ovpl コマンドは，次のオプションをサポートします。

-?

コマンドの使用方法を表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-c <*configuration*>[,configuration...]

指定した構成情報の XML スキーマをエクスポートします。複数のファイルを指定するときは，カンマ（,）で区切ります。なお，複数の構成情報ファイルがある場合，-f オプションには必ずディレクトリを指定します。
指定可能な構成情報ファイル：

account

ユーザーアカウント，ユーザーグループおよびユーザーアカウントのマッピングをエクスポートします。

author

作成者をエクスポートします。-a オプションでフィルターすることもできます。

customCorrelation

カスタム相関処理の設定をエクスポートします。-a オプションでフィルターすることもできます。

comm

通信の設定をエクスポートします。SNMPv3 の通信の設定はエクスポートされません。このデータに対する暗号化アルゴリズムは，NNMi 固有の内部キーに依存しています。 このデータはインポートできないため，エクスポート対象から除外しています。

custpoll

カスタムポーラーの設定をエクスポートします。

device

デバイスのプロファイルをエクスポートします。-a オプションでフィルターすることもできます。

disco

検出の設定（シードを除く）をエクスポートします。

discoseed

検出シードをエクスポートします。

icons

アイコンをエクスポートします。-a オプションでフィルターすることもできます。

ifgroup

インタフェースグループをエクスポートします。

iftype

インタフェース種別（ifTypes）をエクスポートします。

incident

インシデントの設定をエクスポートします。-a オプションでフィルターすることもできます。

menu

メニューをエクスポートします。-a オプションでフィルターすることもできます。

menuitem

アクションメニューに設定されたすべてのメニュー項目をエクスポートします。-a オプションが指定された場合，関連する親のメニューやサブメニューも出力に含まれます。

mibexpr

MIB 式をエクスポートします。-a オプションでフィルターすることもできます。

mibtypes

MIB OID タイプをエクスポートします。

monitoring

モニタリングの設定をエクスポートします。

nodegroup

ノードグループをエクスポートします。

ngmap

ノードグループマップをエクスポートします。ノードの座標を正しくインポートするには，両マシン間でノードのホスト名を一致させる必要があります。

oam

重複する IP アドレスマッピングをエクスポートします。

security

セキュリティグループおよびテナントをエクスポートします。

securitymappings

セキュリティグループのマッピングをエクスポートします。

status

ノードグループのステータス設定をエクスポートします。

trap

トラップログ記録設定をエクスポートします。-a オプションでフィルターすることもできます。

ui

ユーザーインタフェースの設定をエクスポートします。

all

有効な構成領域をすべてエクスポートします。このオプションに関しては，出力先はディレクトリでなければなりません。

-a <*author_key*>

指定した作成者キーを持つ作成者が作成した構成項目だけを，インクリメンタルインポート用の特殊な XML 形式でエクスポートします。nnmconfigimport.ovpl はこの XML 形式を自動的に検出するので，特別なオプションを指定する必要はありません。 このオプションは，author，customCorrelation，device，icons，incident，menu，menuitem，mibexpr，およびtrap の各構成情報ファイルに対してだけ有効です。有効な作成者キーは，作成者のインポートを行うことで調べることができます。「EXAMPLES」の項に示す例を参照してください。

-f <*output file or directory*>

指定したファイルまたはディレクトリに出力内容を保存します。

-x <*file_prefix*>

出力先としてディレクトリを指定した場合に，出力ファイルの命名に使用するファイル名プレフィックスを指定します。ファイルは <*prefix*>-<*area*>.xml の名前になります。

EXAMPLES

nnmconfigexport.ovpl -u myusername -p myadminpassword -c comm

通信の設定を標準出力にエクスポートします。

nnmconfigexport.ovpl -u myusername -p myadminpassword -c comm,disco -f /tmp -x my

通信の設定と検出の設定を/tmp/my-comm.xml ファイルおよび/tmp/my-disco.xml ファイルにエクスポートします。

nnmconfigexport.ovpl -u myusername -p myadminpassword -c author

作成者キーおよびラベルを持つ作成者すべてを標準出力にエクスポートします。

nnmconfigexport.ovpl -u myusername -p myadminpassword -c menuitem -a com.mycorp.nnm.author -f /tmp/mycorpmenuitems.xml

作成者キーがcom.mycorp.nnm.author の作成者によって作成されたメニュー項目の設定を/tmp/mycorpmenuitems.xml ファイルにエクスポートします。

AUTHOR

nnmconfigexport.ovpl was developed by Micro Focus.

SEE ALSO

nnmconfigimport.ovpl, nnm.properties.

付録 G.14　nnmconfigimport.ovpl

nnmconfigexport.ovpl の XML 出力を NNMi データベースにインポートします。

SYNOPSIS

nnmconfigimport.ovpl -? | [-u <*username*> -p <*password*>] -f <*input file or directory*> [-x <*file_prefix*>] [-memory <*number of megabytes*>] [-timeout <*time in seconds*>]

DESCRIPTION

nnmconfigimport.ovpl は，nnmconfigexport.ovpl の出力を NNMi データベースにインポートする Perl スクリプトです。

注意：アクションをインポートし，変更内容を有効にするには，NNMi コンソールを再起動（サインアウトしてからサインイン）する必要があります。

Parameters

nnmconfigimport.ovpl コマンドは，次のオプションをサポートします。

-?

コマンドの使用方法を表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-f <*input file or directory*>

指定した構成 XML ファイルをインポートします。ディレクトリが指定された場合は，ディレクトリ内のファイルすべてをインポートします。

-x <*file_prefix*>

-f オプションを使用してディレクトリを指定する場合に，インポート対象となるファイルをフィルタリングするためのファイルプレフィックスを指定します。依存関係でソートされ，指定したディレクトリ内にある，名前が<*prefix*>-*のファイルがインポートされます。

-memory <*memory in megabytes*>

nnmconfigimport.ovpl コマンドを動作させるために割り当て可能なメモリサイズを指定します。デフォルトは 1536MB ですが，サイズの大きいファイルに対しては，この値を 2048 に設定する必要があるかもしれません。このオプションは実際にインポートを行うツールに渡されるのではなく，ただ単に nnmconfigimport.ovpl を動作させるために使用するものであるため，使用方法メッセージ（Usage）には表示されません。

-timeout <*time in seconds*>

特定のファイルのインポートを完了するために利用できる時間を指定します。インシデントなど一部の構成情報の形式は，データ量に応じてより大きなタイムアウト値を必要とすることがあります。デフォルト値は 3600 秒(60 分)です。このオプションは実際にインポートを行うツールに渡されるのではなく，ただ単にnnmconfigimport.ovpl を動作させるために使用するものであるため，使用方法メッセージ（Usage）には表示されません。

NOTES

nnmconfigimport.ovpl コマンドは，大部分のエリアでは，既存の設定に追加を行います。 また，検出，通信，モニタリング，およびステータスのような一部のエリアでは，既存の設定を置き換えます。詳細は，NNMi ヘルプ［エクスポート/インポート時の動作と依存関係］を参照してください。

設定項目を追加するためには，作成者キーを指定して nnmconfigexport.ovpl を実行してください。

nnmconfigimport.ovpl は自動的にnnmconfigexport.ovpl が作成者キーを指定して実行されたかを検知して，設定を置き換える代わりに追加します。

インポートする前に，nnmconfigexport.ovpl の出力を編集しないでください。

EXAMPLES

nnmconfigimport.ovpl -u username -p password -f /tmp/nnmconfig.xml

カスタム構成情報を/tmp/nnmconfig.xml ファイルから NNMi データベースにインポートします。(NNMi ユーザー名とパスワードを指定する必要があります。この例では，ユーザー名はusername，パスワードはpassword です。)

AUTHOR

nnmconfigimport.ovpl was developed by Micro Focus.

SEE ALSO

nnmconfigexport.ovpl，nnm.properties.

付録 G.15　nnmconfigpoll.ovpl

ノードに対してポーリングを行って検出情報を取得します。

SYNOPSIS

nnmconfigpoll.ovpl [-v] [-t *timeout in secs*] [-u <*username*> -p <*password*>] [-tenant <*name*>] *node*

DESCRIPTION

nnmconfigpoll.ovpl コマンドは，要求を検出サービスに送信して，ノードにポーリングを行って検出情報を取得します。ノードは，検出されたトポロジに存在する必要があります。node パラメータは，トポロジでのノード名またはノードに関連する IP アドレスを指定できます。

nnmconfigpoll.ovpl コマンドを実行すると，ノードに対してレイヤー 2 接続解析が開始されます。検出サービスがデバイスにポーリングを行うと，NNMi コンソールにレイヤー 3 検出情報のステータスメッセージが表示されます。

なお，このコマンドによるポーリングで得られるのは検出情報です。ステータス情報のポーリングを行うには，nnmstatuspoll.ovpl コマンドを使用します。

Parameters

nnmconfigpoll.ovpl コマンドは，次のパラメータおよびオプションをサポートします。

-v

検出ポーリングに関する詳細な情報を表示します。

-t <*timeout in secs*>

クライアントは，秒単位で指定したタイムアウトの秒数だけ応答を待ちます。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-tenant <*name*>

ノードが属するテナントを指定します。重複した IP アドレスを使用しているなど，ノード名がネットワーク内で一意でない場合に使用する必要があります。デフォルトはありません。

EXAMPLES

nnmconfigpoll.ovpl コマンドを使用して，ノードのポーリングを行う方法の例を次に示します。

ノード名を使用してノードのポーリングを行う場合の例：

```
nnmconfigpoll.ovpl -u username -p password thisnode
```

完全修飾されたノード名を使用してノードのポーリングを行う場合の例：

```
nnmconfigpoll.ovpl -u username -p password thisnode.x.y.z
```

IP アドレスを使用してノードのポーリングを行う場合の例：

```
nnmconfigpoll.ovpl -u username -p password 10.97.247.129
```

IP アドレスとテナント名を使用してノードのポーリングを行う場合の例：

```
nnmconfigpoll.ovpl -u username -p password -tenant myDuplicateAddressesDomain 10.97.247.129
```

AUTHOR

nnmconfigpoll.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmconfigpoll.ovpl
- Linux：$NNM_BIN/nnmconfigpoll.ovpl

SEE ALSO

nnmstatuspoll.ovpl.

付録 G.16　nnmconnedit.ovpl

L2（レイヤー 2）接続トポロジを修正します。ユーザーは接続を追加したり削除したりできます。

SYNOPSIS

nnmconnedit.ovpl -f *corrections file* -t [add|delete] [-help] [-u <*username*> -p <*password*>]

DESCRIPTION

さまざまな要因により，NNMi L2 接続トポロジの検出には誤りが混入する可能性があります。nnmconnedit.ovpl コマンドは，ユーザーが接続を追加したり削除したりする方法を提供します。修正ファイルは，次の構造の XML 形式で管理者が作成します。

```
<connectionedits>
    <connection>
        <operation>add または delete</operation>
        <node>ショートネーム，ロングネーム，または IP アドレス</node>
        <interface>ifName, ifAlias, ifDescr または ifIndex</interface>
        <node>ショートネーム，ロングネーム，または IP アドレス</node>
        <interface>ifName, ifAlias, ifDescr または ifIndex</interface>
    </connection>
</connectionedits>
```

各要素の説明：

operation 接続を追加するのか削除するのかを指定します。

node ショートネーム，ロングネーム（DNS 名），または IP アドレスで識別されます。

interface ifIndex, ifName, ifDescr, または ifAlias の順で識別されます。 この値は一意である必要があります。ただし， ifIndex の使用は，デバイスによっては，サポートされているインタフェース再ナンバリング機能によって妨げられることがあります。 非 SNMP ノードでは， ifAlias または ifDescr の使用を推奨します。

connection ごとに，node および interface が少なくとも二つずつ必要です。node の数と interface の数は一致している必要があります。node および interface のペアはエンドポイントとして知られ，connection 一つに，エンドポイントを二つ以上指定できます。修正ファイルに複数の connection を指定できます。

接続を追加するときは，所属している可能性のある既存の接続から各エンドポイントが削除され，新しい接続へ追加されます。connection に三つ以上のエンドポイントが指定されている場合，接続はマップ上に共有メディア接続シンボルとして表示されます。connection に，NNMi データベースに既に存在する接続を指定している場合は，何も変更されません。

接続を削除する場合は，同じ組み合わせのエンドポイントを持つ接続が既に NNMi データベースに存在している場合を除いて，何も変更されません。その場合，指定したエンドポイントすべてが切断状態のまま

となります。ネットワーク機器が接続を報告している場合，その接続を一時的に削除するだけです。この場合，削除された接続は，NNMi が次回その接続先にあるノードを検出したときに再表示されます。

Parameters

nnmconnedit.ovpl コマンドは，次のパラメータおよびオプションをサポートします。

-f *corrections file*

接続追加および削除命令が指定された定型ファイルの名前を指定します。

-t [add | delete]

修正ファイルを作成するときに使用できるテンプレートファイルを生成します。*add* を指定すると追加操作テンプレートが作成され，*delete* を指定すると削除操作テンプレートが作成されます。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

help

このオプションを指定すると，コマンドの使用方法が表示されます。

EXAMPLES

NNMi の検出サービスが，異なるメーカーの二つのスイッチ間の L2 接続を見つけられなかったとします。この場合は，まず各デバイスで接続の必要があるノード名とインタフェース名を取得します。そして，nnmconnedit.ovpl コマンドに-t オプションを指定して add テンプレートファイルを作成します。次に，作成したテンプレートファイルを編集してノードおよびインタフェースの情報を指定します。編集が終わったら，mychg.xml ファイルとして保存します。

mychg.xml（add テンプレートファイル）の例を次に示します。

```
<connectionedits>
    <connection>
        <operation>add</operation>
        <node>nodeA.x.y.z</node>
        <interface>fa/09</interface>
        <node>nodeB.x.y.z</node>
        <interface>fa/05</interface>
    </connection>
</connectionedits>
```

最後に，mychg.xml ファイルを-f オプションに指定してnnmconnedit.ovpl コマンドを実行します。

```
nnmconnedit.ovpl -u username -p password -f mychg.xml
```

NNMi の検出サービスが，存在しないはずの L2 接続を作成する問題があるとします。 この場合は，まず不正な connection から，ノード名とインタフェース名を取得します。そして，nnmconnedit.ovpl コマンドに -t オプションを指定して delete テンプレートファイルを作成します。次に，作成したテンプレートファイルを編集してノードおよびインタフェースの情報を指定します。編集が終わったら，mychg.xml ファイルとして保存します。

mychg.xml（delete テンプレートファイル）の例を次に示します。

```
<connectionedits>
    <connection>
        <operation>delete</operation>
        <node>nodeA.x.y.z</node>
        <interface>fa/09</interface>
        <node>nodeB.x.y.z</node>
        <interface>fa/05</interface>
    </connection>
</connectionedits>
```

最後に， mychg.xml ファイルを -f オプションに指定して nnmconnedit.ovpl コマンドを実行します。

```
nnmconnedit.ovpl -u username -p password -f mychg.xml
```

AUTHOR

nnmconnedit.ovpl was developed by Micro Focus.

付録 G.17　nnmcustompollerconfig.ovpl

nnmcustompollerconfig.ovpl は，カスタムポーラー設定を表示，更新します。

SYNOPSIS

nnmcustompollerconfig.ovpl

nnmcustompollerconfig.ovpl createDeltaMap [-uuid <*object uuid*>] -variable <*name*>|<*uuid*> -stateMapping <*statemapping*> (-increaseInValue <*increase in value*> | -dropInValue <*drop in value*>| -increaseInValue <*increase in value*> -dropInValue <*drop in value*>)

nnmcustompollerconfig.ovpl deleteDeltaMap -deltaMap <*uuid*> | -list <*csv list of identifiers*>

nnmcustompollerconfig.ovpl listDeltaMap [-uuid <*object uuid*>] [-variable <*name*>|<*uuid*>] [-stateMapping <*state mapping*>] [-increaseInValue <*increase in value*>] [-dropInValue <*drop in value*>]

nnmcustompollerconfig.ovpl updateDeltaMap -deltaMap <*uuid*> { [-stateMapping <*state mapping*>] [-increaseInValue <*increase in value*>] [-dropInValue <*drop in value*>] }

nnmcustompollerconfig.ovpl listCollection [-name <*name*>]

nnmcustompollerconfig.ovpl listPolicy [-activeState <*active state*>] [-collection <*name*>|<*uuid*>] [-name <*name*>] [-nodeGroup <*name*>|<*uuid*>] [-uuid <*object uuid*>]

nnmcustompollerconfig.ovpl updatePolicy -policy <*name*>|<*uuid*> [-activeState <*active state*>]

DESCRIPTION

nnmcustompollerconfig.ovpl コマンドラインの一般的なフォーマットは次のとおりです。

nnmcustompollerconfig.ovpl <*command*> <*options*>

下記の Commands 項には，使用可能なコマンドの選択肢が一覧表示されています。 同様に Options 項には，各コマンドで使用可能なオプションが一覧表示されています。 多くのコマンドで類似のオプションを使用できます。 各コマンドで使用できる正しいオプションについては，上記の Synopsis 項を参照してください。

Commands

createDeltaMap

MIB 変数に対する差分マップを作成します。

deleteDeltaMap

MIB 変数から差分マップを削除します。

listDeltaMap

オプションのフィルタに基づいて差分ベースのしきい値を一覧表示します。

updateDeltaMap

差分マップのフィールドをアップデートします。

listCollection

オプションのフィルターに基づいて，収集の一覧を表示します。フィルターを指定しない場合，すべての収集が一覧表示されます。

listPolicy

オプションのフィルターに基づいて，ポリシーの一覧を表示します。フィルターを指定しない場合，すべてのポリシーが一覧表示されます。

updatePolicy

ポリシーのフィールドを更新します。

Options

-activeState <*active state*>

ポリシーのアクティブ状態を指定します。(active | inactive | suspended)

-collection <*name*>|<*uuid*>

収集の名前または uuid を指定します。

-name <*name*>

設定の名前を指定します。

-nodeGroup <*name*>|<*uuid*>

ポリシーが適用されているノードグループの名前または uuid を指定します。

-policy <*name*>|<*uuid*>

ポリシーの名前または uuid を指定します。

Additional Parameters

-fields <*fields*>

表形式のデータが存在しているときに出力フィールドを選択します。

-format <*style*>

表形式のデータが存在しているときに出力形式を変更します。指定できる値は，"TEXT", "LIST", "CSV", または "XML" です。

-http.host <*host*>

サーバーのホスト。デフォルトは localhost です。

-http.port <*port*>

サーバーのポート。デフォルトは 80 です。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。

-quiet

通常の出力を抑制し，エラーだけを表示します。

EXAMPLES

カスタムポーラーポリシーを一括で ACTIVE にする場合は，以下のようなサンプルバッチファイルまたはサンプルシェルスクリプトを参考にスクリプトを作成し実行してください。

Windows の場合のサンプルバッチファイル:

```
@echo off for /f ”usebackq” %%i in (`nnmcustompollerconfig.ovpl listPolicy -format CSV -fields uuid`) do ( if not %%i == UUID ( nnmcustompollerconfig.ovpl updatePolicy -policy %%i -activeState ACTIVE ) )
```

Linux の場合のサンプルシェルスクリプト:

```
#!/bin/sh /opt/OV/bin/nnmcustompollerconfig.ovpl listPolicy -format CSV -fields uuid | while
read line do if [ ! $line == UUID ]; then /opt/OV/bin/nnmcustompollerconfig.ovpl updatePolicy
-policy $line -activeState ACTIVE fi done
```

差分マップの作成

```
nnmcustompollerconfig.ovpl createDeltaMap -variable TestVariable -stateMapping major -
increaseInValue 100 -dropInValue 5
```

差分マップの更新

nnmcustompollerconfig.ovpl updateDeltaMap -deltaMap <*uuid*> -increaseInValue 90

差分ベースのしきい値の一覧表示

```
nnmcustompollerconfig.ovpl listDeltaMap -variable TestVariable
```

差分マップの削除

nnmcustompollerconfig.ovpl deleteDeltaMap -deltamap <*uuid*>

```
nnmcustompollerconfig.ovpl deleteDeltaMap -list uuid1,uuid2,uuid3
```

AUTHOR

nnmcustompollerconfig.ovpl was developed by Micro Focus.

FILES

$NNM_BIN/nnmcustompollerconfig.ovpl

付録 G.18　nnmdeleteattributes.ovpl

CSV ファイルまたはコマンドラインから，ノード，インタフェース，および物理コンポーネント（カード/シャーシ）に対してカスタム属性を削除します。

SYNOPSIS

nnmdeleteattributes.ovpl [-h | -help] -t <*type*> (-f <*path & filename of csv file*>) |(-s <*"csv formatted line"*>) [-u <*username*> -p <*password*>]

DESCRIPTION

nnmdeleteattributes.ovpl は，CSV ファイルに指定したカスタム属性を削除します。以前作成した不要なカスタム属性を削除するときに有用です。ノード，インタフェースまたは物理コンポーネントからカス

タム属性を削除します。ノードからカスタム属性を削除すると，カスタム属性を参照して形成されたノードグループからノードが削除されます。

Parameters

nnmdeleteattributes.ovpl コマンドは，次のオプションをサポートします。

-h | -help

コマンドの使用方法を表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-t <*type*>

削除対象のオブジェクトタイプを指定します。"node"，"interface"または "physcomp"です。

-f <*path & filename of csv file*>

削除するカスタム属性を含む CSV ファイル名（パス名。例: /tmp/csvfile.csv）を指定します。

-s <*"csv formatted line"*>

CSV 形式の 1 行を指定します。小さな変更に対してファイル作成を省略できます。

Syntax of Comma Separated File for Nodes

ノードからカスタム属性を削除するときに指定する CSV ファイルの構文は次のとおりです。

空の行は無視されます。

"#"で始まる行は無視されます。

- Column 1(A)：Node DNS|IP Address
  ノードの DNS 名か，IP アドレスを指定します。このフィールドは必須です。
- Column 2(B)：Attribute Name
  カスタム属性の名前。

削除するカスタム属性は，同じ行に指定できます。または，別の行で同じノードを指定します。

例）

192.168.1.1,Project,Service Type

192.168.1.1,Asset Tracking

192.168.2.2,Project,Service Type,Asset Tracking

Syntax of Comma Separated File for Interfaces

インタフェースからカスタム属性を削除するときに指定する CSV ファイルの構文は次のとおりです。

空の行は無視されます。

"#"で始まる行は無視されます。

- Column 1(A)：Node DNS|IP Address
  ノードの DNS 名か，IP アドレスを指定します。このフィールドは必須です。
- Column 2(B)：Interface Id
  前のフィールドで指定したノードのインタフェースの識別子を指定します。インタフェースのインデックス，エイリアス，インタフェース名，または説明が指定でき，この順序で検索されます。すべての一致するインタフェースに対して，属性が削除されます。このフィールドは必須です。
- Column 3(C)：Attribute Name
  カスタム属性の名前。

削除するカスタム属性は，同じ行に指定できます。または，別の行で同じノードとインタフェース識別子を指定します。

例）

192.168.1.1,1001,Project,Service Type

192.168.1.1,1001,Asset Tracking

192.168.2.2,A1,Project,Service Type,Asset Tracking

Syntax of Comma Separated File For PhysComp Attributes

物理コンポーネント（カード/シャーシ）からカスタム属性を削除するときに指定する CSV ファイルの構文は次のとおりです。

空の行は無視されます。

"#"で始まる行は無視されます。

- Column 1(A)：Node DNS | IP Address
  ノードの DNS 名か，IP アドレスを指定します。このフィールドは必須です。
- Column 2(B)：PhysComp Id
  前のフィールドで指定したノードの物理コンポーネントの識別子を指定します。物理コンポーネントの物理インデックス，名前，または説明が指定でき，この順序で検索されます。 すべての一致する物理コンポーネントに対して，属性が削除されます。このフィールドは必須です。

- Column 3(C)：PhysComp Type
  物理コンポーネントのタイプを識別する名前。"card"と"chassis"が有効なタイプです。
- Column 4(D)：Attribute Name
  カスタム属性の名前。

削除するカスタム属性は，同じ行に指定できます。または，別の行で同じノードと物理コンポーネント識別子，物理コンポーネントタイプを指定します。

例）

192.168.1.1,7,chassis,Location,Service Type

192.168.1.1,7,chassis,Asset Tracking

192.168.2.2,/AmdFE,card,Location,Service Type,Asset Tracking

EXAMPLES

CSV ファイルの内容例は次のようになります（ノードの場合）：

```
192.168.2.2,Project,Service Type,Asset Tracking
```

CSV ファイルからノードのカスタム属性を削除します：

```
nnmdeleteattributes.ovpl -t node -f /tmp/test.csv
```

コマンドラインからノードのカスタム属性を削除します：

```
nnmdeleteattributes.ovpl -t node -s ”192.168.1.1,Project”
```

CSV ファイルの内容例は次のようになります（インタフェースの場合）：

```
192.168.2.2,1001,Project,Service Type,Asset Tracking
```

CSV ファイルからインタフェースのカスタム属性を削除します：

```
nnmdeleteattributes.ovpl -t interface -f /tmp/test.csv
```

コマンドラインからインタフェースのカスタム属性を削除します：

```
nnmdeleteattributes.ovpl -t interface -s ”192.168.1.1,7,Project”
```

CSV ファイル（/tmp/test.csv）の内容例は次のようになります（物理コンポーネントの場合）：

```
192.168.2.2,7,chassis,Location,Service Type,Asset Tracking
```

CSV ファイルから物理コンポーネントのカスタム属性を削除します：

```
nnmdeleteattributes.ovpl -t physcomp -f /tmp/test.csv
```

コマンドラインから物理コンポーネントのカスタム属性を一つ削除します：

```
nnmdeleteattributes.ovpl -t physcomp -s "192.168.1.1,/AmdFE,card,Project"
```

Error Codes

問題を特定するのに役立ついくつかのエラーコードがあります：

- INFO：情報メッセージ。
- ATTR_ERROR：指定された属性で問題が見つかりました。
- DEL_FAIL_ERROR：指定された属性の削除に失敗しました。
- OBJECT_ERROR：指定されたオブジェクトが見つかりませんでした。
- BAD_LINE_ERROR：指定された行のフォーマットが正しくありません。
- IO_ERROR：CSV ファイルが見つからないか，読めませんでした。
- BAD_NAME_WARNING：指定された属性名が長すぎます(最大 50 文字)。
- BAD_VALUE_WARNING：指定された値が長すぎます(最大 2000 文字)。

AUTHOR

`nnmdeleteattributes.ovpl` was developed by Micro Focus.

FILES

- Windows：`%NNM_BIN%¥nnmdeleteattributes.ovpl`
- Linux：`$NNM_BIN/nnmdeleteattributes.ovpl`

SEE ALSO

nnmloadattributes.ovpl, nnmloadnodegroups.ovpl, nnm.properties.

付録 G.19　nnmdeleteurlaction.ovpl

指定した作成者キーのメニューやメニュー項目を削除します。

SYNOPSIS

`nnmdeleteurlaction.ovpl [-? | -h | -help] <authorKey> [-u username] [-p password]`

DESCRIPTION

`nnmdeleteurlaction.ovpl` は，指定した作成者キーに関するメニューやメニュー項目を削除する Perl スクリプトです。

注：メニューやメニュー項目を削除するためにnnmdeleteurlaction.ovplコマンドを使用した場合，変更を反映するために，NNMiコンソールを再起動（サインアウトしてからサインイン）する必要があります。

Parameters

nnmdeleteurlaction.ovplコマンドは，次のオプションをサポートします。

-? | -h | -help

コマンドの使用方法を表示します。

-u <*username*>

コマンドの実行に必要なNNMiの管理者名を指定します。nnm.propertiesファイルが存在していない限り，これが必要になります。詳細はnnm.propertiesリファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要なNNMiの管理者パスワードを指定します。nnm.propertiesファイルが存在していない限り，これが必要になります。詳細はnnm.propertiesリファレンスページを参照してください。

EXAMPLES

nnmdeleteurlaction.ovpl -u *username* -p *password* com.mycompany

作成者キーcom.mycompanyのメニューとメニュー項目を削除します。

AUTHOR

nnmdeleteurlaction.ovpl was developed by Micro Focus.

SEE ALSO

nnmconfigimport.ovpl, nnmconfigexport.ovpl.

付録 G.20　nnmdiscocfg.ovpl

SYNOPSIS

nnmdiscocfg.ovpl -autodisco rule=*rulename* rangetype=*ignore*|*include* [-f *ipAddressRangeFile* | -n *ipAddressRanges*] [-u <*username*> -p <*password*>]

nnmdiscocfg.ovpl -excludeipaddrs [-f *ipAddressRangeFile* | -n *ipAddressRanges*] [-u <*username*> -p <*password*>]

DESCRIPTION

nnmdiscocfg.ovpl コマンドを使用すると，IP アドレスの範囲を既存の自動検出ルールに追加できます。IP アドレスの自動検出範囲は，検出機能がネットワーク上でどのようにデバイスを見つけるかを制御します。

また，IP アドレス範囲除外フィルターを追加することで，NNMi トポロジ内に望ましくない IP アドレスが作成されるのを防ぐことができます。フィルターに一致したアドレスは，ノードにもインタフェースにも関連付けされず，IP アドレスインベントリに表示されることもありません。IP アドレス範囲フィルターは，ネットワーク上のデバイスを自動検出が検索および識別する方法を制御しません。

Parameters

nnmdiscocfg.ovpl コマンドは，次のパラメータおよびオプションをサポートします。

-autodisco rule=*ruleName* rangetype=*ignore*|*include*

ruleName で指定した既存の自動検出ルールに IP アドレス範囲を追加します。この範囲は，「rangetype=include」または「rangetype=ignore」を指定することで，ルールに追加できます。

-excludeipaddrs

検出の「除外対象 IP アドレス」設定に IP アドレス範囲を追加します。

-f *ipAddressRangeFile*

IP アドレス範囲が含まれる読み込み元テキストファイルを指定します。

-n *ipAddressRanges*

コマンドラインから直接読み込む IP アドレス範囲を指定します。複数の IP アドレス範囲を指定する場合はスペースで区切ります。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

IP アドレス範囲のリストを既存の自動検出ルールに追加する場合：

```
nnmdiscocfg.ovpl -u username -p password -autodisco rule=bld1floor2 rangetype=include -n
10.2.112.21-34 10.2.112.36 10.1.*.1-98
```

IP アドレス除外設定に IP アドレス範囲を追加する場合：

```
nnmdiscocfg.ovpl -u username -p password -excludeipaddrs -n 198.2.*.117
```

tmp ディレクトリ内のローカルファイルシステムにあるファイル lab3devices.txt から，IP アドレス範囲を読み込む場合：

```
nnmdiscocfg.ovpl -u username -p password -autodisco rule=bld1floor2 rangetype=include -
f /tmp/lab3devices.txt
```

tmp ディレクトリ内のローカルファイルシステムにあるファイル ignoreAddresses.txt から，IP アドレス範囲フィルターを読み込む場合：

```
nnmdiscocfg.ovpl -u username -p password -excludeipaddrs -f /tmp/ignoreAddresses.txt
```

AUTHOR

`nnmdiscocfg.ovpl` was developed by Micro Focus.

SEE ALSO

nnmnoderediscover.ovpl, nnm.properties.

付録 G.21　nnmengineidfile.ovpl

SNMP V3 Engine ID ファイルの管理が可能になります。

SYNOPSIS

```
nnmengineidfile.ovpl reload
```

```
nnmengineidfile.ovpl validate
```

DESCRIPTION

このコマンドでは，SNMP V3 Engine ID ファイルのリロードや検証を行うことができます。コマンドを使用する前に，ファイルのパスを`com.hp.nnm.snmp.engineid.file` プロパティに指定する必要があります。SNMP V3 Engine ID ファイルの詳細については，NNMi リリースノートの「21.8.1　SNMPv1 または SNMPv2c を使用して管理されているノードまたは 監視対象外のノードの SNMPv3 トラップを認証するための NNMi の設定」の章を参照してください。

`nnmengineidfile.ovpl` コマンドラインの一般的な形式は以下のとおりです。

`nnmengineidfile.ovpl` *<command>*

「Commands」セクションで，*command* 部分に設定可能な 選択肢をそれぞれ説明します。

Commands

`reload`

SNMP V3 Engine ID ファイルの内容をリロードします。

validate

SNMP V3 Engine ID ファイルの内容を検証します。現在の SNMPv3 キャッシュには影響しません。

Additional Parameters

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。

AUTHOR

nnmengineidfile.ovpl was developed by Micro Focus.

FILES

$NNM_BIN/nnmengineidfile.ovpl

付録 G.22　nnmhealth.ovpl

NNMi の自己監視情報を表示します。

SYNOPSIS

nnmhealth.ovpl [-u <*username*> -p <*password*>] (-print [quiet|brief|detailed|conclusions|verbose|agents|history] [-refresh] | -activate <conclusions> | -suppress <*conclusions*>) | -help

DESCRIPTION

nnmhealth.ovpl は，NNMi の自己監視情報を出力します。戻り値のみで出力がない(quiet)から，すべての自己監視情報をレポートする(verbose)まで，いくつかのレベルをサポートします。

NNMi の現在の自己監視情報を表示することに加えて，個々の状態の自己監視結果を抑制したり，有効化したりするために使用できます。管理者は，問題を監視し，問題が解決されるまでそれ以上の警告が表示されることを望まないのであれば，自己監視結果を抑制できます。

管理者がnnmhealth.ovpl コマンドを使用して抑制リストを編集した場合，次の自己監視スキャン時に有効になります。管理者が有効化するか，NNMi を再起動するまで，抑制は継続します。

NNMi の再起動後も抑制を継続したい場合は，nms-topology.properties の com.hp.ov.nms.health.SUPPRESSED_CONCLUSIONS プロパティが使用できます。このプロパティの値の書式は，nnmhealth.ovpl コマンドの-suppress オプションと同じ書式です。

Parameters

nnmhealth.ovpl コマンドは，次のオプションをサポートします。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-print <*level*>

NNMi の自己監視の情報を出力します。レベルは次のいずれかです。

brief|conclusions|detailed|agents|quiet|verbose|history

レベルが指定されていない場合，デフォルトはbrief になります。

-print brief：システムの全体ステータスを出力します。値は次のいずれかです。

正常域

注意域

警戒域

重要警戒域

危険域

-print conclusions：有効な結果と抑制の結果を出力します。

-print detailed：システムの詳細な自己監視情報を出力します。

-print agents：登録されているエージェントを出力します。

-print quiet：システムのステータスを表す整数値を返します。値は次のいずれかです。

0 - 正常域

1 - 注意域

2 - 警戒域

3 - 重要警戒域

4 - 危険域

-print verbose [-filter AgentList]：登録されているすべてのエージェントの詳細な自己監視情報を出力します。必要に応じて，-filter オプションを使用して，カンマ区切りのエージェント名を指定することができます。詳細な出力はサポートにだけ使用します。

-print history：登録されているエージェントの履歴情報を出力します。

-refresh

-print のオプション引数です。このオプションを使用すると，監視システムがレポートを返す前に情報を更新します。

-suppress *<conclusions>*

NNMi が次に再起動されるか，結果が再度有効化されるまで，指定した結果を抑制します。結果は，カンマ区切りのリストで指定できます。抑制できる有効な結果は-print conclusions で表示されます。

抑制している結果は，それまで有効であったということではないことに注意してください。

-activate *<conclusions>*

抑制リストから指定した結果を削除します。結果は，カンマ区切りのリストで指定できます。 再度有効化できる結果は-print conclusions で取得できます。

-help

コマンドの使用方法を表示します。

EXAMPLES

nnmhealth.ovpl -u username -p password -print brief

NNMi の全体ステータスを出力します。

nnmhealth.ovpl -u username -p password -print brief -refresh

NNMi の現在のステータスを更新して出力します。

nnmhealth.ovpl -u username -p password -print detailed

現在の自己監視の警告一覧を出力します。

nnmhealth.ovpl -u username -p password -print agents

自己監視の関連情報を報告する登録済みエージェントの現在の一覧を出力します。

nnmhealth.ovpl -u username -p password -print history

登録済み自己監視エージェントの履歴情報を出力します。

nnmhealth.ovpl -u username -p password -suppress "SystemLowSwap,SystemLowSwapPercent"

スワップ領域の絶対値とパーセンテージに関する自己監視のチェックをスキップするよう NNMi を設定します。

nnmhealth.ovpl -u username -p password -activate "SystemLowSwap"

SystemLowSwap 結果に対して，再度チェックするよう NNMi を設定します。

AUTHOR

nnmhealth.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_PROPS%¥nms-topology.properties
- UNIX：$NNM_PROPS/nms-topology.properties

付録 G.23　nnmicons.ovpl

NNMi の UI 設定

SYNOPSIS

nnmicons.ovpl -help

nnmicons.ovpl -list | -create (<*iconSpec1,iconSpec2,...*> | -file <*file*>) | -update (<*iconSpec1,iconSpec2,...*> | -file <*file*>) | -delete (<*iconName1,iconName2,...*> | -file <*file*>) [-u <*username*> -p <*password*>]

DESCRIPTION

nnmicons.ovpl は，NNMi のデータベースに保存されているアイコンへのアクセスを提供します。アイコンは，一覧表示，作成，更新，および削除することができます。

Parameters

nnmicons.ovpl コマンドは，次のオプションをサポートします。

-list

NNMi データベースに保存されているアイコンの一覧を表示します。

-create (<*iconSpec1,iconSpec2,...*> | -file <*file*>)

アイコンの定義，または入力ファイルのいずれかを使用して，アイコンを作成します。

<*iconSpec1,iconSpec2,...*>

アイコンの定義のリストはカンマで区切ります。iconSpecN は，次の形式で指定します。
iconName:authorKey[:<*iconImageSpec1*>[:<*iconImageSpec2*>]]

iconImageSpecN は，size:path の形式で指定します。size はピクセル単位の正方形の画像の大きさで，16 または 32 である必要があります。path は画像ファイルへのファイルパスを指定します。画像ファイルは，GIF，JPEG，または PNG 形式で，対応するファイルの拡張子は，.gif，.jpeg，.jpg，または.png のいずれかである必要があります。

-file

1 行につき一つのiconSpec のリストが含まれるファイルへのパスです。空白行やコメントをファイルに含めることができます。コメントは，行の先頭の'#'記号で表します。

-update (<*iconSpec1,iconSpec2,...*> | -file <*file*>)

アイコンの定義，または入力ファイルのいずれかを使用して，アイコンを更新します。アイコンが存在しない場合は，作成されます。

<*iconSpec1,iconSpec2,...*>

アイコンの定義のリストはカンマで区切ります。iconSpecN は，次の形式で指定します。
iconName:authorKey:<*iconImageSpec1*>:<*iconImageSpec2*>:...

iconImageSpec は，size:path の形式で指定します。size はピクセル単位の正方形の画像の大きさです。path は画像ファイルへのファイルパスです。画像ファイルは，GIF，JPEG，または PNG 形式で，対応するファイルの拡張子は，.gif，.jpeg，.jpg，または.png のいずれかである必要があります。

-file

1 行につき一つのiconSpec のリストが含まれるファイルへのパスです。空白行やコメントをファイルに含めることができます。コメントは，行の先頭の'#'記号で表します。

-delete (<*iconName1,iconName2,...*> | -file <*file*>)

アイコン名，または入力ファイルのいずれかを使用して，アイコンを削除します。アイコンが存在しない場合は，これらは無視されます。

<*iconName1,iconName2,...*>

iconNameN は，アイコンに関連付けられているアイコン名です。

-file

1 行につき一つのアイコン名のリストが含まれるファイルへのパスです。空白行やコメントをファイルに含めることができます。コメントは，行の先頭の'#'記号で表します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-help

コマンドの使用方法を表示します。

EXAMPLES

NNMi データベース内のアイコンの一覧を表示します。

```
nnmicons.ovpl -list
```

アイコンの定義を使用して，アイコンを作成します。

```
nnmicons.ovpl -create
iconName1:com.customer.author:16:image16.gif:32:image32.gif,iconName2:com.customer.author:16:a
notherimage16.gif
```

定義ファイルを使用して，アイコンを更新します。

```
nnmicons.ovpl -update -file /tmp/iconSpecificationFile.txt
```

アイコン名を使用して，アイコンを削除します。

```
nnmicons.ovpl -delete iconName1,iconName2
```

AUTHOR

nnmicons.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmicons.ovpl
- Linux：$NNM_BIN/nnmicons.ovpl

付録 G.24　nnmincidentcfg.ovpl

SNMP MIB から，インシデント構成を作成します。

SYNOPSIS

nnmincidentcfg.ovpl [[-loadTraps *mib_module_name* [-authorLabel *author_label* -authorKey *author_key*] [-skipExisting]] | -deleteAuthor *author_key* | -deleteCategory *category_key* | -deleteFamily *family_key* | -disableAllTraps <*true*|*false*> | -unloadTraps *mib_module_name* [-u *username*] [-p *password*]]

DESCRIPTION

nnmincidentcfg.ovpl コマンドは，MIB ファイルに TRAP-TYPE または，NOTIFICATION-TYPE マクロで定義された SNMP トラップのインシデント構成を作成するときに使用します。MIB 式の定義や，数値のオブジェクト ID をテキストで表示するために MIB をロードするには，nnmloadmib.ovpl コマンドを使用します。

作成直後のインシデントは次のデフォルト値を持ちます（値は NNMi コンソールで更新できます）。

1. "名前" は，MIB ファイル内に指定されたトラップ/通知の名前となります。
2. "SNMP のオブジェクト ID"は，MIB ファイル内に指定されたトラップ/通知の OID となります。
3. "有効にする"は，"true"となります。
4. "カテゴリ"は，"ステータス"となります。
5. "ファミリー"は，"ノード"となります。
6. "重大度"は，"正常域"となります。
7. "メッセージの形式"は，インシデント構成の名前となります。
8. "説明"は，MIB ファイル内に指定されたトラップ/通知の説明文となります。

作成されたインシデントに対する操作は，［SNMP トラップの設定］フォームで行えますので，必要に応じてカスタマイズできます。

nnmincidentcfg.ovpl は，#SUMMARY と呼ばれる特殊な注釈をサポートします。#SUMMARY 注釈の値は，インシデント設定項目中のメッセージ形式として適用されます。この注釈は，MIB ファイル内のトラップ説明の直後に，MIB コメントとして適用されます。次に例を示します。

```
    MyTrap TRAP-TYPE
    ENTERPRISE   hp
    VARIABLES {
            serverName, trapTime, volumeName, volumeNum
    }
    DESCRIPTION ”The disk volume is out of space. Please consult your sysop, and/or the prop
er manual.”
    --#SUMMARY  ”Volume $1 on system $2 is out of space.”
```

Parameters

nnmincidentcfg.ovpl コマンドは，次のオプションをサポートします。

-loadTraps 〈*mib_module_name*〉

トラップ定義を持つ MIB モジュール名を指定します。nnmincidentcfg.ovpl は，MIB ファイル内のトラップ/通知定義（TRAP-TYPE またはNOTIFICATION-TYPE マクロ）を解析し，それらに対応するインシデント定義を作成します。

-authorLabel 〈*author_label*〉

対象インシデント構成の作成者のラベルを指定します。 これは任意指定のパラメータです。作成者ラベルを指定した場合は，作成者キーも指定する必要があります。

-authorKey 〈*author_key*〉

対象インシデント構成の作成者のキーを指定します。これは任意指定のパラメータです。作成者キーを指定した場合は，作成者ラベルも指定する必要があります。com.example.nnm.author のように，会社のドメインによる java のパッケージ表記の使用を推奨します。

-skipExisting

このオプションが存在する場合，既存のインシデント構成は上書きされません。

-deleteAuthor 〈*author_key*〉

インシデント構成を使用しなくなった作成者を削除した方がよい場合があります。このオプションに作成者キーの値を指定すると作成者を削除できます。ただし，作成者オブジェクトを参照している設定がない場合に限ります。

-deleteCategory 〈*category_key*〉

インシデント構成によって使用されなくなったカテゴリを削除した方がよい場合があります。このオプションにカテゴリキーの値を指定するとカテゴリを削除できます。ただし，カテゴリオブジェクトを参照している設定がない場合に限ります。

-deleteFamily <*family_key*>

インシデント構成によって使用されなくなったファミリーを削除した方がよい場合があります。このオプションにファミリーキーの値を指定するとファミリーを削除できます。ただし，ファミリーオブジェクトを参照している設定がない場合に限ります。

-disableAllTraps <*true/false*>

この引数に true を指定した場合，インシデントの設定にすべてのトラップが無効として読み込まれます。デフォルトは false で，インシデント設定は有効です。

-unloadTraps <*mib_module_name*>

トラップ定義がある MIB モジュール名を指定します。その MIB モジュール内で見つかったトラップ定義や通知定義（TRAP-TYPE マクロまたはNOTIFICATION-TYPE マクロ）をnnmincidentcfg.ovpl が解析し，エントリの各 OID のインシデント設定を削除します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

```
nnmincidentcfg.ovpl -loadTraps "CISCO-VTP-MIB" -authorLabel "Cisco" -authorKey
com.example.cisco.nnm.author
```

```
nnmincidentcfg.ovpl -loadTraps "mpls"
```

AUTHOR

nnmincidentcfg.ovpl was developed by Micro Focus.

SEE ALSO

RFC 2578 Structure of Management Information Version 2 (SMIv2)

RFCs 1155, 1212, 1215: SNMP Version 1 Structure of Management Information

RFCs 1902, 1903, 1904: SNMP Version 2 Structure of Management Information

nnnloadmib.ovpl, nnm.properties.

付録 G.25　nnmincidentcfgdump.ovpl

インシデント構成を NNMi データベースからタグフォーマットファイルに出力します。

SYNOPSIS

nnmincidentcfgdump.ovpl { -dump <*filename*> [-uuid] [-authorKey <*author(s)*> | -name <*name(s)*> | -oid <*oid pattern(s)*> | -mib <*mib name(s)*>] [-type <*type(s)*>] [-timeout <*timeout*>] [-memory <*memory*>] [-u <*user name*> -p <*password*>] } { -listAuthors [-u <*user name*> -p <*password*>] }

DESCRIPTION

nnmincidentcfgdump.ovpl は，インシデント構成を，NNMi データベースからタグフォーマットファイルに出力します。このタグフォーマットファイルは，編集後nnmincidentcfgload.ovpl を用いることで NNMi データベースにロードすることができます。

nnmincidentcfgdump.ovpl がサポートするインシデント構成の種別を次に示します。

```
*MgmtEventConfig
*PairwiseConfig
*SnmpTrapConfig
*SyslogMessageConfig
```

なお，出力するインシデント構成は，事前に NNMi コンソールで作成するか，nnmincidentcfg.ovpl コマンドを用いて NNMi データベースにロードされている必要があります。

Parameters

nnmincidentcfgdump.ovpl コマンドは，次のオプションをサポートします。

-dump <*filename*>

インシデント構成をコピーするファイルを指定します。 指定したファイルが既に存在する場合，nnmincidentcfgdump.ovpl は警告を表示して終了します。

-uuid

UUID をほかのインシデント構成と共に出力したい場合に指定します。このコマンドを実行する際には，このオプションを指定して実行することを推奨します。

-authorKey <*author(s)*>

特定の作成者キーを持つインシデント構成だけを出力したい場合に指定します。このオプションの指定が無い場合は，すべての作成者キーを持つインシデントが出力されます。

-name や-oid，または-mib オプションと同時に指定することはできません。

-name <*name(s)*>

特定の名前を持つインシデント構成だけを出力したい場合に指定します。1 つ以上のインシデントの名前を指定することができます。このオプションの指定が無い場合はすべてのインシデント構成が出力されます。

-authorKey や-oid，または-mib オプションと同時に指定することはできません。

-oid <*oid pattern(s)*>

特定の OID を持つインシデント構成だけを出力したい場合に指定します。1 つ以上の OID を指定することができます。指定する OID は，次の書式に従う必要があります。

```
*ワイルドカード ”*” を1つだけ使用することができます。
*OIDは ”.” から始まる必要があります。
*OIDに使える文字は，区切り文字の ”.” と，数字とワイルドカードだけです。
```

-name や-authorKey，または-mib や-type オプションと同時に指定することはできません。

-mib <*mib name(s)*>

特定の MIB モジュールを含むインシデント構成だけを出力したい場合に指定します。 指定する MIB モジュールは，次を満たしている必要があります。

```
*すでにMIBモジュールがNNMiデータベースにロードされている。
*すでに MIB モジュールのトラップ定義がNNMiデータベースにロードされている。
```

-name や-authorKey，または-oid や-type オプションと同時に指定することはできません。

-type <*type(s)*>

特定のインシデント構成の種別だけ出力したい場合に指定します。1 つ以上のインシデント構成の種別を指定することができます。このオプションの指定が無い場合はすべてのインシデントの構成の種別が出力されます。

指定可能なインシデント構成の種別を次に示します。

注意：次の設定値は，大文字小文字を区別しません。

```
*MgmtEventConfig
*PairwiseConfig
*SnmpTrapConfig
*SyslogMessageConfig
```

-oid や-mib オプションと同時に指定することはできません。

-timeout <*timeout*>

nnmincidentcfgdump.ovpl コマンドのためのトランザクションタイムアウト時間を変更したい場合に秒で指定します。

-memory <*memory*>

最大ヒープサイズを MB で指定します。デフォルトは1536MB で，指定できる最小値は512MB です。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-listAuthors

すべての作成者キーと，そのラベルを出力します。

EXAMPLES

すべてのインシデント構成を出力します。

```
nnmincidentcfgdump.ovpl -dump full-dump.tag
```

すべての管理イベントと，SNMP トラップの構成を出力します。

```
nnmincidentcfgdump.ovpl -dump type-dump.tag -type MgmtEventConfig SnmpTrapConfig
```

作成者キーが Network Node Manager とカスタマのインシデント構成を出力します。

```
nnmincidentcfgdump.ovpl -dump nnm-and-customer-author-dump.tag -authorKey
com.hp.nms.author.nnm com.customer.author
```

作成者キーが Network Node Manager の管理イベントの構成を出力します。

```
nnmincidentcfgdump.ovpl -dump author-type-dump.tag -authorKey com.hp.nms.author.nnm -type
MgmtEventConfig
```

名前がNodeDown またはDuplicateCorrelation のインシデント構成を出力します。

```
nnmincidentcfgdump.ovpl -dump names-dump.tag -name NodeDown DuplicateCorrelation
```

CISCO-VTP-MIB MIB モジュールからロードされた SNMP トラップの構成を出力します。

```
nnmincidentcfgdump.ovpl -dump ciscoVtpMib.tag -mib CISCO-VTP-MIB
```

SnmpLinkDown とSnmpLinkUp トラップの構成を出力します。

```
nnmincidentcfgdump.ovpl -dump snmpLinkDownAndUp.tag -
oid .1.3.6.1.6.3.1.1.5.3 .1.3.6.1.6.3.1.1.5.4
```

すべての LinkDown トラップを出力します。CiscoLinkDown が含まれます。

```
nnmincidentcfgdump.ovpl -dump linkDownTraps.tag -oid .1.3.6.1.6.3.1.1.5.3.*
```

すべての Cisco の SNMP トラップ構成を出力します。

```
nnmincidentcfgdump.ovpl -dump ciscoSnmpTraps.tag -oid .1.3.6.1.6.3.1.1.5.*.1.3.6.1.4.1.9
```

すべての作成者キーと，そのラベルを出力します。

```
nnmincidentcfgdump.ovpl -listAuthors
```

AUTHOR

nnmincidentcfgdump.ovpl was developed by Micro Focus.

FILES

NNMi は，インシデント構成ファイルの例と，タグフォーマットファイルの正しい書式を次の場所で提供しています。

- Windows：%NnmInstallDir%examples¥nnm¥incidentcfg

- Linux：/opt/OV/examples/nnm/incidentcfg

SEE ALSO

nnmincidentcfgload.ovpl.

incidentconfiguration.format.

付録 G.26　nnmincidentcfgload.ovpl

インシデント構成ファイルをロード，または検証します。

SYNOPSIS

nnmincidentcfgload.ovpl { -load *filename* [-timeout *timeout*] [-memory *memory*] [-u <*user name*> -p <*password*>] } { -validate *filename* [-timeout *timeout*] [-memory *memory*] [-u <*user name*> -p <*password*>] } { -formats *sourceFilename* -formatd *destinationFilename* [-u <*user name*> -p <*password*>] }

DESCRIPTION

nnmincidentcfgload.ovpl はインシデント構成ファイルのロード，または評価を行います。このインシデント構成ファイルは，incidentconfiguration.format で説明されたタグフォーマットの書式に従う必要があります。

nnmincidentcfgload.ovpl がサポートするインシデント構成の種別を次に示します。

```
*MgmtEventConfig
*PairwiseConfig
*SnmpTrapConfig
*SyslogMessageConfig
```

nnmincidentcfgload.ovpl コマンドを利用する前に，次の 1 つを実施してください。

```
*nnmincidentcfgdump.ovplコマンドでタグフォーマットファイルを出力し，incidentconfiguration.fo
rmatで説明された書式に従いファイルを編集してください。
で説明された書式に従ってタグフォーマットファイルをテキストエディタで作成してください。
```

nnmincidentcfgload.ovpl コマンドを使用するとき，次の点に注意してください。

```
*複雑なインシデント構成を作成する場合，エラーを回避するために，まずnnmincidentcfgdump.ovplコ
マンドを使用してタグフォーマットファイルを作成してください。
*タグフォーマットファイルの記述内容は，既存のNNMiデータベースの内容を置き換えます。
*nnmincidentcfgload.ovplを使用してタグフォーマットファイルの再フォーマットを行うと，タグの階
層を表すためのホワイトスペースが挿入されます。ただし，コメントはすべて削除されます。
```

Parameters

nnmincidentcfgload.ovpl コマンドは，次のオプションをサポートします。

-load <*filename*>

NNMi データベースにロードするためのインシデント構成ファイルを指定します。不正なフォーマットのファイルはロードされません。NNMi はエラーをファイルの行番号とともに報告します。

-validate <*filename*>

指定されたインシデント構成ファイルを検証し，エラーを行番号とともに出力します。このオプションでは，インシデント構成ファイルは NNMi データベースにロードされません。

-formats <*sourceFilename*>

指定されたファイルを再フォーマットし，-formatd <*destinationFilename*>で指定されたファイルに出力します。

-formatd <*destinationFilename*>

-formats <*sourceFilename*>で指定したファイルを再フォーマットした結果を出力するためのファイルを指定します。

-timeout <*timeout*>

nnmincidentcfgload.ovpl コマンドのためのトランザクションタイムアウト時間を変更したい場合に秒で指定します。

-memory <*memory*>

最大ヒープサイズを MB で指定します。デフォルトは1536MB で，指定できる最小値は512MB です。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

インシデント構成ファイルをロードします。

```
nnmincidentcfgload.ovpl -load dumped-config.tag
```

インシデント構成ファイルを検証します。

```
nnmincidentcfgload.ovpl -validate modified-config.tag
```

インシデント構成ファイルを再フォーマットします。

```
nnmincidentcfgload.ovpl -formats custom.tag -formatd formatted-output.tag
```

次の構成ファイル例は，SnmpTrapConfig のためのすべての必須タグを含みます。

```
*ConfigurationType=SnmpTrapConfig
    *Name MinimalistTrapConfig
    *Oid .1.3.4.5.6
    -Author
        -Key com.customer.author
    -Category
        -Key com.hp.nms.incident.category.Fault
    -Family
        -Key com.hp.nms.incident.family.Node
    -MessageFormat Custom message format
    -Severity MINOR
```

次の構成ファイル例は，SNMP トラップにアクションを追加します。

```
*ConfigurationType=SnmpTrapConfig
    *Name MinimalistTrapConfig
    *Oid .1.3.4.5.6
    -Author
        -Key com.customer.author
    -Category
        -Key com.hp.nms.incident.category.Fault
    -ActionConfiguration
        -Actions
            -Action
                -Command echo "hello" > /tmp/hello.test
                -CommandType SCRIPT_OR_EXECUTABLE
                -LifecycleState InProgress
    -Family
        -Key com.hp.nms.incident.family.Node
    -MessageFormat Custom message format
    -Severity MINOR
```

AUTHOR

`nnmincidentcfgload.ovpl` was developed by Micro Focus.

FILES

NNMi は，インシデント構成ファイルの例と，タグフォーマットファイルの正しいフォーマットを次の場所で提供しています。

- Windows：`%NnmInstallDir%examples¥nnm¥incidentcfg`
- Linux：`/opt/OV/examples/nnm/incidentcfg`

SEE ALSO

nnmincidentcfgdump.ovpl.

incidentconfiguration.format.

付録 G.27　nnmldap.ovpl

LDAP 設定の再読み込みまたは参照

SYNOPSIS

nnmldap.ovpl -reload | -info | -diagnose <*username*> | -encrypt <*password*>

DESCRIPTION

nnmldap.ovpl は，NNMi の再起動なしで，Lightweight Directory Access Protocol（LDAP）のサインイン設定の再読み込みや，参照，評価をするコマンドです。

Parameters

nnmldap.ovpl コマンドは，次のオプションをサポートします。

-info

LDAP 設定を次のように表示します。

Configuration=providerURL:”ldap://example.com:636/”. Number of available Incident assignment users:0

-reload

LDAP 設定を再読み込みします。

-diagnose <*username*>

LDAP 設定パラメータを使用してディレクトリサービスの<*username*>にアクセスし，ldap.properties ファイルの設定を検証します。このコマンドは LDAP 設定の問題を特定する手助けとなる情報を返します。

<*username*>は，ディレクトリサービスで有効なユーザー名でなければなりません。NNMi ログイン画面の NNMi コンソールのユーザー名プロンプトで使用されているものと同じユーザー名です。

-encrypt <*password*>

ldap.properties ファイルに安全に格納できるように，指定した LDAP バインドパスワードを暗号化します。

このコマンドの出力は，ldap.properties ファイル内のbindCredential プロパティにコピーする必要があります。暗号化されたパスワードは{ENC}というプレフィックスで始まります。

暗号化されたパスワードは，それを作成したのと同じ NNMi によってだけ解読することができます。データベースがリセットされたり，プロパティが新しい NNMi システムにコピーされたりした場合，暗号化されたパスワードを新たに生成するために，このコマンドを再実行する必要があります。これに対する例外は，アプリケーションフェイルオーバーまたは HA（高可用性）構成で NNMi を使用している場合です。アプリケーションフェイルオーバーまたは HA 構成では，nnmldap.ovpl コマンドで生成した暗号化パスワードは，両方の NNMi 管理サーバーで有効です。これは，両方の NNMi 管理サーバーでデータベースが同一であるためです。

EXAMPLES

`nnmldap.ovpl -info`

現在の LDAP 設定を返します。

`nnmldap.ovpl -reload`

ldap.properties（LDAP の有効化や無効化など）の変更を読み込みます。

`nnmldap.ovpl -diagnose <username>`

ldap.properties ファイルの設定を表示して，ディレクトリサービスから情報が抽出できるか検証します。

`nnmldap.ovpl -encrypt password`

指定したパスワード文字列の暗号化された値を返します。次に例を示します。

```
{ENC}Mgnb1w007XYYenHvAFf3dQ==
```

AUTHOR

`nnmldap.ovpl` was developed by Micro Focus.

付録 G.28　nnmlicense.ovpl

Network Node Manager i のライセンス管理を行います。

SYNOPSIS

```
nnmlicense.ovpl [-h | -help]
```

```
nnmlicense.ovpl [ <PRODUCT> [(-install|-f <LicenseFile>)|-r <LicenseIndex>|(-l|-long)] ]
```

DESCRIPTION

`nnmlicense.ovpl` は，NNMi のライセンスを管理します。

ライセンスの追加は，ライセンスデータベースの更新と，実行中の NNMi プロセスに新しいライセンスが有効になったことの通知の，二つのステップで行います。自動的に実行中の NNMi プロセスに通知されるため，NNMi の再起動は必要ありません。

Parameters

PRODUCT

ライセンスされている製品のショートネーム。

`-f|-install <LicenseFile>`

指定したライセンスファイルに格納されているライセンスをインストールします。

-r|-remove <*LicenseIndex*>

インデックスで指定したライセンスを削除し，現在のライセンスステータスに反映します。ライセンスインデックスは，nnmlicense ovpl NNM -l コマンドによって返されたものと同じインデックスです。

-l|-long

現在のライセンスのステータスを表示します。

EXAMPLES

ファイル"license.txt"に格納されたライセンスパスワードをインストールする場合は，次のコマンドを実行します。

```
nnmlicense.ovpl NNM -f license.txt
```

これにより，ライセンスデータベースが更新され，ライセンス変更が NNMi に通知されます。

nnmlicense.ovpl NNM -l コマンドで表示されたライセンスインデックス「2」のライセンスを削除するには，以下を実行します。

```
$NnmInstallDir/bin/nnmlicense.ovpl NNM -l

$NnmInstallDir/bin/nnmlicense.ovpl NNM -r 2
```

AUTHOR

nnmlicense.ovpl was developed by Micro Focus.

FILES

- Windows：%NnmInstallDir%misc¥nms¥lic¥NNM.pdf
- Linux：$NnmInstallDir/misc/nms/lic/NNM.pdf

  ライセンス管理が使用する製品定義ファイル。

- Windows：%NnmDataDir%shared¥nnm¥conf¥licensing¥NNM.bin
- Linux：$NnmDataDir/shared/nnm/conf/licensing/NNM.bin

  ovjboss によって使用されるライセンス情報を表すデータファイル。

SEE ALSO

Installation Guide for future details on licensing.

付録 G.29　nnmloadattributes.ovpl

CSV ファイルまたはコマンドラインから，ノード，インタフェース，および物理コンポーネント（カード/シャーシ）に対してカスタム属性をロードします。

SYNOPSIS

nnmloadattributes.ovpl [-h | -help] [-r <*true / false*>] -t <*type*> (-f <*path & filename of csv file*>) | (-s <*"csv formatted line"*>) [-u <*username*> -p <*password*>]

DESCRIPTION

nnmloadattributes.ovpl は，CSV ファイルからカスタム属性をロードします。 外部データストアに定義されている多くのノード，インタフェースまたは物理コンポーネントがあり，NNMi にこれらの属性をロードしたい場合，このコマンドは有用です。ノードについては，NNMi へロードした後に，それらのカスタム属性によってノードをグループ化するためにノードグループ形式を使用することができます。

Parameters

nnmloadattributes.ovpl コマンドは，次のオプションをサポートします。

-h | -help

コマンドの使用方法を表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-r <*true / false*>

この引数が値 true で指定されていない場合，既存の属性値は変更されません。

-t <*type*>

ロード対象のオブジェクトタイプを指定します。"node"，"interface" または "physcomp"です。

-f <*path & filename of csv file*>

ロードするカスタム属性を含む CSV ファイル名（パス名。例：/tmp/csvfile.csv）を指定します。

-s <*"csv formatted line"*>

CSV 形式の 1 行を指定します。小さな変更に対してファイル作成を省略できます。

Syntax of Comma Separated File For Node Attributes

ノードにカスタム属性を追加するときに指定する CSV ファイルの構文は次のとおりです。

空の行は無視されます。

"#"で始まる行は無視されます。

- Column 1(A)：Node DNS|IP Address
  ノードの DNS 名か，IP アドレスを指定します。このフィールドは必須です。
- Column 2(B)：Attribute Name
  カスタム属性の名前。
- Column 3(C)：Attribute Value
  カスタム属性の値。

追加のカスタム属性の名前と値の組み合わせは，同じ行に指定できます。または，別の行で同じノードを指定します。

例）

192.168.1.1,Location,Building Five Upper,Service Type,eCommerce

192.168.1.1,Asset Tracking,N1234

192.168.2.2,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321

Syntax of Comma Separated File For Interface Attributes

インタフェースにカスタム属性を追加するときに指定する CSV ファイルの構文は次のとおりです。

空の行は無視されます。

"#" で始まる行は無視されます。

- Column 1(A)：Node DNS|IP Address
  ノードの DNS 名か，IP アドレスを指定します。このフィールドは必須です。
- Column 2(B)：Interface Id
  前のフィールドで指定したノードのインタフェースの識別子を指定します。インタフェースのインデックス，エイリアス，インタフェース名，または説明が指定でき，この順序で検索されます。すべての一致するインタフェースに対して，属性がロードされます。このフィールドは必須です。
- Column 3(C)：Attribute Name
  カスタム属性の名前。
- Column 4(D)：Attribute Value
  カスタム属性の値。

追加のカスタム属性の名前と値の組み合わせは，同じ行に指定できます。または，別の行で同じノードとインタフェース識別子を指定します。

例）

192.168.1.1,1001,Location,Building Five Upper,Service Type,eCommerce

192.168.1.1,1001,Asset Tracking,N1234

192.168.2.2,A1,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321

Syntax of Comma Separated File For PhysComp Attributes

物理コンポーネント（カード/シャーシ）にカスタム属性を追加するときに指定する CSV ファイルの構文は次のとおりです。

空の行は無視されます。

"#"で始まる行は無視されます。

- Column 1(A)：Node DNS | IP Address
  ノードの DNS 名か，IP アドレスを指定します。このフィールドは必須です。
- Column 2(B)：PhysComp Id
  前のフィールドで指定したノードの物理コンポーネントの識別子を指定します。物理コンポーネントの物理インデックス，名前，または説明が指定でき，この順序で検索されます。 すべての一致する物理コンポーネントに対して，属性がロードされます。このフィールドは必須です。
- Column 3(C)：PhysComp Type
  物理コンポーネントのタイプを識別する名前。"card"と"chassis"が有効なタイプです。
- Column 4(D)：Attribute Name
  カスタム属性の名前。
- Column 5(E)：Attribute Value
  カスタム属性の値。

追加のカスタム属性の名前と値の組み合わせは，同じ行に指定できます。 または，別の行で同じノードと物理コンポーネント識別子，物理コンポーネントタイプを指定します。

例）

192.168.1.1,7,chassis,Location,Building Five Upper,Service Type,eCommerce

192.168.1.1,7,chassis,Asset Tracking,N1234

192.168.2.2,/AmdFE,card,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321

Error Codes

問題を特定するのに役立ついくつかのエラーコードがあります：

- INFO：情報メッセージ。
- ATTR_ERROR：指定された属性で問題が見つかりました。
- DEL_FAIL_ERROR：指定された属性の削除に失敗しました。
- OBJECT_ERROR：指定されたオブジェクトが見つかりませんでした。
- BAD_LINE_ERROR：指定された行のフォーマットが正しくありません。
- IO_ERROR：CSV ファイルが見つからないか，読めませんでした。
- BAD_NAME_WARNING：指定された属性名が長すぎます(最大 50 文字)。
- BAD_VALUE_WARNING：指定された値が長すぎます(最大 2000 文字)。

EXAMPLES

CSV ファイル(/tmp/test.csv)の内容例は次のようになります (ノードの場合)：

```
192.168.2.2,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321
```

CSV ファイルからノードのカスタム属性を既存の値を上書きしてロードする場合：

```
nnmloadattributes.ovpl -t node -f /tmp/test.csv -r true
```

コマンドに指定したノードのカスタム属性を一つロードする場合：

```
nnmloadattributes.ovpl -t node -s "192.168.1.1,Project,IT Update of Building Five"
```

CSV ファイル(/tmp/test.csv)の内容例は次のようになります(インタフェースの場合)：

```
192.168.2.2,A1,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321
```

CSV ファイルからインタフェースのカスタム属性を既存の値を上書きしてロードする場合：

```
nnmloadattributes.ovpl -t interface -f /tmp/test.csv -r true
```

コマンドに指定したインタフェースのカスタム属性を一つロードする場合：

```
nnmloadattributes.ovpl -t interface -s "192.168.1.1,1001,Project,IT Update of Building Five"
```

CSV ファイル (/tmp/test.csv) の内容例は次のようになります(物理コンポーネントの場合)：

```
192.168.2.2,7,chassis,Location,Fort Collins,Service Type,IT,Asset Tracking,F4321
```

CSV ファイルから物理コンポーネントのカスタム属性を既存の値を上書きしてロードする場合：

```
nnmloadattributes.ovpl -t physcomp -f /tmp/test.csv -r true
```

コマンドに指定した物理コンポーネントのカスタム属性を一つロードする場合：

```
nnmloadattributes.ovpl -t physcomp -s "192.168.1.1,/AmdFE,card,Project,IT Update of Building
Five"
```

AUTHOR

nnmloadattributes.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmloadattributes.ovpl
- Linux：$NNM_BIN/nnmloadattributes.ovpl

SEE ALSO

nnmdeleteattributes.ovpl, nnmloadnodegroups.ovpl, nnm.properties.

付録 G.30　nnmloadinterfacegroups.ovpl

このコマンドは，CSV ファイルからインタフェースグループ定義を読み込みます。

SYNOPSIS

nnmloadinterfacegroups.ovpl [-?] [-u <*username*> -p <*password*>] [-r true | false] -f <*csv_filename*>

DESCRIPTION

nnmloadinterfacegroups.ovpl コマンドは，CSV ファイルからインタフェースグループの定義を読み込みます。このコマンドは，外部データストアで定義されているインタフェースデータが多数あり，かつ，インタフェースグループの定義として NNMi のデータベースに読み込みたい場合に便利です。NNMi に読み込んだ後，インタフェースグループフォームを使用すると，インタフェースグループの定義を変更できます。

Parameters

nnmloadinterfacegroups.ovpl コマンドは，次のオプションをサポートします。

-?

コマンドの使用方法を表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-r true | false

このオプションを使用する前に，既存のインタフェースグループ設定をバックアップしてください。

-r false（デフォルト設定）の場合，インタフェースグループ名が既に NNMi データベースに存在すると，nnmloadinterfacegroups.ovpl コマンドは以前の設定を変更しません。

-r true の場合，同じ Name（Column 1）の既存のインタフェースグループのすべての設定が，CSV ファイルの値で上書きされます。

注意：マージではなく，そのインタフェースグループのすべての設定が完全に置き換えられます。

-f <*csv_filename*>

インタフェースグループを定義した CSV ファイルを指定します。

Syntax of Comma-Separated File

注意：改行文字，カンマ，引用符などの特殊文字を含むフィールドは，CSV ファイルでは二重引用符で囲む必要があります。 二重引用符が埋め込まれたフィールドも，RFC4180（CSV ファイルの一般的書式，および MIME タイプ）の仕様に準拠し，二重引用符で囲まれたフィールド中の二つの二重引用符によるエスケープされた形式を使用する必要があります。

ユーザーが指定する CSV ファイルには次の構文が必要です。

（必須：Column1）インタフェースグループ名を指定する必要があります。

（任意：Column2-Column7）値のないカラムは空にしてください。

Column7 の終わりを示すカンマは必要ありません。カラム内で最後の項目の後（とカンマ「,」の間）にセミコロン「;」は必要ありません。

NNMi は次の方法で，すべての設定の結果を統合します。

1. NNMi は最初に ifType Filters（Column6）を評価します。このインタフェースグループに含まれるためには，インタフェースは少なくとも一つの定義に一致する必要があります。
2. NNMi は次に Additional Filters（Column7）を評価します。このインタフェースグループに含まれるためには，すべての Additional Filters の定義もパスする必要があります。
3. Node Group（Column5）がこのインタフェースグループに指定されている場合，このグループ内の任意のインタフェースは，そのノードグループのメンバーであるノードに含まれている必要があります。

空白行または「#」で始まる行はコメントとして無視されます。次のコメントを 1 行目に追加することで，必要となるカラムの構文の参考になります。

#InterfaceGroupName,[Notes],[AddtoFilterList],[AddtoPerformanceFilterList],
[NodeGroupName],[ifType1;...],[AdditionalFilters]

- Column1(A)：Interface Group Name

  （必須）インポートしたいインタフェースグループの名前を指定します（インタフェースグループフォームの名前属性の値になります）。

- Column2(B)：Notes

  （任意）インタフェースグループの説明を記述します（インタフェースグループフォームの「注」フィールドのテキストになります）。

- Column3(C)：Add to View Filter List

  （任意）インタフェースグループフォームの「ビューフィルターリストに追加」を設定します。

  1（デフォルト設定）の場合，インタフェースビューなどのテーブルビューを表示するとき，ドロップダウンリストでこのインタフェースグループが利用可能になります。

  0 の場合，ビュードロップダウンフィルターリストにこのインタフェースグループを含めません。

  推奨：使用頻度が最も多いインタフェースグループでのみ，この値を 1 に設定してください。極端に多くのインタフェースグループで設定すると，リストが長くなり使用しにくくなります。

- Column4(D)：Add to Performance Filter List

  （任意）インタフェースグループフォームの「NNMi iSPI Performance」を設定します。NNMi では使用できませんので，何も指定しないでください。

- Column5(E)：Node Group Name

  （任意）指定されたノードグループは，このインタフェースグループのフィルタとして機能します。

  注意：既に NNMi データベースに存在するノードグループを指定してください。

- Column6(F)：ifType Filters

  （任意）「;」で区切って，ifType フィルターの設定を追加します（インポート後，これらの定義はインタフェースグループフォームの ifType フィルタータブに表示されます）。それぞれの ifType は ifType 名によって識別されます。

  NNMi のコンソールに表示される正確な ifType 名を指定します。

  ifType フィルターに指定するエントリの例は次のとおりです。

  - ds0;ds0Bundle;ds1;ds1FDL;ds3;g703at2mb
  - ppp;pppMultilinkBundle;propPointToPointSerial;slip
  - ethernetCsmacd

- Column7(G)：Additional Filters

  （任意）インタフェースグループに含まれるインタフェースを絞り込むために使用される追加のフィルター式を指定します。追加のフィルターのフォーマットは次のとおりです。

  1. 一致する括弧のセット内に，フィルターの条件演算子とそれに関連付けられているフィルター条件を定義します。

2. フィルター属性，フィルター演算子，フィルター値の順番で指定して，フィルターを定義します。

インタフェースグループフォームで使用可能なすべてのフィルター属性と演算子がサポートされています。複数のフィルター値を指定する場合は，「:」を区切り文字として使用します。一つのフィルター条件演算子に複数のフィルターを指定する場合は，「;」を区切り文字として使用します。スペースは，フィルター属性とフィルター演算子，フィルター演算子とフィルター値を区切るために使用されます。

フィルター属性：

- ifAlias
- ifDesc
- ifIndex
- ifName
- ifSpeed
- hostedOn
- ipAddress
- isSnmpInterface
- sysOidInterface
- devCategoryInterface
- devVendorInterface
- devFamilyInterface
- customAttrName
- customAttrValue
- capability
- vlanId
- vlanName
- ifPhysAddress
- configuredDuplexSetting

フィルター演算子：

- !=
- >
- >=
- <
- <=
- =

- between
- in
- is_not_null
- is_null
- like
- not_between
- not_in
- not_like

フィルター条件演算子：

- AND
- OR
- NOT
- EXISTS
- NOT_EXISTS

例：

- (AND hostedOn like *.mycompany.com (OR (EXISTS (AND customAttrName = circuit; customAttrValue = 12)) (EXISTS (AND customAttrName = circuit; customAttrValue = 15))))
- (AND hostedOn like *.mycompany.com (EXISTS (AND customAttrName = circuit; customAttrValue in 12:15)))
- hostedOn like *.mycompany.com
- ifAlias = ” Alias with leading and trailing spaces ”
- ifAlias = Alias with embedded ¥”double quotes¥”¥

カラム 7 では，ダブルクォートをダブルクォートでエスケープする，RFC4180 の規則は適用されません。その結果，次のようなファイルを作成します。
”My Group”,,,,,,”ifAlias = Alias with embedded ¥”double quotes¥””

Note

フィルター値を入力する場合，特殊文字「"」，「(」，「)」，「:」および「;」は使用しないでください。特殊文字を使用したい場合は，「¥」でエスケープしてください。例えば

- 'circuit:57' は 'circuit¥:57' のように入力します。
- 'circuit(57)' は 'circuit¥(57¥)' のように入力します。
- 'circuit"57"' は 'circuit¥"57¥"' のように入力します。
- 'circuit;57' は 'circuit¥;57' のように入力します。

- 'circuit¥:57' は 'circuit¥¥¥:57' のように入力します。

カラム 7 では，ダブルクォートをダブルクォートでエスケープする，RFC4180 の規則は適用されません。その結果，次のようなファイルを作成します。

```
”My Group”,,,,,,”ifAlias = circuit¥(57¥)”

”My Group”,,,,,,”ifAlias = circuit¥”57¥””
```

Use of Microsoft Excel

カンマ区切りファイルを作成する場合，Microsoft Excel が手ごろなツールですが，CSV ファイルではカラム幅，コメントなどが維持されません。nnmloadinterfacegroups.ovpl の入力ファイルをネイティブの XLS フォーマットとして保存後に，"名前を付けて保存(A)..."を実行して CSV ファイルを作成するのが賢明です。この場合，XLS フォーマットのファイルに Excel コメントを追加し，カラム幅を拡大することが可能であり，カンマ文字をエスケープする必要もありません。

Microsoft Excel で先頭に#がある行でカンマを入力すると，XLS ファイルを CSV ファイルとして保存したときに，コメントなしのエントリができます（#で始まる名前を持つインタフェースグループが作成されます）。

EXAMPLES

CSV ファイルの内容例は次のようになります。

```
Point to Point Interfaces,Point to Point Interfaces are usually associated with dial-
up.,1,,,ppp;pppMultilinkBundle;propPointToPointSerial;slip
```

CSV ファイルのカラム 1 に定義した Name に一致する既存のインタフェースグループを上書きしないように，CSV ファイルからインタフェースグループを読み込むには；

```
nnmloadinterfacegroups.ovpl -u system -p myadminpasswd -f /tmp/test.csv
```

CSV ファイルのカラム 1 に定義した Name に一致する既存のインタフェースグループを上書きして，CSV ファイルからインタフェースグループを読み込むには：

```
nnmloadinterfacegroups.ovpl -u system -p myadminpasswd -r true -f /tmp/test.csv
```

AUTHOR

nnmloadinterfacegroups.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmloadinterfacegroups.ovpl
- Linux：$NNM_BIN/nnmloadinterfacegroups.ovpl

SEE ALSO

nnmconfigimport.ovpl, nnmloadnodegroups.ovpl, nnm.properties.

付録 G.31　nnmloadipmappings.ovpl

重複する IP アドレスマッピングの情報を読み込みます。

SYNOPSIS

nnmloadipmappings.ovpl -f *mapping file* [-u <*username*> -p <*password*>]

DESCRIPTION

nnmloadipmappings.ovpl を使用すると，静的 NAT［RFC2663］環境で設定した IP アドレスのマッピングをテキストファイルからロードできます。ロードされたマッピングは，対応する IP アドレスのインベントリに追加されます。

-f オプションは，1 行に 1 エントリを指定したファイルを受け付けます。各行の形式を次に示します。

Tenant Name, "Public IP Address", "Private IP Address"

指定内容の説明：

Tenant Name＝テナント名。*Public IP Address*＝外部ネットワークにさらされる，ネットワークアドレス変換（NAT）した特定の IPv4 アドレス。*Private IP Address*＝ネットワークアドレス変換（NAT）したパブリック IP アドレスに対応する，ある特定の内部 IPv4 アドレス。

#を区切り文字として複数のコメントを指定できます。

一つのパブリック IP アドレスは，一つのテナント内の一つのプライベート IP アドレスに対してのみマッピングできることに注意してください。同様に，一つのテナント内のプライベート IP アドレスは一つのパブリック IP アドレスに対してのみマッピングできます。ただし，デバイス上の複数のマッピングがサポートされています。

Parameters

nnmloadipmappings.ovpl コマンドは，次のオプションをサポートします。

-f *mapping file*

IP アドレスマッピングの読み込み元テキストファイルを指定します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

Tenant1Mappings.txt という名前のファイルから IP アドレスマッピングを読み込みます。

```
nnmloadipmappings.ovpl -f /tmp/Tenant1Mappings.txt
```

ユーザー名とパスワードを指定して，Tenant2Mappings.txt という名前のファイルから IP アドレスマッピングを読み込みます。

```
nnmloadipmappings.ovpl -u username -p password -f /tmp/Tenant2Mappings.txt
```

AUTHOR

nnmloadipmappings.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmloadipmappings.ovpl
- Linux：$NNM_BIN/nnmloadipmappings.ovpl

付録 G.32　nnmloadmib.ovpl

SNMP MIB のロードとアンロードを行います。

SYNOPSIS

nnmloadmib.ovpl [[-load *mib-file*] [-unload *mib-module*[:*mib-module*...]] [-list] [-u *username*] [-p *password*]]

DESCRIPTION

nnmloadmib.ovpl は，NNMi が SNMP Object Identifiers（OIDs）を数値形式から可読テキスト形式に変換するために，Internet SMI（Structure of Management Information）形式の SNMP MIB（Management Information Base）を読み込みます。NNMi コンソールで MIB 式を作成する前に，MIB をロードする必要があります。NNMi は SMI Version 1（RFC1155，1212，1215）および SMI Version 2（RFC2578）形式をサポートします。

nnmloadmib.ovpl コマンドは，NNMi で使用するために，MIB モジュールをコンパイルしてロードし，得られる情報を NNMi データベースに格納します。

MIB から TRAP-TYPE および NOTIFICATION-TYPE マクロを NNMi のインシデントの設定にロードする場合は，nnmloadmib.ovpl コマンドで NNMi の MIB データベースに MIB をロードした後に，nnmincidentcfg.ovpl コマンドを使用します。

Parameters

nnmloadmib.ovpl コマンドは，次のオプションをサポートします。

-load *mib-file*

mib-file に指定したファイルのコンテンツを MIB データベースに読み込みます。

注意：nnmloadmib.ovpl コマンドを実行する前に，*mib-file* ファイルを%NNM_DATA%¥shared¥nnm¥user-snmp-mibs（Windows）ディレクトリまたは，$NNM_DATA/shared/nnm/user-snmp-mibs（Linux）ディレクトリ（またはその配下のディレクトリ）にコピーすることを推奨します。これにより，アクション->MIB ファイルを表示メニューと-list オプションで，MIB ファイルの元の場所を見つけることができます。

-unload *mib-module*[:*mib-module*...]

MIB データベースから*mib-module* のリストをアンロードします。*mib-module* とは，MIB データベースに読み込まれている MIB モジュールの名前です。

注意：もしnnmloadmib.ovpl スクリプトを使って MIB をアンロードしてnnmincidentcfg.ovpl スクリプトを使って MIB からのインシデント設定をアンロードしたい場合，先にインシデント設定のアンロードをしてください。これはnnmincidentcfg.ovpl スクリプトではアンロードされた MIB モジュール名は指定できないためです。

-list

データベースに読み込まれている MIB の一覧を表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

Loading/Unloading Validation

MIB のロードおよびアンロード時に，既存のすべてのロード済み MIB は，ロード/アンロード操作が問題なく実行されることを保障するために解析されます。そのため，ロードおよびアンロード対象の MIB とは必ずしも関係のない警告（OID が重複しているなど）が表示されることがあります。コマンドラインの実行時に成功のステータスが返されていれば問題はないため，これらの警告は無視することができます。

Syntax of MIB Files

MIB ファイルに関連する構文のほとんどは，さまざまな RFC ドキュメントに記述されています。このページの「SEE ALSO」の項を参照してください。

Diagnostics

nnmloadmib.ovpl コマンドは，次の終了コードを返します。

0

コマンドの実行に成功しました。

1

コマンドラインの指定が正しくありません。

20

コマンドを実行した後に予期しない例外が発生しました。

21

指定した資格情報を使ってコマンドを実行することは許可されませんでした。

22

MIB ファイル内に構文エラーを検出したか，サービスに不具合があったため MIB を読み込めませんでした。

23

不正な引数が使用されました。多くの場合，必要なファイル名を指定せずに，-load のようなオプションが指定されています。

24

コマンドは，NNMi で実行されている MIB ローダーサービスと通信できませんでした。

25

引数が指定されていません。このコマンドには，引数を指定する必要があります。

27

NNMi は動作中ですが，MIB ローダーサービスを解決できません。

30

データベースに問題があるため，一覧表示操作に失敗しました。

nnmloadmib.ovpl コマンドに失敗すると，MIB に関する潜在的な問題を分析し解決するために，説明的なエラーメッセージが表示されます。すべてのエラーメッセージは，同じような形式です。

```
SEVERITY: MESSAGE FILENAME:LINE_NUMBER:COLUMN_NUMBER: DETAIL_MESSAGE
```

一般的な障害と推奨される解決策を次に示します。

ERROR : Cannot find symbol file:///tmp/CHECKPOINT-MIB.mib:Line 2620:Column 16:cpvTNlMonCurrAddr

記載されているシンボル名は，読み込まれる MIB 内に宣言として見つからなかったか，MIB 定義の上部に import として記載されていません。これは，シンボル名の入力ミスか，import 宣言がないために発生している可能性があります。

ERROR : Cannot find symbol file:///var/opt/OV/shared/nnm/user-snmp-mibs/example.mib:Line 13233:Column 16:COUNTER64

MIB ファイルを NNMi に適切にロードするには，MIB ファイルが SNMP SMI v1（RFC1155）または SMI v2（RFC2578）のいずれかの規格に準拠している必要があります。大文字と小文字の区別や正しい MIB 定義からのオブジェクトのインポートについて，特に注意する必要があります。場合によっては，旧バージョンの NNMi で正しくロードされていた MIB を修正する必要があるかもしれません。一般的な例としては，Counter64 が正しい定義のときに誤って COUNTER64 を使用してしまう場合です。例えば，"SYNTAX COUNTER64"などは，"SYNTAX Counter64"に修正する必要があります。

ERROR : Found symbol file:///tmp/CHECKPOINT-MIB.mib:Line 3509:Column 27:routingDest but expected a class org.jsmiparser.smi.SmiType instead of class org.jsmiparser.smi.SmiVariable

記載されているシンボル名は，期待されていたタイプではありません。この場合，SMI タイプが期待されていましたが，MIB 変数名が代わりに提供されました。このケースの解決策は，正しい SMI タイプ，つまり IP アドレスを指定することです。

ERROR : Cannot find module file:///tmp/rfc1472-PPP-SEC-MIB.mib:Line 9:Column 26:PPP-LCP-MIB

このエラーは，import として記載されている MIB モジュールが読み込まれていないため，結果として，パーサーが import できないことを示しています。解決策は，参照されている MIB を最初に読み込むことです。

ERROR file:///tmp/vinemib2:Line 4360:Column 17: Parse error: unexpected token: --#

NNMi は，TRAP-TYPE または NOTIFICATION-TYPE のマクロ定義で指定できる，一部のカスタムトラップメッセージ形式情報をサポートしています。有効な値を次に示します。

```
--#TYPE
--#SUMMARY
--#ARGUMENTS
--#SEVERITY
--#GENERIC
--#CATEGORY
--#SOURCE_ID
--#TIMEINDEX
--#HELP
--#HELPTAG
--#STATE
```

このエラーは，--#の後に無効なキーワード，または NNMi が予期しない一連のキーワードを指定したことを示しています。この問題を修正するには，上記のリストに対応しない --# エントリを削除するか，行の始めに余分なコメント文字シーケンス（--）を追加します。

EXAMPLES

MIB ファイル $NNM_DATA/shared/nnm/user-snmp-mibs/corp.mib を読み込む場合，次のように nnmloadmib.ovpl コマンドを実行します。

```
nnmloadmib.ovpl -load $NNM_DATA/shared/nnm/user-snmp-mibs/corp.mib -u user -p password
```

ロードされている MIB の一覧を表示する場合，次のようにnnmloadmib.ovpl コマンドを実行します。

```
nnmloadmib.ovpl -list -u user -p password
```

AUTHOR

nnmloadmib.ovpl was developed by Micro Focus.

FILES

Windows：

```
%NnmInstallDir%misc¥nnm¥snmp-mibs¥*
%NNM_DATA%¥shared¥nnm¥user-snmp-mibs¥*
```

Linux：

```
$NnmInstallDir/misc/nnm/snmp-mibs/*
$NNM_DATA/shared/nnm/user-snmp-mibs/*
```

SEE ALSO

RFC 2578 Structure of Management Information Version 2 (SMIv2)

RFCs 1155, 1212, 1215: SNMP Version 1 Structure of Management Information

RFCs 1902, 1903, 1904: SNMP Version 2 Structure of Management Information

nnmincidentcfg.ovpl, nnmsnmpwalk.ovpl.

付録 G.33　nnmloadnodegroups.ovpl

このコマンドは，CSV ファイルからノードグループ定義を読み込みます。

SYNOPSIS

nnmloadnodegroups.ovpl [-?] [-u <*username*> -p <*password*>] [-r true | false] -f <*csv_filename*>

DESCRIPTION

nnmloadnodegroups.ovpl は，CSV ファイルからノードグループの定義を読み込みます。このコマンドは，外部データストアで定義されているノードデータが多数あり，かつ，ノードグループの初期定義として NNMi のデータベースに読み込みたい場合に便利です。NNMi に読み込んだ後，ノードグループフォームを使用すると，ノードグループの定義を変更できます。

次の設定は，CSV ファイルで設定することはできません。ノードグループを読み込んでから，ノードグループフォームを使用してデフォルト設定を変更する必要があります。

- ステータスの計算 = true（NNMi はこのノードグループのステータスを計算します）

Parameters

nnmloadnodegroups.ovpl コマンドは，次のオプションをサポートします。

-?

コマンドの使用方法を表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。 nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。 nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-r true | false

このオプションを使用する前に，既存のノードグループ設定をバックアップしてください。
-r false（デフォルト設定）の場合，ノードグループ名が既に NNMi データベースに存在すると，nnmloadnodegroups.ovpl コマンドは以前の設定を変更しません。
-r true の場合，同じ Name（Column 1）の既存のノードグループのすべての設定が，CSV ファイルの値で上書きされます。
注意：マージではなく，そのノードグループのすべての設定が完全に置き換えられます。
注意：階層内のグループでこの置き換えオプションを使用する場合は，すべての子グループが同一の CSV ファイルに存在する必要があります。また，子ノードグループは親ノードグループよりも先に記載してください。

-f <*csv_filename*>

ノードグループを定義した CSV ファイルを指定します。

Syntax of Comma Separated File

ユーザーが指定する CSV ファイルには次の構文が必要です。

（必須: Column 1）Node Group Name の値を指定する必要があります。

（任意: Column 2 - Column 14）値のないカラムは空にしてください。

Column 14 の終わりを示すカンマは必要ありません。カラム内で最後の項目の後（とカンマ「,」の間）にセミコロン「;」は必要ありません。

NNMi は次の方法で，すべての設定の結果を統合します。

1. NNMi は最初に Device Filters（Column 5）を評価します。このノードグループに含まれるためには，ノードは少なくとも一つの定義に一致する必要があります。
2. NNMi は次に Additional Filters（Column 7 - Column 14）を評価します。このノードグループに含まれるためには，すべての Additional Filters の定義もパスする必要があります。

   注意：読み込んだ後で，ノードグループフォームの追加のフィルタータブにカラム 7-14 の設定が統合されて表示されます。ノードグループフォームの追加のフィルターエディタで，デフォルトの論理演算を変更できます。
3. フィルターに関係なく，すべての Additional Nodes（Column 6）はノードグループに含まれます。
4. すべての Child Node Groups（Column 4）の結果は，Additional Nodes と同様の扱いになります。

空のラインまたは「#」で始まるラインはコメントとして無視されます。次のコメントを 1 行目に追加することで，必要となるカラムの構文の参考になります。

#[NodeGroupName],[Notes],[AddtoFilterList],
[ChildNodeGroup:0/1;...],DeviceFilter[Category1:Vendor1:Family1:Profile1;...],AdditionalNodes[
Fully-Qaul-hostname;...],AdditionalFilters > [hostname;...],[hostedIPAddress;...],
[mgmtIPAddress;...],[customAttrName/customAttrValue;...],[capability;...],
[SecurityGroupDetails;...],[TenantDetails;...],[NodeName;...]

- Column 1(A)：Node Group Name

  （必須）インポートしたいノードグループの名前を指定します（ノードグループフォームの名前属性の値になります）。
- Column 2(B)：Notes

  （任意）ノードグループの説明を記述します（ノードグループフォームの「注」フィールドのテキストになります）。
- Column 3(C)：Add to View Filter List

  （任意）ノードグループフォームの「ビューフィルターリストに追加」を設定します。

  1（デフォルト設定）の場合，ノードビューなどのテーブルビューを表示するとき，ドロップダウンリストでこのノードグループが利用可能になります。

  0 の場合，ビュードロップダウンリストにこのノードグループを含めません。

  推奨：最上位の親ノードグループまたは使用頻度が最も多いノードグループでのみ，この値を1 に設定してください。極端に多くのノードグループで設定すると，リストが長くなり使用しにくくなります。

- Column 4(D) ：Child Node Groups

  （任意）「;」で区切って，子のノードグループの一覧を指定します（ノードグループフォームの子のノードグループタブに表示されます）。

  注意：子のノードグループを設定する場合，既に NNMi データベースに存在するか，同じ CSV ファイルで定義されているノードグループを指定してください。

  例：`ChildNodeGroup1:1[;ChildNodeGroup2:0;...]`

  `0`（デフォルトの設定）の場合，子のノードグループが親のノードグループのマップにノードグループアイコンで表示されます。

  `1` の場合，親のノードグループのマップに子のノードグループを拡張します。親ノードグループに定義されているかように，すべてのノードを表示します。

  子ノードグループに対する有効なエントリは次のとおりです。

  - computers:1
  - computers:0
  - computers:
  - computers:;printers:1

- Column 5(E) ：Device Filters

  （任意）「;」で区切って，デバイスフィルターを追加します（ノードグループフォームのデバイスフィルタータブに表示されます）。各フィルターは，次の書式のように，「:」で区切られた四つの部分で構成されています。

  Category1:Vendor1:Family1:Profile1[;Category2:Vendor2:Family2:Profile2 ...]

  一致するフィルターの件数を増やすために，フィルター指定の一部を省略することができます。例えば，Category1 および Vendor1 に対する任意のファミリを一致させたい場合，次のエントリを追加してください。

  `Category1:Vendor1::`

  filter1 に対するファミリ，および filter2 に対するファミリとプロファイルを入力したくない場合，次のエントリを指定してください。

  `Category1:vendor1::profile1;Category2:vendor2::;`

  デバイスプロファイルに対する有効なエントリの例は次のとおりです。

  - com.hp.ov.nms.devices.printer:com.hp.ov.nms.devices.hewlettpackard::.1.3.6.1.4.1.9.1.380
  - com.mycomp.ov.nms.devices.printer:com.hp.ov.nms.devices.mycompanyname::
  - com.hp.ov.nms.devices.printer:::
  - :::.1.3.6.1.4.1.9.1.380

- Column 6(F) ：Additional Nodes

  （任意）「;」で区切られたこのノードグループに追加したいノードホスト名の一覧を指定します（ノードグループフォームの追加のノードタブに表示されます）。ホスト名はノードフォームのホスト名属性に表示される現在の完全な大文字小文字を区別した値を指定します。

例えば：hostname1.x.y.z;hostname2.x.y.z;hostname3.x.y.z

- Column 7(G) ：Additional Filters "hostname" code (Hostname Wildcards)

  (任意)「;」で区切られたホスト名ワイルドカードの一覧を指定します（演算子「like」と同じ)。その他の演算子を使用する場合は，ノードグループフォームを使用してください（追加のフィルタータブに表示されます)。

  例えば：*.cnd.hp.com;*snmp.hp.com

- Column 8(H) ：Additional Filters "hostedIPAddress" code (Hosted IP Address Ranges)

  (任意)「;」で区切られた所有する IP アドレスの範囲の一覧を指定します（演算子「between」と同じ)。その他の演算子を使用する場合は，ノードグループフォームを使用してください（追加のフィルタータブに表示されます)。範囲の指定には，下方アドレスと上方アドレスをダッシュで区切ってください。両端のアドレスが範囲に含まれます。単一の IP アドレスを範囲に指定する場合，下方アドレスと上方アドレスの両方に同一の値を使用してください。ノード上の任意のアドレスがこの範囲に一致する場合，ノードグループにこのノードが指定されます。

  有効な例：10.20.30.1-10.20.30.254;192.168.177.1-192.168.180.254;1.1.1.1-1.1.1.1

- Column 9(I) ：Additional Filters "mgmtIPAddress" code (Management Address Ranges)

  (任意)「;」で区切られた管理アドレス範囲の一覧を指定します（演算子「between」と同じ)。範囲は所有する IP アドレスの範囲と同じ形式です。その他の演算子を使用する場合は，ノードグループフォームを使用してください（追加のフィルタータブに表示されます)。SNMP をサポートしているノードのみ，スパイラル検出が管理 IP アドレスを作成することに注意してください。スパイラル検出で管理アドレスを選択する方法の詳細については，ノードフォームの管理アドレスフィールドに対するオンラインヘルプを参照してください。

- Column 10(J) ：Additional Filters "customAttrName:customAttrValue" codes (Custom Node Attributes)

  (任意) '"カスタム属性名" オペレータ "カスタム属性値"[;...]' のフォーマットで，ノードのカスタム属性を記述します。名前と値は引用符で囲む必要があることに注意してください（単一引用符（')は定義に含みません。このヘルプで先頭と末尾を示すためにのみ使用)。Custom Node Attributes の設定は，ノードフォームの追加のフィルタータブに表示されます。

  オペレータの有効な値は次のとおりです。

  =, !=, like, not like, >, >=, <, <=（その他の演算子を使用する場合は，ノードグループフォームを使用してください)。

  複数のカスタム属性文は，セミコロンで区切って指定できます。例えば，'"Location" = "Bldg. Five";"Service Type" = "eCommerce"' です。複数の "customAttrName:customAttrValue" の記述は「AND」で処理されます。つまり，ノードがノードグループに含まれるには，すべての文が true である必要があります。

- Column 11(K) ：Additional Filters "capability" code (Capabilities)

  (任意) 'capability オペレータ "ケーパビリティ値"[;...]' のフォーマットで，ノードのケーパビリティを記述します。値は引用符で囲む必要があることに注意してください（単一引用符（')は定義に含みません。このヘルプで先頭と末尾を示すためにのみ使用)。Capabilities の設定は，ノードフォームの追加のフィルタータブに表示されます。

オペレータの有効な値は次のとおりです。

=, !=, like, not like（その他の演算子を使用する場合は，ノードグループフォームを使用してください）。

複数のケーパビリティ文はセミコロンで区切って指定できます。例えば，'capability = ”com.hp.ov.nms.isLANSwitch”;capability != ”com.hp.ov.nms.isIPv4Router”' です。 複数のケーパビリティ文は「AND」で処理されます。つまり，ノードがノードグループに含まれるには，すべての文が true である必要があります。

- Column 12(L) ：Security Group Details

  （任意）このノードグループに追加するセキュリティグループのプロパティの一覧を，セミコロン「;」で区切って指定します。このノードグループを UUID でセキュリティグループに関連付けるには，セキュリティグループの UUID を（xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx）形式で入力します。ここで，（x）は 16 進数（0～9a～fA～F）であり，そうでない場合は，ノードグループは名前によってセキュリティグループに関係付けられていると仮定されます。

  例えば，12345678-1234-1234-1234-123456123456;test_security_group_name です。

- Column 13(M) ：Tenant Details

  （任意）このノードグループに追加するテナントのプロパティの一覧を，セミコロン「;」で区切って指定します。このノードグループを UUID でテナントに関連付けるには，テナントの UUID を（xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx）形式で入力します。ここで，（x）は 16 進数（0～9a～fA～F）であり，そうでない場合は，ノードグループは名前によってテナントに関係付けられていると仮定されます。

  例えば，12345678-1234-1234-1234-123456123456;test_tenant_name です。

- Column 14(N) ：Node Name

  （任意）このノードグループに追加するノードの一覧をセミコロン「;」で区切って指定します。このノードグループをノード名でノードに関連付けるには，セミコロン区切りの値の一覧にノード名を含めます。

  例えば，test_node_name;node_name_2 です。

Use of Microsoft Excel

カンマ区切りファイルを作成する場合，Microsoft Excel が手ごろなツールですが，CSV ファイルではカラム幅，コメントなどが維持されません。nnmloadnodegroups.ovpl 入力ファイルをネイティブの XLS フォーマットとして保存後に，"名前を付けて保存(A)..." を実行して CSV ファイルを作成するのが賢明です。この場合，XLS フォーマットのファイルに Excel コメントを追加し，カラム幅を拡大することが可能であり，カンマ文字をエスケープする必要もありません。また，Microsoft Excel では，子ノードグループのリストを作成することが容易です。Column 4(D)に次のような計算値を指定するだけです。

```
=$A1&”:0;”&$A2&”:0;”&$A3&”:0;”&$A4&”:0;”&$A5&”:0;”&$A6&”:0;”&$A7&”:0;”
```

この例では最初の 7 行分の最初のカラムで定義されたノードグループを現在の行の最初のセルで定義されたノードグループ名を持つノードグループの子として統合します。この Excel 参照を利用すると，最初のカラムで子ノードグループをリネームしても，元に返って親ノードグループのカラムにある参照を変更す

る必要がありません。Microsoft Excel で先頭に # がある行でカンマを入力すると，XLS ファイルを CSV ファイルとして保存したときに，コメントなしのエントリができます（#で始まる名前を持つノードグループが作成されます）。

EXAMPLES

CSV ファイルの内容例は次のようになります。

```
SNMP,Nodes that support SNMP and that are present in
Colorado,,,,server1.myco.com;server2.myco.com,*.hp.com
```

> **メモ**
>
> CSV ファイルにデータを入力する場合，その他の目的で区切り文字「:」と「;」を使用しないでください（例えば，子ノードグループの名前など）。区切り文字を使用したい場合は，「¥」でエスケープしてください。 例えば：
>
> - "computer:1" は "computer¥:1" のように入力します。
> - "computer;1" は "computer¥;1" のように入力します。
> - "computer¥:1" は "computer¥¥:1" のように入力します。

CSV ファイルのカラム 1 に定義した Name に一致する既存のノードグループを上書きしないように，CSV ファイルからノードグループを読み込むには：

```
nnmloadnodegroups.ovpl -u system -p myadminpasswd -f /tmp/test.csv
```

CSV ファイルのカラム 1 に定義した Name に一致する既存のノードグループを上書きして，CSV ファイルからノードグループを読み込むには：

```
nnmloadnodegroups.ovpl -u system -p myadminpasswd -r true -f /tmp/test.csv
```

AUTHOR

`nnmloadnodegroups.ovpl` was developed by Micro Focus.

FILES

- Windows：`%NNM_BIN%¥nnmloadnodegroups.ovpl`
- Linux：`$NNM_BIN/nnmloadnodegroups.ovpl`

SEE ALSO

nnmconfigimport.ovpl, nnmloadinterfacegroups.ovpl, nnm.properties.

付録 G.34　nnmloadseeds.ovpl

検出ノードのシード情報を読み込みます。

SYNOPSIS

nnmloadseeds.ovpl -f *seedFile* [-t *tenant*] | -n *seeds* [-t *tenant*] -list [-format LIST|TEXT|CSV|XML] [-fields name,tenant,results,modified,notes] [-u *<username>* -p *<password>*]

DESCRIPTION

nnmloadseeds.ovpl コマンドを使用すると，検出シードをコマンドライン引数の指定（-n オプション）またはテキストファイルから（-f オプション）読み込むことができます。また，一覧表示（-list オプション）することができます。シードとは，NNMi にスパイラル検出プロセスの開始点として使用させるデバイスのことです。シードの値は IP アドレスまたはホスト名です。-n オプションを使用する場合，シードはスペース区切りでコマンドラインに指定します。シードは，SNMP をサポートしていなくても常に NNMi に追加されます。

-f オプションは，1 行に 1 エントリを指定したファイルを受け付けます。各行の形式を次に示します。

IPAddress|*HostName*,"*任意で，テナント名または UUID*"#（任意で，ノードの識別を容易にするためのコメントを指定可能）

指定内容の説明

IPAddress ＝ 追加するノードの IP アドレス。

HostName ＝ 追加するノードのホスト名。

テナントは，任意でテナント名またはテナント UUID を使用して指定できます。テナントの指定は，引用符で囲んで行う必要があります。シードから検出されたノードは，指定したテナントに割り当てられます。テナントを指定しない場合，ノードはデフォルトのテナントに割り当てられます。

#を区切り文字として複数のコメントを指定できます。また，INCLUDE-FILE<ファイル名>を使用してほかのシードファイルを含めることもできます。

-t オプションを指定すると，指定されたテナントは，-n オプションで渡されたすべてのノード，または-f オプションで指定されたシードファイルのすべてのノードに使用されます。-t オプションと-f オプションを共に使用する場合，シードファイルにおいて，-t オプションで指定したテナントとは異なるテナントが指定されたシードは，すべて無効なシードとして拒否されます。

このコマンドを実行する前に，読み込み対象のデバイスの SNMP 構成をセットアップする必要があります。

Parameters

nnmloadseeds.ovpl コマンドは，次のパラメータおよびオプションをサポートします。

-f *seedFile*

シードの読み込み元テキストファイルを指定します。

注：シードファイルのディレクトリとファイル名は，Windows システムの場合は管理者以外，Linux システムの場合は root 以外のユーザーがアクセスできる必要があります。

-n *seeds*

コマンドラインから直接読み込むシードを指定します。複数のシードを指定する場合は，スペースで区切ります。

-list

NNMi が検出のためにロードしたすべてのシードを一覧表示します。ユーティリティは，ほかの NNMi のツール用に定義された次の標準フォーマットを使用しています。TEXT,CSV,XML,LIST。シードデータのフィールドは，次の値を指定して限定できます: name, tenant, results, modified, notes。これらは，NNMi UI に表示されるカラムヘッダーの別称です。

-t *tenant name or UUID*

読み込むすべてのシードに使用されるテナントを指定します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

シードとしてのデバイスの一覧を読み込みます。

```
nnmloadseeds.ovpl -u username -p password -n mimcisco8540 15.2.112.22
```

完全修飾名を指定してノードからシードを読み込みます。

```
nnmloadseeds.ovpl -u username -p password -n mimcisco8540.superpoller3.mim
```

seeds_to_load.txt ファイルからシードを読み込みます。

```
nnmloadseeds.ovpl -u username -p password -f /tmp/seeds_to_load.txt
```

完全修飾名および特定のテナント割り当てを指定して，ノードからシードを読み込みます。

```
nnmloadseeds.ovpl -u username -p password -n mimcisco8540.superpoller3.mim -t Customer1
```

seeds_to_load.txt ファイルからシードを読み込み，特定のテナントにすべてのシードを割り当てます。

```
nnmloadseeds.ovpl -u username -p password -f /tmp/seeds_to_load.txt -t Customer2
```

ロードされたすべてのシードを CSV 形式で一覧表示します。一覧には，名前と検出シードの結果が表示されます。

```
nnmloadseeds.ovpl -u username -p password -list -format CSV -fields name,results
```

AUTHOR

nnmloadseeds.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmloadseeds.ovpl
- Linux：$NNM_BIN/nnmloadseeds.ovpl

SEE ALSO

nnmseeddelete.ovpl, nnmnodedelete.ovpl, nnmnoderediscover.ovpl, nnm.properties.

付録 G.35　nnmmanagementmode.ovpl

1 つまたは複数のノードやインターフェイス，IP アドレスの NNMi 管理モードを変更します。

SYNOPSIS

nnmmanagementmode.ovpl -node.name <*nodename*> -mode <*mode*> [-u <*username*> -p <*password*>]

nnmmanagementmode.ovpl -t <*object type*> -f <*csv_filename*> [-b <*batch size*>] [-u <*username*> -p <*password*>]

nnmmanagementmode.ovpl -t <*object type*> -s <*csv_line*> [-u <*username*> -p <*password*>]

nnmmanagementmode.ovpl -t <*object type*> -f <*uuid_file*> [-b <*batch size*>] -mode <*MODE_SPEC*> [-format <*format*>] [-fields <*fields*>] [-u <*username*> -p <*password*>]

nnmmanagementmode.ovpl -h | -help

DESCRIPTION

nnmmanagementmode.ovpl を使用すると，システム管理者は NNMi データベース内のノードやインターフェイス，IP アドレスの管理モードを設定できます。

Parameters

nnmmanagementmode.ovpl コマンドは次のオプションをサポートします。サポートされないオプションはヘルプメッセージで通知されます。

-h | -help

コマンドの使用方法を表示します。

-node.name *<nodename>*

管理モードをオンにするノードの名前を指定します。

-mode *<mode>*

設定する管理モードを指定します。有効な値は，"MANAGED"，"NOTMANAGED"または"OUTOFSERVICE"です。インターフェイスと IP アドレスについては，NNMi が"MANAGED"を"INHERITED"と読み変えます。

-mode *<MODE_SPEC>*

入力 uuid ファイルを使用して一括変更を実行する場合は，モードの引数として 2 種類のスタイルが使用できます。1 つ目は，値が MANAGED，NOTMANAGED または OUTOFSERVICE（インターフェイスと IP アドレスについては NNMi が MANAGED を INHERITED と読み変えます）のシンプルモードです。2 つ目のスタイルは現在および最新のモードで，形式は *CURRENT:NEW* となります。*CURRENT* と *NEW* には MANAGED，NOTMANAGED または OUTOFSERVICE と同じ値が入ります。2 つ目のスタイルでは，ツールは *CURRENT* と同じ現在値を持つオブジェクトだけを更新し，異なる値を持つオブジェクトはスキップします。

-t *<object type>*

管理モードを設定するオブジェクトタイプを指定します。指定できる値は，"node"，"interface"または"ipaddress"です。

-f *<csv_filename>*

管理モードを設定する CSV ファイル名（パス名つきで，例えば /opt/tmp/mynodes.csv）を指定します。ファイルに不正な定義（カラムが多い/少ない，モードが不正など）がある場合，コマンドはその定義を出力し，すべての定義が実行されません。

-b *<batch_size>*

ファイルを指定した場合，コマンドはすべてのエントリを処理し，このオプションに指定された単位でサーバーに要求を送ります。 デフォルトは 1000 エントリです。

-s *<csv_line>*

CSV 形式の 1 行を指定します。単純な変更に対してファイル作成を省略できます。

-format (csv|list|text|xml)

表形式のデータが存在しているときに出力形式を変更します

-fields *<comma_separated_fields>*

表形式のデータが存在しているときに出力フィールドを選択します。

-u *<username>*

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p *<password>*

コマンドの実行に必要な NNMi の管理者パスワードを指定します。

nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

Batch changing management modes

管理モードの一括変更を実行するために，次のコマンドを使用できます。 このコマンドへの入力は，更新するオブジェクトの UUID を含むファイルから行い，モードはコマンドラインから指定します。ファイルに記載されたすべてのオブジェクトが，指定されたモードに従って処理されます。

nnmmanagementmode.ovpl -t <*type*> -f <*filepath*> -mode <*MODE_SPEC*>

このモードでは，コマンドの出力は入力した UUID と，そのオブジェクトの結果コードのテーブルで構成されます。結果コードは，次のとおりです。

- UPDATED - オブジェクトは，新しいモードに更新されました。
- NOT_FOUND - 指定された UUID のオブジェクトがシステム内に見つかりませんでした。
- SKIPPED - オブジェクトは，現在のモードフィルターと一致しませんでした。
- NO_CHANGE - オブジェクトは，既に要求されたモードを持っています。
- ACCESS_DENIED - ユーザーは，オブジェクトの管理モードを変更する権限がありません。

例えば，neighbors.txt ファイルに，1 行に 1 つずつ UUID が記載された，インタフェースの UUID のリストが含まれている場合があります。次のコマンドを実行すると，現在 MANAGED であるインタフェースに限り，それぞれのインタフェースが OUTOFSERVICE モードに変更されます。NOTMANAGED モードのインタフェースは，変更されないままです。

```
nnmmanagementmode.ovpl -t interface -f neighbors.txt -mode MANAGED:OUTOFSERVICE

+--------------------------------------+---------+
| UUID                                 | 状態    |
+--------------------------------------+---------+
| 6f988a0f-c759-42d0-8e99-f3adce84fdd7 | UPDATED |
| 4dd87c33-9d0d-4713-b484-68c964ede6a6 | UPDATED |
| bc0af80f-d729-40d6-bb2f-ad8bd68e2deb | SKIPPED |
| bed3fff5-2385-4b09-9e3a-2572caaa624d | UPDATED |
| bd0723c0-6c53-47b1-ac13-41ecab16994e | UPDATED |
+--------------------------------------+---------+
```

ノードの管理モードに対する CSV ファイルの構文

ノードの管理モードを設定するときに指定する CSV ファイルは次の構文です。

空の行は無視されます。

"#"で始まる行は無視されます。

すべてのカラムが必要です。

- Column 1(A) : Node DNS|IP Address
  ノードの DNS 名か，IP アドレスを指定します。検索アルゴリズムは次のとおりです。
  - 一致する管理 IP アドレスを持つノードを検索します。
  - デフォルトのドメインで一致する IP アドレスを持つノードを検索します。
  - 一致するホスト名を持つノードを検索します。
  - 一致する名前を持つノードを検索します。
- Column 2(B) : Management mode
  ノードに設定するモードを指定します。有効な値は，"MANAGED", "NOTMANAGED", または "OUTOFSERVICE" です。

例）

192.168.1.1,OUTOFSERVICE

my.fqdn.com, MANAGED

インタフェースの管理モードに対する CSV ファイルの構文

インタフェースの管理モードを設定するときに指定する CSV ファイルは次の構文です。

空の行は無視されます。

"#"で始まる行は無視されます。

すべてのカラムが必要です。

- Column 1(A) : Node DNS|IP Address
  ノードの DNS 名か，IP アドレスを指定します。検索アルゴリズムは次のとおりです。
  - 一致する管理 IP アドレスを持つノードを検索します。
  - デフォルトのドメインで一致する IP アドレスを持つノードを検索します。
  - 一致するホスト名を持つノードを検索します。
  - 一致する名前を持つノードを検索します。
- Column 2(B) : Interface id
  前のフィールドで指定したノードのインタフェースの識別子を指定します。検索アルゴリズムは次のとおりです。
  - ifIndex
  - ifName
  - ifAlias
  - ifDescription
- Column 3(C) : Mode

インタフェースに設定するモードを指定します。有効な値は，"INHERITED", "NOTMANAGED", "OUTOFSERVICE" です。

例）

192.168.1.1,1,OUTOFSERVICE

my.fqdn.com, myAlias, INHERITED

IP アドレスの管理モードに対する CSV ファイルの構文

IP アドレスの管理モードを設定するときに指定する CSV ファイルは次の構文です。

空の行は無視されます。

"#" で始まる行は無視されます。

すべてのカラムが必要です。

- Column 1(A) : IP Address
  IP アドレスを指定します。同じ値の IP アドレスが複数ある場合は，次のカラムにテナント名を指定して区別します。
- Column 2(B) : Tenant name
  IP アドレスのホスト元ノードが属するテナントの名前。この値が空の場合は，すべてのテナントから IP アドレスのオブジェクトが検索されます。
- Column 3(C) : Mode
  IP アドレスに設定するモードを指定します。有効な値は，"INHERITED", "NOTMANAGED", "OUTOFSERVICE" です。

例）

192.168.1.1, my tenant, OUTOFSERVICE

192.168.1.1, ,INHERITED

RETURN VALUE

エラーが発生しなかった場合，`nnmmanagementmode.ovpl` は`0`（ゼロ）を返します。それ以外の場合，1 を返します。

AUTHOR

`nnmmanagementmode.ovpl` was developed by Micro Focus.

付録 G.36　nnmmonconfig.ovpl

SYNOPSIS

nnmmonconfig.ovpl

nnmmonconfig.ovpl listGlobalSettings

nnmmonconfig.ovpl updateGlobalSettings [-statePolling <*true*|*false*>]

nnmmonconfig.ovpl listEffective -node <*node*|*uuid*> [-interface <*interface*> | -interfaces]

DESCRIPTION

nnmmonconfig.ovpl スクリプトは，State Poller 設定を一覧表示，更新するオプションとパラメーターを提供します。nnmmonconfig.ovpl コマンドラインの一般的なフォーマットは次のとおりです。

nnmmonconfig.ovpl <*command*> <*options*>

下記の Commands 項には，使用可能なコマンドの選択肢が一覧表示されています。同様に Options 項には，各コマンドで使用可能なオプションが一覧表示されています。多くのコマンドで類似のオプションを使用できます。各コマンドで使用できる正しいオプションについては，上記の Synopsis 項を参照してください。

Commands

listGlobalSettings

グローバルな監視設定の一覧を表示します。

updateGlobalSettings

グローバルな監視設定のフィールドを更新します。

listEffective

ノードやインターフェイスに設定された監視設定の一覧を表示します。

Options

-statePolling <*true*/*false*>

状態ポーリングを有効または無効にします。

-node <*node*/*uuid*>

監視設定を表示するノードを指定します。*node* にはノードの名前またはホスト名を指定します。

-interface <*interface*>

監視設定を表示するインターフェイスを指定します。*interface* にはインターフェイスの名前を指定します。

-interfaces

指定されたノードのすべてのインタフェースを一覧表示します。

Additional Parameters

-fields <*fields*>

表形式のデータが存在しているときに出力フィールドを選択します。

-format <*style*>

表形式のデータが存在しているときに出力形式を変更します。指定できる値は，"TEXT", "LIST", "CSV", または "XML" です。

-http.host <*host*>

サーバーのホスト。デフォルトは localhost です。

-http.port <*port*>

サーバーのポート。デフォルトは 80 です。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。

-quiet

通常の出力を抑制し，エラーだけを表示します。

付録 G.37　nnmnodedelete.ovpl

NNMi トポロジデータベースからノードと関連データを削除します。

SYNOPSIS

nnmnodedelete.ovpl -help | -node <*hostName*> | -rm <*Regional NNMi management server*> | -file <*filename*> | -all [-u <*username*> -p <*password*>]

DESCRIPTION

nnmnodedelete.ovpl はシステムからノードと関連データ（インタフェース，IP アドレスなど）を削除します。この処理の結果，VLAN やサブネットが空になると，これらも同様に削除されます。インシデントがこのノードを指している場合は，ソースノードフィールドが空白になりますが，インシデントは削除されません。ノードはホスト名フィールドを使用して特定されます。

-rm オプションはリージョナル NNMi サーバーの名前を指定します。リージョナルマネージャに管理されているノードは，ローカルのデータベースから削除されます。

-file オプションでは，1 行ごとに単一エントリがあるファイルを受け入れます。各行のフォーマットは次のとおりです。

HostName #（ノードを識別するために役に立つ任意のコメント，必要な場合）

ここで，*HostName* は追加するノードのホスト名です。コメントは#文字の後に追加します。

Parameters

nnmnodedelete.ovpl コマンドは，次のオプションをサポートします。

-node <*hostName*>

削除するノードのホスト名を指定します。

-rm <*Regional NNMi management server*>

リージョナル NNMi サーバーの名前を指定します。

-file <*fileName*>

ノードを読み込む元のテキストファイルを指定します。

-all

すべてのノードを削除します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-help

コマンドの使用方法を表示します。

EXAMPLES

nnmnodedelete.ovpl -u username -p password -node myNode

ノードmyNode を削除します。（NNMi ユーザー名とパスワードを指定する必要があります。）

nnmnodedelete.ovpl -u username -p password -rm myRegionalManager

myRegionalManager に関連するすべてのノードを削除します。（NNMi ユーザー名とパスワードを指定する必要があります。）

`nnmnodedelete.ovpl -u username -p password -file myFile`

myFile ファイルで指定したノードを読み込み，データベースからノードを削除することを試みます。（NNMi ユーザー名とパスワードを指定する必要があります。）

Diagnostics

nnmnodedelete.ovpl コマンドは，次の終了コードを返します。

0

処理は成功しました。

1

エラーが発生しました。詳細はエラーメッセージを参照してください。

2

一部成功しましたが，削除されなかったノードがあります。詳細はエラーメッセージを参照してください。

AUTHOR

nnmnodedelete.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmnodedelete.ovpl
- Linux：$NNM_BIN/nnmnodedelete.ovpl

NOTES

削除したノードが自動検出ルールによって作成されたノードだった場合，そのノードは再検出されることがあります。これを回避するには，［検出の設定］フォームの［除外対象 IP アドレス］タブに対象ノードの IP アドレスを追加します。

SEE ALSO

nnmseeddelete.ovpl, nnmnoderediscover.ovpl, nnmtopodump.ovpl, nnmresetembdb.ovpl, nnm.properties.

付録 G.38　nnmnodegroup.ovpl

ノードグループおよびノードグループ階層を管理します。

SYNOPSIS

nnmnodegroup.ovpl -h | -help

nnmnodegroup.ovpl -add -group (<*name*>|<*uuid*>) [-file <*file*>|-node <*nodeList*>] [-deviceType <*deviceType*>] [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -create -name <*name*> [-file <*file*>|-node <*nodeList*>] [-filter <*filter*>] [-notes <*notes*>] [-parent <*parent*> [-expand (true|false)]] [-uuid <*uuid*>] [-addToViewFilterList (true|false)] [-calculateStatus (true|false)] [-deviceType <*deviceType*>] [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -delete -group (<*name*>|<*uuid*>) [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -list [-fields <*fields*>] [-format <*format*>] [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -listChildGroups -group <*group*> [-fields <*fields*>] [-format <*format*>] [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -listMembers -group (<*name*>|<*uuid*>) [-deep] [-fields <*fields*>] [-format <*format*>] [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -listParentGroups -group <*group*> [-fields <*fields*>] [-format <*format*>] [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -printNodes <*group name*> [-hostName | -shortName | -uuid | -ip] [-u <*username*> -p <*password*>]

nnmnodegroup.ovpl -relate -child <*child*> -parent <*parent*> [-expand (true|false)] [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -reload [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -remove -group (<*name*>|<*uuid*>) [-file <*file*>|-node <*nodeList*>] [-deviceType <*deviceType*>] [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -unrelate -child <*child*> -parent <*parent*> [-u <*username*> -p <*password*>] [-quiet]

nnmnodegroup.ovpl -update -group (<*name*>|<*uuid*>) [-addToViewFilterList (true|false)] [-calculateStatus (true|false)] [-deviceType <*deviceType*>] [-filter <*filter*>] [-name <*newName*>] [-notes <*notes*>] [-u <*username*> -p <*password*>] [-quiet]

DESCRIPTION

ノードグループおよびノードグループの階層を管理するために，nnmnodegroup.ovpl コマンドを使用します。このコマンドによって，ノードグループの作成，更新，および削除や，ノードグループの階層の作成

および削除することができます。また，ノードの追加またはノードグループフィルタの設定により，ノードグループに関連付けられているノードを管理することができます。

nnmnodegroup.ovpl コマンドは，次を含む，複数の一覧表示の機能を提供します。

- 子ノードグループまたは親ノードグループのどちらかを一覧表示することによって，ノードグループの情報と階層を一覧表示します。
- ノードグループ内のノードを一覧表示するには，-listMembers オプションを使用します。
- データベース内のすべてのノードグループを一覧表示するには，-list オプションを使用します。

-printNodes オプションは，特定のノードグループに属するノード属性を出力します。ノード属性を出力する場合，グループ名は必須の引数です。ほかの引数が与えられていない場合，各ノードのホスト名，短縮名，UUID，および管理 IP アドレス属性がダンプされます。ノードごとに一つのカンマで行が区切られます。-printNodes オプションは，以前のバージョンの NNMi で提供されていた，非推奨/従来のコマンドであることに注意してください。この使用法は，後方互換性のために残してあります。このため，行頭にダッシュ記号を指定する必要があります。すなわち，-printNodes は動作しますが，printNodes は動作しません。新しいlistMembers オプションは printNodes に置き換わるものです。

Parameters

nnmnodegroup.ovpl コマンドは，次のオプションをサポートします。

-add -group (<*name*>|<*uuid*>) [-file <*file*>|-node <*nodeList*>] [-deviceType <*deviceType*>]

ノードグループにノードまたはデバイスタイプのフィルターを追加します。-group に加えて，少なくとも一つの追加の引数を指定する必要があります。

-group (<*name*>|<*uuid*>)

ノードグループの名前または UUID です。

-file <*file*>

ノード名，UUID，および IP アドレスを含むテキストファイルです。

-node <*nodeList*>

ノード名，UUID，および IP アドレスのカンマ区切りリストです。

-deviceType <*deviceType*>

ノードグループのノードを選択するために使用します。 デバイスタイプのパラメータは，文字列をコロンで区切ったリストです。各カテゴリの形式は次のとおりです。

category:vendor:family:profile

-create -name <*node group name*> [-addToViewFilterList (true|false)] [-calculateStatus (true|false)] [-deviceType <*deviceType*>] [-expand (true|false)] [-file <*file*>|-node <*nodeList*>] [-filter <*filter*>] [-notes <*notes*>] [-parent <*parent*>] [-uuid <*uuid*>]

新しいノードグループを作成します。

-name <*name*>

作成されるノードグループの名前です。

-addToViewFilterList (true|false)

テーブルを表示するときに，ドロップダウンフィルターリストにノードグループが含まれます。

-calculateStatus (true|false)

ノードグループのステータスを計算します。

-deviceType <*deviceType*>

ノードグループのノードを選択するために使用します。 デバイスタイプのパラメータは，文字列をコロンで区切ったリストです。各カテゴリの形式は次のとおりです。

category:vendor:family:profile

-expand (true|false)

親マップでノードグループを展開します。-parent オプションを指定する必要があります。

-file <*file*>

ノード名，UUID，および IP アドレスを含むテキストファイルです。

-filter <*filter*>

ノードグループ内のノードを選択するために使用されるフィルター式です。属性が式に一致するノードがグループに含められます。

-node <*nodeList*>

ノード名，UUID，および IP アドレスのカンマ区切りリストです。

-notes <*notes*>

ノードグループに添付する注記です。

-parent <*parent*>

追加しようとしているノードグループが，子ノードグループとして追加される場合の，親ノードグループの名前です。

-uuid <*uuid*>

ノードグループに割り当てる UUID です。

-delete -group (<*name*>|<*uuid*>)

ノードグループを削除します。

-group (<*name*>|<*uuid*>)

削除するノードグループの名前または UUID です。

-list [-fields <*fields*>] [-format <*format*>]

データベース内のノードグループの名前を出力します。

-fields <*fields*>

表データを出力するフィールドを選択します。

-format <*format*>

テーブルデータの出力形式を設定します；有効な値はTEXT，LIST，CSV またはXML です。

-listChildGroups -group <*group*> [-fields <*fields*>] [-format <*format*>]

指定されたノードグループの子ノードグループを一覧表示します。

-group <*group*>

子ノードグループを一覧表示するためのノードグループ名です。

-fields <*fields*>

表データを出力するフィールドを選択します。

-format <*format*>

テーブルデータの出力形式を設定します；有効な値はTEXT，LIST，CSV またはXML です。

-listMembers -group (<*name*>|<*uuid*>) [-deep] [-fields <*fields*>] [-format <*format*>]

指定されたノードグループのノードを出力します

-group (<*name*>|<*uuid*>)

ノードグループの名前または UUID です。

-deep

デフォルトでは，listMembers はノードグループのノードだけを出力します。-deep オプションは，ノードグループのすべてのノードとすべての子ノードグループのノードが一覧表示されます。

-fields <*fields*>

表データを出力するフィールドを選択します。

-format <*format*>

テーブルデータの出力形式を設定します；有効な値は TEXT，LIST，CSV またはXML です。

-listParentGroups -group <*group*> [-fields <*fields*>] [-format <*format*>]

指定されたノードグループの親ノードグループを一覧表示します。

-group <*group*>

親グループを一覧表示するためのノードグループ名です。

-fields <*fields*>

表データを出力するフィールドを選択します。

-format <*format*>

テーブルデータの出力形式を設定します；有効な値はTEXT，LIST，CSV またはXML です。

-printNodes <*node group name*> [-hostName | -shortName | -uuid | -ip]

指定したノードグループに属するノードの属性を出力します。追加のパラメータが指定されていない場合，各ノードのホスト名，短縮名，UUID，および管理 IP アドレス属性が出力されます。ノードごとに一つのカンマで行が区切られます。

必要に応じて，次のパラメータの一つを指定することができます。指定した属性だけが出力されます。

-hostName

ノードグループに属する各ノードのホスト名を出力します。

-shortName

ノードグループに属する各ノードの短縮名を出力します。

-uuid

ノードグループに属する各ノードの UUID を出力します。

-ip

ノードグループに属する各ノードの管理 IP アドレスを出力します。管理 IP アドレスが決定できない場合は，null が出力されます。

-relate -child <*child*> -parent <*parent*> [-expand (true|false)]

親と子の関係を作成するために，別のノードグループにノードグループをリンクします。

-child <*child*>

子ノードグループの名前です。

-parent <*parent*>

親ノードグループの名前です。

-expand (true|false)

親マップでノードグループを展開します。

-reload

ノードグループのキャッシュを再ロードします。追加ノードのリストまたはノードグループのフィルターを更新することによって，ノードを追加または削除した場合，数分間はキャッシュに変更が反映されないことがあるため，この操作が必要になることがあります。

-remove -group (<*name*>|<*uuid*>) [-file <*file*>|-node <*nodeList*>] [-deviceType <*deviceType*>]

ノードグループからノードを削除します。-group に加えて，少なくとも一つの追加の引数を指定する必要があります。

-group (<*name*>|<*uuid*>)

ノードグループの名前または UUID です。

-file <*file*>

ノード名，UUID，および IP アドレスを含むテキストファイルです。

-node <*nodeList*>

ノード名，UUID，および IP アドレスのカンマ区切りリストです。

-deviceType <*deviceType*>

ノードグループのノードを選択するために使用します。 デバイスタイプのパラメータは，文字列をコロンで区切ったリストです。各カテゴリの形式は次のとおりです。

category:vendor:family:profile

-unrelate -child <*child*> -parent <*parent*>

子と親のノードグループ間のノードグループ階層を削除します。

-child <*child*>

子ノードグループの名前です。

-parent <*parent*>

親ノードグループの名前です。

-update -group (<*name*>|<*uuid*>) [-addToViewFilterList (true|false)] [-calculateStatus (true|false)] [-deviceType <*deviceType*>] [-filter <*filter*>] [-name <*newName*>] [-notes <*notes*>]

ノードグループの属性を設定します。-group に加えて，少なくとも一つの追加の引数を指定する必要があります。

-group (<*name*>|<*uuid*>)

ノードグループの名前または UUID です。

-addToViewFilterList (true|false)

テーブルを表示するときに，ドロップダウンフィルターリストにノードグループが含まれます。

-calculateStatus (true|false)

ノードグループのステータスを計算します。

-deviceType <*deviceType*>

ノードグループのノードを選択するために使用します。 デバイスタイプのパラメータは，文字列をコロンで区切ったリストです。各カテゴリの形式は次のとおりです。

category:vendor:family:profile

-filter <*filter*>

ノードグループ内のノードを選択するために使用されるフィルター式です。属性が式に一致するノードがグループに含められます

-name <*newName*>

ノードグループの名前を，指定した newName に変更します。

-notes <*notes*>

ノードグループに添付する注記です。

-h | -help

コマンドの使用方法を表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-quiet

通常の出力を抑制し，エラーだけを表示します。

EXAMPLES

"mygroup"という名前のノードグループを作成する場合：

```
nnmnodegroup.ovpl -create -name mygroup
```

ノード"a"と"b"を含む"mygroups2"という名前のノードグループを作成する場合：

```
nnmnodegroup.ovpl -create -name mygroup2 -node ”a,b”
```

親として"mygroup"，子として"mygroup2"を持つノードグループ階層を作成する場合：

```
nnmnodegroup.ovpl -relate -parent mygroup -child mygroup2
```

ノードグループ"ルーター"に属するすべてのノードのホスト名，短縮名，UUID，および管理 IP アドレスを出力する場合：

```
nnmnodegroup.ovpl -printNodes ルーター
```

ノードグループ"Non-SNMP Devices"に属するすべてのノードのホスト名だけを出力する場合：

```
nnmnodegroup.ovpl -printNodes ”Non-SNMP Devices” -hostName
```

データベース内のノードグループ名を一覧表示する場合：

```
nnmnodegroup.ovpl -list
```

既存のノードグループの名前を変更し，ステータスの計算を false に変更する場合：

```
nnmnodegroup.ovpl -update -group myGroup -name newName -calculateStatus false
```

RETURN VALUE

エラーが発生しなかった場合はステータス0（ゼロ），それ以外の場合は1 で終了します。

AUTHOR

nnmnodegroup.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmnodegroup.ovpl
- Linux：$NNM_BIN/nnmnodegroup.ovpl

付録 G.39　nnmnodegroupmapsettings.ovpl

ノードグループマップの設定およびノードグループマップのレイアウトを新規作成，更新，変更，および削除するために使用するコマンドラインツールです。

SYNOPSIS

nnmnodegroupmapsettings.ovpl -h | -help

nnmnodegroupmapsettings.ovpl create -nodeGroup <*name*/*uuid*> [-bgImage <*string*>] [-bgScale <*float*>] [-connThresh <*number*>] [-connType <*string*>] [-ifaceFilter <*string*>] [-isConnNodeGroups (true|false)] [-isConnNodes (true|false)] [-isL2Conn (true|false)] [-isL2ConnEdit (true|false)] [-isNeighborConn (true|false)] [-isShowIncidents (true|false)] [-mapRefresh <*interval*>] [-maxEndpoints <*number*>] [-maxNodes <*number*>] [-neighborConnFilter <*string*>] [-order <*number*>] [-role <*string*>]

nnmnodegroupmapsettings.ovpl list (-nodeGroup <*name*/*uuid*>|-uuid <*uuid*>)

nnmnodegroupmapsettings.ovpl update (-nodeGroup <*name*/*uuid*>|-uuid <*uuid*>) [-bgImage <*string*>] [-bgScale <*float*>] [-connThresh <*number*>] [-connType <*string*>] [-ifaceFilter <*string*>] [-isConnNodeGroups (true|false)] [-isConnNodes (true|false)] [-isL2Conn (true|false)] [-isL2ConnEdit (true|false)] [-isNeighborConn (true|false)] [-isShowIncidents (true|false)] [-mapRefresh <*interval*>] [-maxEndpoints <*number*>] [-maxNodes <*number*>] [-neighborConnFilter <*string*>] [-order <*number*>] [-role <*string*>]

nnmnodegroupmapsettings.ovpl delete (-nodeGroup <*name*|*uuid*>|-uuid <*uuid*>)

nnmnodegroupmapsettings.ovpl exists (-nodeGroup <*name*|*uuid*>|-uuid <*uuid*>)

nnmnodegroupmapsettings.ovpl pin list -nodeGroup <*name*|*uuid*>

nnmnodegroupmapsettings.ovpl pin update (-nodeGroup <*name*|*uuid*> -nodeName <*name*> [-x <*value*>] [-y <*value*>]) | -f <*csv_file*>

nnmnodegroupmapsettings.ovpl annotation create -nodeGroup <*name*|*uuid*> [-text <*string*>] [-style <*string*>] [-fontFamily <*string*>] [-fgColor <*color*>] [-bgColor <*color*>] [-bgAlpha <*value*>] [-x <*value*>] [-y <*value*>] [-scale <*value*>] [-width <*value*>] [-height <*value*>] [-zIndex <*value*>]

nnmnodegroupmapsettings.ovpl annotation list -nodeGroup <*name*|*uuid*>

nnmnodegroupmapsettings.ovpl annotation update -uuid <*annotation_uuid*> [-text <*string*>] [-style <*string*>] [-fontFamily <*string*>] [-fgColor <*color*>] [-bgColor <*color*>] [-bgAlpha <*value*>] [-x <*value*>] [-y <*value*>] [-scale <*value*>] [-width <*value*>] [-height <*value*>] [-zIndex <*value*>]

nnmnodegroupmapsettings.ovpl annotation delete -uuid <*annotation_uuid*>

DESCRIPTION

nnmnodegroupmapsettings.ovpl は，ノードグループマップの設定を新規作成，更新，変更，および削除するために使用するコマンドラインツールです。ノードグループマップの設定は，ノードグループごとに個別に作成されます。ノードグループマップの設定は，ノードグループ名または UUID またはマップ設定 UUID によって更新または削除される場合があります。さらに，任意のノードグループ設定の詳細を画面に一覧表示したり，CVS や XML などの各種形式にフォーマットしたりできます。

nnmnodegroupmapsettings.ovpl は，ノードグループマップビューのノードやサブネットなどのマップオブジェクトの位置の調整やマップ注釈の編集も可能です。この機能を使うためには，ユーザーはノードグループマップの設定で指定されたマップの保存に必要なロールを満たす必要があります。

Commands

nnmnodegroupmapsettings.ovpl コマンドは，次のサブコマンドの先頭に-（ハイフン）をつけた場合も動作可能です。

create

指定したノードグループ用の新しいノードグループマップの設定を作成します。

list

指定したノードグループ名または UUID のノードグループマップの設定情報を一覧表示します。ノードグループ名または UUID を指定しない場合は，すべてのノードグループのすべての設定を一覧表示します。

update

指定されたノードグループマップの設定を更新します。ノードグループマップの設定フィールドに null を設定する場合は，値なしで引数を指定してください。例えば，

nnmnodegroupmapsettings.ovpl -update -nodeGroup ルーター -ifaceFilter

ノードグループ，接続タイプ，およびロールのフィールドは値を必要とし，null に設定することはできません。

delete

指定したノードグループマップの設定を削除します。

exists

指定したノードグループのノードグループマップ設定が存在するかどうかを確認します。ノードグループマップ設定が存在する場合は"true"を出力し，存在しない場合はその旨のメッセージを出力します。

pin list

ノードグループマップで保存されたマップオブジェクトの位置を一覧表示します。あるノードグループマップビューに新たに追加されたマップオブジェクトは，位置が保存されるまではこの一覧には表示されません。マップオブジェクトの位置を保存するには，NNMi コンソールのノードグループマップビューで「マップを保存」を使用してください。

「-format csv」オプションを使用したこのコマンドの出力は，ヘッダー行を削除することで「pin update」の入力に使用可能です。

`pin update`

ノードグループマップの指定したマップオブジェクトの位置を更新します。コマンドライン引数を使用した単一のマップオブジェクトの位置を更新と，CSV ファイルを使用した複数の位置の更新が可能です。このコマンドを使用する前に，NNMi コンソールのノードグループマップビューで「マップを保存」を使用しマップオブジェクトの位置を保存する必要があります。

このコマンドによって，NNMi コンソールでは編集できない位置にマップオブジェクトを移動することが可能になる場合があります。そのようになった場合は，このコマンドを用いて有効な位置にマップオブジェクトを移動するか，NNMi コンソールのノードグループマップビューで「レイアウトのクリア」を使用してください。

`annotation create`

指定したノードグループマップにマップ注釈を作成します。

`annotation list`

指定したノードグループマップのマップ注釈を一覧表示します。

`annotation update`

指定したマップ注釈を更新します。

`annotation delete`

指定したマップ注釈を削除します。

Parameters

`nnmnodegroupmapsettings.ovpl` コマンドは，次のオプションをサポートします。

`-h | -help`

コマンドの使用方法を表示します。

`-bgAlpha` 〈*value*〉

マップ注釈のボックスの背景色の透明度です。有効な値の範囲は 0.0 から 1.0 です。

`-bgColor` 〈*value*〉

マップ注釈のボックスの背景色です。値の形式は"<Red>,<Green>,<Blue>"で有効な値の範囲は 0 から 255 です。例："0,128,255"

`-bgImage` 〈*string*〉

ノードグループマップに関連付けられる，背景イメージファイルの名前です。

`-bgScale` 〈*float*〉

ノードグループマップに関連付けられる，背景イメージファイルのスケールです。浮動小数点の値として解釈されます。デフォルト値は 1.0 です。

`-connThresh` 〈*number*〉

複数接続しきい値です。接続を多重接続に折りたたまないで表示する，ノード間またはノードグループ間の接続の最大数です。

-connType <*string*>

接続タイプです。接続のタイプは，関連するノードグループマップに表示されます。有効な値は，"none"，"L2" または"L3"です。引数は大文字と小文字を区別しない方法で解釈されます。デフォルト値は"none"です。

-f <*CSV file*>

マップオブジェクトの位置の更新に使う CSV ファイルです。

CSV ファイルは以下の形式に従う必要があります。

空行は無視されます。"#"から始まる行は無視されます。全ての列は必須です。

- 列 1(A)：ノードグループの名前または UUID
- 列 2(B)：位置を更新するマップオブジェクトの名前

  このマップオブジェクトの名前は，pin list コマンドに表示される名前と同じにします。インベントリやノードグループマップビューに表示される名前とは異なる可能性があります。
- 列 3(C)：マップオブジェクトの位置の X 座標
- 列 4(D)：マップオブジェクトの位置の Y 座標

行の例：

ルーター,test-node,100,200

-fgColor <*value*>

マップ注釈の文字色です。値の形式は"<Red>,<Green>,<Blue>"で有効な値の範囲は 0 から 255 です。例:"0,128,255"

-fontFamily <*value*>

マップ注釈のテキストのフォントです。有効な値は"sans-serif"，"serif"，"monospace"または"cursive"です。

-height <*value*>

マップ注釈の領域の高さです。

-ifaceFilter <*string*>

関連するノードグループマップに適用する，終了ポイントインタフェースのフィルターの名前です。

-isConnNodeGroups (true|false)

関連するノードグループマップにノードグループを接続するかどうかを示すフラグです。ブール値として解釈します。"true"は，どのような大文字と小文字の組み合わせで設定しても true と解釈されますが，その他すべての引数の値は false と解釈されます。

-isConnNodes (true|false)

関連するノードグループマップにノードを接続するかどうかを示すフラグです。ブール値として解釈します。"true"は，どのような大文字と小文字の組み合わせで設定しても true と解釈されますが，その他すべての引数の値は false と解釈されます。

-isL2Conn (true|false)

IPv4 サブネット接続ルールによって判別されたレイヤー 2 接続を，関連するノードグループマップ上に表示するかどうかを示すフラグです。ブール値として解釈します。"true"は，どのような大文字と小文字の組み合わせで設定しても true と解釈されますが，その他すべての引数の値は false と解釈されます。

-isL2ConnEdit (true|false)

nnmconnedit.ovpl コマンドラインツールを使用して追加したレイヤー 2 接続編集を，関連するノードグループマップ上に表示するかどうかを示すフラグです。ブール値として解釈します。"true"は，どのような大文字と小文字の組み合わせで設定しても true と解釈されますが，その他すべての引数の値は false と解釈されます。

-isNeighborConn (true|false)

ノードグループのメンバーではない（1 ホップ離れた距離にある）主要なネットワークデバイスへの追加接続を行うかどうかを示すフラグです。ブール値として解釈します。"true"は，どのような大文字と小文字の組み合わせで設定しても true と解釈されますが，その他すべての引数の値は false と解釈されます。フィルターが提供されていない場合は，「ネットワーキングインフラストラクチャデバイス」ノードグループがフィルターとして使用されます。子ノードグループのデバイスフィルターがあっても無視されます。

-isShowIncidents (true|false)

ノードグループマップ上で重要なインシデントと関連するノードのマップシンボルを拡大表示するかどうかを示すフラグです。ブール値として解釈します。"true"は，どのような大文字と小文字の組み合わせで設定しても true と解釈されますが，その他すべての引数の値は false と解釈されます。

-mapRefresh <*interval*>

分と秒で指定するマップのリフレッシュ間隔です。フォーマットは mmMssS の形で，mm，ss には 0 ～59 の数値を入れます。例：59M59S(最大値)，1S(1 秒)，5M(5 分)，5M30S(5 分 30 秒)

-maxEndpoints <*number*>

関連するノードグループマップ上に表示される，エンドポイントの最大数です。

-maxNodes <*number*>

関連するノードグループマップ上に表示される，ノードの最大数です。

-neighborConnFilter <*string*>

ノードグループに所属していない，1 ホップ近隣のノードをノードグループマップに表示する場合に，フィルターの役割を果たすノードグループです。ノードグループに対して定義されたデバイスフィルターがあれば，候補となる近隣ノードに適用されます。重要な注意 - フィルターを適用するとき，子ノードグループは無視されます。isNeighborConn オプションの説明も参照してください。

-nodeGroup (<*name*>|<*uuid*>)

ノードグループの名前です。

-nodeName <*name*>

マップオブジェクトの名前です。

-order <*number*>

整数で指定された，ノードグループマップの設定の優先順序です。番号が低いほど優先度が高くなります。例えば，1 は 10 よりも高い優先順序です。

-role <*string*>

ノードグループマップ設定にアクセスするために必要なユーザーロールです。有効な値は"admin"，"client"，"level1"，"level2"です。引数は大文字と小文字を区別しない方法で解釈されます。デフォルト値は"admin"です。

-scale <*value*>

マップ注釈のテキストのスケールです。有効な値の範囲は 0.2 から 10000 です。

-style <*value*>

マップ注釈のテキストのスタイルです。有効な値は"normal"，"bold"，"italic"または"bold-italic"です。

-text <*string*>

マップ注釈のテキストです。

-uuid <*value*>

ノードグループマップ設定かマップ注釈の UUID です。

-width <*value*>

マップ注釈の領域の幅です。

-x <*value*>

マップ注釈かマップオブジェクト位置の X 座標です。

-y <*value*>

マップ注釈かマップオブジェクト位置の Y 座標です。

-zIndex <*value*>

マップ注釈の Z-index です。

Additional Parameters

-fields <*fields*>

表形式のデータが存在しているときに出力フィールドを選択します。

-format <*style*>

一覧表示の出力形式です。'text','list','csv'および'xml'が利用できます。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-quiet

通常の出力を抑制し，エラーだけを表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。 詳細は nnm.properties リファレンスページを参照してください。

FORMATTING RULES

ノードグループ名を指定するとき，スペースが存在する場合は，名前を二重引用符で囲みます。(例："Important Nodes")

EXAMPLES

必須の引数だけでノードグループマップの設定を作成します。

```
nnmnodegroupmapsettings.ovpl create -nodeGroup Routers -connType L3 -role admin
```

オプションの引数でノードグループマップの設定を作成します。

```
nnmnodegroupmapsettings.ovpl create -nodeGroup Switches -connType L2 -bgImage /images/
Colorado.png -bgScale .75 -connThresh 3 -ifaceFilter ”VLAN Interfaces” -isConnNodeGroups true
-isConnNodes t -isL2Conn true -isL2ConnEdit True -mapRefresh 5M -maxEndpoints 200 -maxNodes
100 -order 10 -role admin
```

指定されたノードグループのノードグループマップ設定を一覧表示します。属性データがテーブルに表示されます。

```
nnmnodegroupmapsettings.ovpl list -nodeGroup ”Important Nodes”
```

ノードグループマップ設定のすべてを一覧表示します。属性データがテーブルに表示されます。

```
nnmnodegroupmapsettings.ovpl list
```

指定されたノードグループのノードグループマップの設定を CSV 形式で一覧表示します。

```
nnmnodegroupmapsettings.ovpl list -nodeGroup ”Non-SNMP Devices” -format csv
```

ノードグループ名によってノードグループマップ設定を削除します。

```
nnmnodegroupmapsettings.ovpl delete -nodeGroup ”Routers”
```

設定の UUID によってノードグループマップ設定を削除します。

```
nnmnodegroupmapsettings.ovpl delete -uuid 204846c0-a35b-4a92-9726-4ce0a8be596d
```

ノードグループのノードグループマップ設定を更新します。

```
nnmnodegroupmapsettings.ovpl update -nodeGroup Routers -bgImage /images/Denver.png -
bgScale .75 -connThresh 2 -ifaceFilter ”Point to Point Interface” -isConnNodeGroups true -
```

isConnNodes t -isL2Conn false -isL2ConnEdit False -mapRefresh 4M30S -maxEndpoints 200 -maxNodes 100 -order 15 -role admin

AUTHOR

nnmnodegroupmapsettings.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmnodegroupmapsettings.ovpl
- Linux：$NNM_BIN/nnmnodegroupmapsettings.ovpl

付録 G.40　nnmnoderediscover.ovpl

ノードの再検出

SYNOPSIS

nnmnoderediscover.ovpl -help | -node <*hostName*> [-tenant <*name*>] [-fullsync] | -rm <*Regional NNMi management server*> [-fullsync] | -file <*filename*> [-tenant <*name*>] [-fullsync] | -all [-fullsync] [-u <*username*> -p <*password*>]

DESCRIPTION

nnmnoderediscover.ovpl は NNMi 検出キューにノードを挿入します。ノードが検出を始める前の時間の量は，キュー中のノードに到着するために，NNMi がどれくらいの時間がかかるかに依存します。

ノードが既に検出キューにある場合，それが再び加えられることはありません。

NNMi Advanced のグローバルネットワーク管理機能を使用している場合，リージョナルマネージャによって管理されるノードは，リージョナルマネージャで 検出され，グローバルマネージャで再検出されません。

-rm オプションは，NNMi Advanced のグローバルネットワーク管理機能で，グローバルマネージャで nnmnoderediscover.ovpl コマンドを実行する場合に使用します。NNMi は，リージョナルマネージャに対して，リージョナルマネージャからグローバルなマネージャへ最新の利用可能な検出結果を送ることを要求するリクエストを送ります。

-file オプションは，1 行に一つの項目を指定したファイルを受けつけます。それぞれの行には短い名前，完全修飾 DNS ドメインネーム，あるいは IP アドレスを含めます。それぞれの行は次のフォーマットです：HostName#（必要に応じてノードを識別するためのコメント）HostName は挿入したいノードの短い名前，完全修飾 DNS ドメインネーム，あるいは IP アドレスです。

-all オプションは，ローカルの NNMi 管理サーバーによって管理されたノードをすべて再検出します。NNMi Advanced のグローバルネットワーク管理機能を使用していて，グローバルマネージャで

nnmnoderediscover.ovpl コマンドを実行する場合，リージョナルマネージャによって管理されるノードは，リージョナルマネージャからグローバルマネージャの元へ最新の利用可能な検出結果を送ります。

-tenant オプションは，アドレスのドメインが重複している場合など，名前または IP アドレスが一意でないノードを特定します。この引数によって渡される名称は，そのノードが属しているテナント名です。

-fullsync オプションは，ノードの再検出後に，そのノードの状態およびステータスを再同期します。 グローバルマネージャから実行した場合，グローバルマネージャのノードがリージョナルマネージャ上のノード情報に基づいて更新されます。 リージョナルマネージャから実行した場合，コマンドはリージョナルのノードについて再同期を実行し，さらにそのリージョナルマネージャに属するノードについてグローバルマネージャ上で再同期を実行します。これはオプションのフラグであり，再検出する対象ノードを選択するアルゴリズムには影響を与えません。

Parameters

nnmnoderediscover.ovpl コマンドは，次のオプションをサポートします。

-node <*hostName*>

再検出するノードのホスト名を指定します。

-rm <*Regional NNMi management server*>

NNMi Advanced のグローバルネットワーク管理機能を使用していて，グローバルマネージャで nnmnoderediscover.ovpl コマンドを実行する場合，リージョナルマネージャの接続の設定の「名前」属性です。

-file <*filename*>

ノードを読み込むテキストファイルを指定します。

-all

すべてのノードを再検出するときに指定します。

-tenant <*name*>

ノードの名前または IP アドレスと組み合わせて，名前または IP アドレスが一意でない可能性のあるドメインでノードを特定するための任意のオプションです。

-fullsync

各ノードの再検出後に，ノードの状態とステータスを再同期するよう NNMi に指示する任意のオプションです。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-help

コマンドの使用方法を表示します。

EXAMPLES

nnmnoderediscover.ovpl -u username -p password -node myNode -tenant myTenant

テナント myTenant のメンバーであるノードmyNode を再検出します。有効な NNMi 管理者のユーザー名とパスワードを指定する必要があります。

nnmnoderediscover.ovpl -u username -p password -rm myRegionalManager

myRegionalManager に関連するすべてのノードを再検出します。有効な NNMi 管理者のユーザー名とパスワードを指定する必要があります。

nnmnoderediscover.ovpl -u username -p password -file myFile -fullsync

myFile ファイルに指定されたノードを読み込み，NNMi 検出キューに挿入します。各ノードの再検出後に，ノードの状態とステータスを再同期します。有効な NNMi 管理者のユーザー名とパスワードを指定する必要があります。

Diagnostics

nnmnoderediscover.ovpl コマンドは，次の終了コードを返します。

0

処理は成功しました。

1

エラーが発生しました。詳細はエラーメッセージを参照してください。

AUTHOR

nnmnoderediscover.ovpl was developed by Micro Focus.

SEE ALSO

nnmdiscocfg.ovpl, nnmloadseeds.ovpl, nnmseeddelete.ovpl, nnm.properties.

付録 G.41　nnmnotes.ovpl

1 つまたは複数のノードの注を表示または更新します。

SYNOPSIS

nnmnotes.ovpl print -type <*object type*> (-name <*name*> | -uuid <*uuid*> | -filter <*filter*> | -all) [-format <*format*>] [-fields <*fields*>]

nnmnotes.ovpl update -type <*object type*> { ((-name <*name*> | -uuid <*uuid*>) -notes <*notes*>) | -file <*file*>} [-format <*format*>] [-fields <*fields*>]

DESCRIPTION

nnmnotes.ovpl を使用すると，NNMi データベース内の 1 つまたは複数のノードの注を表示または更新できます。

Parameters

nnmnotes.ovpl print コマンドは，次のオプションをサポートします。

-h

コマンドの使用方法を表示します。

-all

引数"-type"で指定されたタイプのすべてのオブジェクトを選択します。

-filter <*filter*>

オブジェクトを選択するためのスペースを含まないフィルター文字列を指定します。フィルター文字列を含む注を持つオブジェクトが選択されます。

-name <*name*>

オブジェクトの名前を指定します。オブジェクトのタイプが"node"である場合はノードの名前を指定します。

-type <*object type*>

注を出力または更新するオブジェクトのタイプを指定します。有効な値は"node"です。

-uuid <*uuid*>

オブジェクトの UUID を指定します。

Options for print command

nnmnotes.ovpl print コマンドは，次のオプションをサポートします。

-h

コマンドの使用方法を表示します。

-all

引数"-type"で指定されたタイプのすべてのオブジェクトを選択します。

-filter <*filter*>

オブジェクトを選択するためのスペースを含まないフィルター文字列を指定します。フィルター文字列を含む注を持つオブジェクトが選択されます。

-name <*name*>

オブジェクトの名前を指定します。オブジェクトのタイプが"node"である場合はノードの名前を指定します。

-type <*object type*>

注を出力または更新するオブジェクトのタイプを指定します。有効な値は"node"です。

-uuid <*uuid*>

オブジェクトの UUID を指定します。

Options for update command

nnmnotes.ovpl update コマンドは，次のオプションをサポートします。

-h

コマンドの使用方法を表示します。

-file <*file*>

CSV ファイル名（パス名付きで，例えば/opt/tmp/mynodes.csv）を指定します。ファイルに不正な定義（カラムが多い，または少ないなど）がある場合，コマンドはその定義を出力し，すべての定義が実行されません。

-name <*name*>

オブジェクトの名前を指定します。オブジェクトのタイプが"node"である場合はノードの名前を指定します。

-notes <*notes*>

指定された文字列で注を更新します。文字列にスペースが含まれている場合は，文字列をダブルクォーテーション（"）で囲んでください。文字列が空文字（""），または文字列が指定されていない場合は，注を空文字で更新します。

-type <*object type*>

注を出力または更新するオブジェクトのタイプを指定します。有効な値は"node"です。

-uuid <*uuid*>

オブジェクトの UUID を指定します。

Additional Parameters

-fields <*comma_separated_fields*>

表形式のデータが存在しているときに出力フィールドを選択します。

-format <*style*>

表形式のデータの出力形式を設定します。有効な値は，"TEXT"，"LIST"，"CSV"，または"XML"です。

-jndiHost <*hostname*>

jndi サーバーのホスト名です。デフォルトの値は"localhost"です。

-jndiPort <*port*>

jndi サーバーのポート番号です。デフォルト値は 1099 です。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，この引数の指定が必要になります。詳細は nnm.properties リファレンスページを参照してください。

-quiet

通常の出力を抑制し，エラーだけを表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，この引数の指定が必要になります。詳細は nnm.properties リファレンスページを参照してください。

"-fields"引数がサポートする値

オブジェクトのタイプが"node"である場合，有効な値は，"name"，"uuid"，"tenant"，"notes"です。

update コマンドに対する CSV ファイルがサポートする文字コード

注を更新するときに指定する CSV ファイルの文字コードは NNMi サーバーのロケールに一致している必要があります。例えば Windows 日本語環境の場合は Shift-JIS（Windows-31J）を指定してください。

update コマンドに対する CSV ファイルの構文

注を更新するときに指定する CSV ファイルは次の構文です。

空の行は無視されます。

"#"で始まる行は無視されます。

すべてのカラムが必要です。

- Column 1(A): Object's Name|UUID
  オブジェクトの名前，または UUID を指定します。
- Column 2(B): Notes
  注を指定します。オブジェクトの注は指定された文字列に更新されます。

文字列にスペースやカンマ（,）が含まれている場合は，文字列をダブルクォーテーション（"）で囲んでください。

文字列が空文字（""），または文字列が指定されていない場合は，注を空文字で更新します。

EXAMPLES

オブジェクトの名前を指定して注を表示します。

```
nnmnotes.ovpl print -type node -name mynode
```

オブジェクトの UUID を指定して注を表示します。

```
nnmnotes.ovpl print -type node -uuid d913a5c6-e297-4243-8c18-4ea2ff61e32b
```

フィルター文字列を指定して注を表示します。

```
nnmnotes.ovpl print -type node -filter server
```

指定したタイプのすべてのオブジェクトの注を表示します。

```
nnmnotes.ovpl print -type node -all
```

オブジェクトの名前を指定して注を更新します。

```
nnmnotes.ovpl update -type node -name mynode -notes "This is test machine."
```

オブジェクトの UUID を指定して注を更新します。

```
nnmnotes.ovpl update -type node -uuid d913a5c6-e297-4243-8c18-4ea2ff61e32b -notes "This is
test machine."
```

CSV ファイルを指定して注を更新します。

```
nnmnotes.ovpl update -type node -file /opt/tmp/mynodes.csv
```

RETURN VALUE

エラーが発生しなかった場合，nnmnotes.ovpl は0（ゼロ）を返します。それ以外の場合，1 を返します。

AUTHOR

nnmnotes.ovpl was developed by Micro Focus.

付録 G.42　nnmofficialfqdn.ovpl

NNMi 管理サーバーの正式な完全修飾名（FQDN）を表示します。

SYNOPSIS

```
nnmofficialfqdn.ovpl
```

DESCRIPTION

nnmofficialfqdn.ovpl は，正式な完全修飾名（FQDN）を表示する場合に使用します。正式 FQDN は，インストール時に NNMi が設定します。インストール後は，nnmsetofficialfqdn.ovpl コマンドで変更できます。

Parameters

デフォルトでnnmofficialfqdn.ovpl は ping のテストを含めた，冗長な結果を表示します。
nnmofficialfqdn.ovpl コマンドは，次のオプションをサポートします。

-d

正式 FQDN のドメイン名を表示します。FQDN がショートホスト名または IP アドレスの場合，および-t オプション（簡易表示モード）を使用した場合，NNMi は値を表示しません。それ以外の場合，NNMi はドメイン名が見つからなかった旨のメッセージを表示します。

-t

簡易表示モード。FQDN またはドメイン名の値だけを表示します。FQDN またはドメイン名が見つからなかった場合に，警告メッセージ等のテキストは表示されません。このオプションを指定した場合は，ping テストも省略されます。

-m

デフォルト FQDN を照会し，値を表示します。NNMi は次の順序で照会を行って，最初に見つかった値を表示します。

- FQDN
- ショートホスト名
- IP アドレス

どれも見つからなかった場合は，'localhost'が表示されます。

-h

すべてのオプションを一覧表示したヘルプメニューを表示します。

EXAMPLES

オプションを何も指定せずにコマンドを実行した場合は，正式 FQDN を表示し，ping テストを行います。

```
# nnmofficialfqdn.ovpl
      FQDN:
      hostname.somedomain
      pingテスト: hostname.somedomainに対してpingを行っています。しばらくお待ちください
      pingが正常に完了しました
```

-t オプションを指定してコマンドを実行した場合は，正式 FQDN が次のように表示されます。

```
# nnmofficialfqdn.ovpl -t
      hostname.somedomain
```

-d オプションを指定してコマンドを実行した場合は，ドメイン名が次のように表示されます。

```
# nnmofficialfqdn.ovpl -d
       ドメイン:
       somedomain
```

-d オプションおよび-t オプションを指定してコマンドを実行した場合は，ドメイン名だけが表示されます。

```
# nnmofficialfqdn.ovpl -dt
      somedomain
```

AUTHOR

nnmofficialfqdn.ovpl was developed by Micro Focus.

FILES

nnmofficialfqdn.ovpl は，次のディレクトリにあります。

- Windows：%NNM_BIN%
- Linux：$NNM_BIN

SEE ALSO

nnmsetofficialfqdn.ovpl.

付録 G.43　nnmresetembdb.ovpl

NNMi が組み込みデータベースと連携するように設定されている場合は，組み込みデータベースを空に（削除）してから再作成します。

SYNOPSIS

nnmresetembdb.ovpl [-?|-h|-help] [-silent] [-nostart]

DESCRIPTION

nnmresetembdb.ovpl は，NNMi 組み込みデータベースを空に（削除）してから再作成するときに使用します。このコマンドは，組み込みデータベースオプションで NNMi をインストールした場合にだけ有用です。このコマンドは，データベースが破損した場合にデータをすべて破棄してよいとき，または単にデータベースをインストール直後の初期状態に戻すときにだけ使用してください。

このコマンドを実行するときにNNMiが起動していると，このコマンドはまず（ovstopによって）NNMiを停止してから，（ovstartによって）データベースと連携するためのnmsdbmgrプロセスを開始します。-nostartオプションが指定された場合を除いて，初期化プロセスの完了時に（ovstartによって）NNMiを再起動します。

データベースの初期化が完了すると，組み込みデータベースにはテーブルもデータもなくなります。テーブルは，ovstartコマンドでNNMiを再起動したとき，またはnnmresetembdb.ovplコマンドによってNNMiが自動的に起動したときに再作成されます。

このコマンドを実行するには，WindowsシステムのAdministrator権限またはLinuxシステムのrootとしてログインする必要があります。

Parameters

nnmresetembdb.ovplコマンドは，次のパラメータをサポートします。

-silent

このオプションを使用すると，nnmresetembdb.ovplコマンドは実行結果の表示を抑止します。

-nostart

このオプションを使用すると，nnmresetembdb.ovplコマンドはデータベースの初期化後にNNMiを起動しません。

-?|-h|-help

コマンドの使用方法を表示します。

EXAMPLES

検出性能が悪い場合，データベースが破損した場合，またはデータベース（データベースに格納された設定項目を含む）をインストール直後の初期状態にしたい場合，このコマンドを使用してデータベースを初期化できます。

次のメッセージが表示されます。

```
# nnmresetembdb.ovpl -nostart

警告: このツールを実行すると、NNMが停止し、データベースが破棄および再作成され、
          NNMが再起動されます。現在、重要なアクティビティが発生していないことを
          確認してください。


NNMを停止してもよいですか (ovstop)? [n]
y
ありがとうございます!


警告: これにより、すべての設定データと検出済みデータが削除されます。
          バックアップを取得していない限り、リセットから回復できません。
```

```
データベースをリセットしてもよいですか? [n]
y

組み込みデータベースのリセットを試みています...

NNMを停止しています...

NNMは正常に停止しました。

データベースのリセットのnmsdbmgrプロセスを開始しています...

データベースリセットのnmsdbmgrプロセスを正常に開始しました。

組み込みデータベース nnm を正常にリセットしました。

メッセージングフォルダを正常に削除しました

#
```

AUTHOR

nnmresetembdb.ovpl was developed by Micro Focus.

FILES

nnmresetembdb.ovpl は%NNM_BIN%（Windows）または$NNM_BIN（Linux）ディレクトリにあります。

SEE ALSO

nmsdbmgr, nnmnodedelete.ovpl, ovstart, ovstop, ovstatus.

付録 G.44　nnmrestore.ovpl

このコマンドは，nnmbackup.ovpl コマンドで作成されたバックアップをリストアします。

SYNOPSIS

nnmrestore.ovpl [-?|-h|-help] [-force] [-lic] [-partial] -source <*directory*>

DESCRIPTION

nnmrestore.ovpl コマンドは，NNMi に対する主要リストアコマンドして機能します。このコマンドは，バックアップファイルで保存された状態に NNMi をリストアするため，nnmbackup.ovpl コマンドで実行された以前の NNMi バックアップを使用します。リストアの範囲は，バックアップの内容と，指定されたコマンドライン引数により決まります。バックアップ内に存在しているデータのみがリストアの対象です。

nnmbackup.ovpl コマンドを使用して NNMi のバックアップを作成し，次にnnmrestore.ovpl コマンドを使用してデータベースレコードを別の NNMi 管理サーバーに配置する場合は，どちらの NNMi 管理サーバー

も同種のオペレーティングシステムと同一バージョンの NNMi がインストールされ，同一のパッチレベルが適用されている必要があります。バックアップデータをある NNMi 管理サーバーから別の NNMi 管理サーバーに配置するということは，どちらのサーバーも同一のデータベース UUID を持つことを意味します。グローバルネットワーク管理機能を使用する場合は，2 台目の NNMi 管理サーバーに NNMi をリストアした場合，元の NNMi 管理サーバーから NNMi をアンインストールしてください。

このコマンドは，次の状況を検出します。

- 対象システムでバックアップが作成されたかどうか。-lic を指定すると，同一システムでバックアップが作成された場合のみ，ライセンス情報をリストアできます。

ソースディレクトリには，指定のリストア動作に要求されるファイルのすべてが単一の tar ファイルとして指定されています。ソースが tar ファイルの場合，この tar ファイルは現在の作業ディレクトリで一時フォルダとして展開されます。この一時フォルダは，リストア完了後に削除されます。

リストア動作を完了するには，必ず NNMi を停止する必要があります。-force オプションを指定すると，このコマンドは NNMi を停止します。当初のバックアップがオンラインバックアップであったとソースフォルダに存在するファイルに表示されている場合，リストア動作はnmsdbmgr プロセスを起動し， 組み込みデータベースが利用できるかどうかを確認します。つまり，オンラインバックアップからリストアするには，-force オプションの使用が絶対条件であることを意味しています。

nnmrestore.ovpl コマンドを実行するには，Windows システムの管理者または Linux システムの root としてログオンする必要があります。

Parameters

nnmrestore.ovpl コマンドは，次のオプションをサポートします。

-force

このオプションを指定すると，このコマンドは，リストア手順が実行される前に NNMi を停止します。また，リストアが準拠しているバックアップのタイプをリストアする必要がある場合，このコマンドは nmsdbmgr プロセスを起動します。このオプションは，オンラインバックアップからリストアする場合に必要です。

-lic

このオプションを指定すると，このコマンドによりライセンス情報がリストアされます。同じシステムでバックアップが作成されたことが確認された場合にのみ，このコマンドでライセンス情報がリストアされるということに注意が必要です。

-partial

このオプションを指定しないと，データベースおよび対応する SSL 証明書がリストアされます。 システムからシステムへのリストアの場合，nnm.keystore とnnm.truststore が対象のシステムのものとマージされます。 マージはターゲットシステムに存在しない，バックアップに格納されたすべての証明書エイリアスを統合します。例外として，両方のストアに FQDN の自己証明書が存在する場合，ターゲットシステムのキーエイリアスは削除され，バックアップのものと入れ替えられます。証明書がマー

ジまたはリストアされる前に，バックアップが作成され，同じディレクトリに格納されます。名前は *.backup になります。

-partial オプションを指定すると，データベースおよび対応する SSL 証明書がリストアされません。これは，設定ファイルのみをリストアする場合に便利です。

-source <*directory*>

-source オプションの引数には，nnmbackup.ovpl コマンドで取得した，バックアップデータのディレクトリまたは tar ファイルを指定してください。ユーザーが指定するソースが tar ファイルの場合，この tar ファイルは現在の作業ディレクトリで一時フォルダとして展開されます。この一時フォルダは，リストア完了時に削除されます。

-?|-h|-help

コマンドの使用方法を表示します。

EXAMPLES

以前のバックアップをリストアするには：

```
#./nnmrestore.ovpl -source /tmp/bak/config
```

-force オプションを指定してリストアするには：

```
#./nnmrestore.ovpl -force -source /tmp/bak/all
```

-partial オプションを使用して，データベース，SSL 証明書，およびライセンスを除くすべてのバックアップデータをリストアするには：

```
#./nnmrestore.ovpl -partial -source /tmp/bak/all
```

-lic オプションを使用して，ローカルシステムのライセンス情報を含め，すべてをリストアするには：

```
#./nnmrestore.ovpl -lic -source /tmp/bak/all
```

AUTHOR

nnmrestore.ovpl was developed by Micro Focus.

SEE ALSO

nnmbackup.ovpl.

付録 G.45　nnmrestoreembdb.ovpl

このコマンドは，nnmbackupembdb.ovpl コマンドで作成されたバックアップをリストアします。

SYNOPSIS

nnmrestoreembdb.ovpl [-?|-h|-help] [-force] -source <*backup file*>

DESCRIPTION

nnmrestoreembdb.ovpl コマンドを使用すると，NNMi 組み込みデータベースの完全なバックアップを復元することができます。 復元を行うときに必要なバックアップファイルは，nnmbackupembdb.ovpl コマンドを使用して作成されます。

nnmbackupembdb.ovpl コマンドを使用して NNMi 組み込みデータベースのバックアップを作成し，次に nnmrestoreembdb.ovpl コマンドを使用して組み込みデータベースレコードを別の NNMi 管理サーバーに配置する場合は，どちらの NNMi 管理サーバーも同種のオペレーティングシステムと同一バージョンの NNMi がインストールされ，同一のパッチレベルが適用されている必要があります。

nnmrestoreembdb.ovpl コマンドを実行する前に，nnmresetembdb.ovpl コマンドを実行して組み込みデータベースを空にしておくことを推奨します。 データベースを空にしておかなかった場合は，復元中にこのコマンドが自動的に実行されますが，もしもこのコマンドが失敗すると復元も失敗します。

このコマンドは，-force オプション指定時を除いて，NNMi の起動中に実行しないでください。このコマンドを実行するときに起動してよい（していなければならない）のはnmsdbmgr プロセスだけです。

このコマンドを実行するには，Windows システムの管理者または Linux システムの root としてログインする必要があります。

Parameters

-source <*backup file*>

バックアップの復元元とするファイル名。nnmbackupembdb.ovpl コマンドで作成したファイルを指定します。

-force

このオプションを指定した場合，起動中の NNMi を停止してnmsdbmgr プロセスを開始します。

-?|-h|-help

コマンドの使用方法を表示します。

EXAMPLES

このスクリプトは，バックアップからの回復が必要な場合に，データベースの完全な回復を実行するときに使用できます。

次のメッセージが表示されます。

```
# nnmrestoreembdb.ovpl -source /backups/nnm-bak-20070929064743.pgd

警告: NNMの実行中にこのコマンドを実行すると、
          アクティブセッションにエラーが発生することがあります。このコマンド（ovstart nmsdbmg
```

```
r）の実行時に、
        nmsdbmgrプロセスのみが実行していることを確認してください。
データベースの全復元の実行を今すぐ開始してもよいですか? [n]
y
組み込みデータベースの全復元を実行しています...
データベース nnm の復元と統計情報の分析が正常に完了しました。

NNM組み込みデータベースは、/backups/nnm-bak-20070929064743.pgd から正常に復元されました。
#
```

AUTHOR

nnmrestoreembdb.ovpl was developed by Micro Focus.

FILES

nnmrestoreembdb.ovpl は，次のディレクトリにあります。

- Windows：%NNM_BIN%
- Linux：$NNM_BIN

SEE ALSO

ovstart, ovstop, ovstatus, nmsdbmgr, nnmbackupembdb.ovpl.

付録 G.46　nnmsamlconfig.ovpl

nnmsamlconfig.ovpl は，Network Node Manager i（NNMi）でサービスプロバイダー（SP）メタデータと Security Assertion Markup Language（SAML）認証設定を生成します。

SYNOPSIS

nnmsamlconfig.ovpl -spEntityID <*spEntityID*> -nameIDFormat <*nameIDFormat*> -idpMetadata <*full path to idp metadata file*> -aliasForSigning <*alias*> [-aliasForEncryption <*aliasForEncryption*> | -useSameAlias]

nnmsamlconfig.ovpl -help

DESCRIPTION

nnmsamlconfig.ovpl コマンドは，NNMi で SP メタデータ（nnm_spmetadata.xml）と SAML 設定を生成します。

コマンドが正常に実行されると，次のディレクトリに NNMi サービスプロバイダーのメタデータファイルが作成されます。

- Windows の場合：%NnmDataDir%¥nmsas¥NNM¥conf¥nnm_spmetadata.xml

- Linux の場合：$NnmDataDir/nmsas/NNM/conf/nnm_spmetadata.xml

> **メモ**
>
> nnmsamlconfig.ovpl コマンドを実行すると，現在の SAML 認証設定が存在する場合は，常にその設定が上書きされます。

nnmsamlconfig.ovpl -help オプションはヘルプを表示します。コマンドの各種オプションと使用法が表示されます。

Parameters

nnmsamlconfig.ovpl コマンドは，次のオプションをサポートします。

-spEntityID <*spEntityID*>

ID プロバイダー（IdP）で NNMi を SAML サービスプロバイダー（SP）として識別するための固有名です。このサーバーの完全修飾ドメイン名（FQDN）または URL を指定します。

-nameIDFormat <*nameIDFormat*>

ユーザー名（Subject NameID フィールド）のオプションの SAML NameID 形式を指定します。指定する場合は，サポートされている形式（サポートされている NameID 形式を参照）および IdP での選択と一致する必要があります。

デフォルト値は，urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified です。

サポートされている NameID 形式は次のとおりです。

- urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
- urn:oasis:names:tc:SAML:2.0:nameid-format:transient
- urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
- urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified
- urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName
- urn:oasis:names:tc:SAML:1.1:nameid-format:WindowsDomainQualifiedName
- urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos
- urn:oasis:names:tc:SAML:2.0:nameid-format:entity
- urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted

-idpMetadata <*full path to idp metadata file*>

IdP によって提供されるメタデータファイルの絶対パスを指定します。メタデータファイルには IdP との通信のために NNMi によって使用される設定と証明書が含まれています。

-aliasForSigning <*aliasForSigning*>

NNMi が送信する SAML 要求の署名に使用される証明書のエイリアスです。

-aliasForEncryption <*aliasForEncryption*>

NNMi への応答を暗号化するために IdP が使用する証明書のエイリアスです。

-useSameAlias

このオプションは，署名と暗号化の両方に同じ証明書を使用する場合に指定します。

-help

コマンドの使用法を表示します。

Additional Parameters

-jndiHost <*hostname*>

jndi サーバーのホスト名です。デフォルトの値は"localhost"です。

-jndiPort <*port*>

jndi サーバーのポート番号です。デフォルト値は 1099 です。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，この引数の指定が必要になります。詳細は nnm.properties リファレンスページを参照してください。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，この引数の指定が必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

Linux および Windows の場合：

```
nnmsamlconfig.ovpl -spEntityID <mfserver.net> -nameIDFormat
urn:oasis:names:tc:SAML:2.0:nameid-format:persistent -idpMetadata <metadata_File_location> -
aliasForSigning <mfserver.net.selfsigned> -aliasForEncryption <mfserver.net.selfsigned>
```

付録 G.47　nnmscheduledoutage.ovpl

ノードの停止を計画するために使用するコマンドラインツールです。

SYNOPSIS

```
nnmscheduledoutage.ovpl -u <user> -p <password> [-h | -help]
```

```
-create (-node <NODE>| -nodeGroup <NODEGROUP>| -file <FILE) <TIME-SPEC> -name <NAME> [ -
description <DESCRIPTION>] [-retroactive]
```

-delete (-name <*NAME*> [-node <*NODE*>]) | -uuid <*UUID*>

-list [-v] [-node <*NODE*> | -uuid <*UUID*> | -name <*NAME*>] [-format <*FORMAT*>]

-dump {-node <*NODE*> | <*TIME-SPEC*>} [-retroactive] [-format <*FORMAT*>]

-update (-name <*NAME*> [-node <*NODE*>] | -uuid <*UUID*>) (<*TIME-SPEC*> | -addNode <*NODE*> | -removeNode <*NODE*>)

[-newName <*NEWNAME*>] [-description <*DESCRIPTION*>] [-enable | -disable] <*TIME-SPEC*> ::= -start <*TIME*> (-duration <*DURATION*> | -end <*TIME*>) [-tz <*TIMEZONE*>]

DESCRIPTION

nnmscheduledoutage.ovpl コマンドは，ノードの計画停止を作成するために使用するコマンドラインツールです。計画された停止の期間中，停止中のノードは管理モードが「サービス停止中」に変更され，停止の期間が終了すると「管理対象」に戻されます。テキストファイルを使用して，複数のノードをノードグループ単位で個別の停止グループにまとめたり，既存の停止グループに追加したりできます。計画停止は，停止名，停止に含まれるノード，または計画停止の UUID の単位で削除できます。計画停止は，任意の参加ノード別に画面に詳細を一覧表示したり，CSV などの各種形式でフォーマットしたりできます。計画停止を更新して，ノードを追加または削除したり，停止の時間指定を変更したりできます。

Parameters

nnmscheduledoutage.ovpl コマンドは，次のオプションをサポートします。

-h | -help

コマンドの使用方法を表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。 詳細は nnm.properties リファレンスページを参照してください。

-addNode <*NODE*>

指定した計画停止にノードを追加するために，計画停止を更新するときに使用します。

-create

指定したシステムの計画停止を新規作成します。

-delete

指定した計画停止を削除します。

-description <*DESCRIPTION*>

計画停止についての追加の説明情報を提供します。

-disable

計画停止を無効化します。

-duration <*DURATION*>

計画停止の期間を分単位で設定します。これは，「分」を示す"m"（"m" を省略した場合はデフォルトで「分」を示す），「時間」を示す "h"，または「日」を示す"d"を，数字の後につけて指定できます。例えば，"3h"は 3 時間の停止です。

-enable

計画停止を有効化します。

-end <*TIME*>

計画停止の終了日時です。構文については，下記の「FORMATTING RULES」項を参照してください。終了日時が現在日時より前である場合，"-retroactive"フラグが必要です。

-file <*FILE*>

計画停止に関連付けるノードのリストを含むテキストファイルの名前です。ファイルには，空白行や"#"文字で始まるコメントを含めることができ，残りの行はノードと見なされます。ノードは，ホスト名，IP アドレス，またはノードの UUID のどれかを指定することができます。ホスト名または IP アドレスが一意でない場合は，停止が適用されるノードを決定することができないので，エラーが発生します。

-format <*FORMAT*>

一覧表示の出力形式です。'text', 'list', 'csv' および 'xml'が利用できます。

-nodeGroup <*NODEGROUP*>

ノードグループ名を指定します。

-list

指定したシステムの計画停止の情報を一覧表示します。 システムを指定しない場合は，すべての計画停止を一覧表示します。詳細表示フラグが立っている場合は，影響を受けるノードなど，詳細な情報が表示されます。UUID による一覧表示は，常に指定した計画停止の詳細な情報を提供します。

-dump

開始日時と終了日時を持ったノード停止の履歴を一覧表示します。停止の履歴には，ノードが管理対象状態から非管理対象またはサービス停止中に変更されてから（開始日時），管理対象状態に戻った場合には（終了日時），いつでも記録されます。

When the -retroactive オプションを指定した場合は，さかのぼって作成された停止のみが出力されます。

ノードパラメータを付与した場合は，指定されたノードの停止をダンプします。

時間指定を付与した場合は，その時間枠内に完了した停止をダンプします。 時間指定には開始日時を含める必要があり，さらに期間や終了日時を含めることもできます。期間や終了日時がない場合は，実

質的に，"開始日時以降のすべての停止をダンプする" ことを意味します。 開始日時は時間枠に含まれますが，終了日時は含まれません。

ノードか開始日時を指定する必要があることに注意してください。 両方を指定した場合は，指定したノードの時間枠内の停止を一覧表示します。

-name <*NAME*>

計画停止の名前です。名前は当該の停止に一意に準じる名前であり，その名前で停止を参照できます。しかし，停止の名前は必ずしも一意である必要はなく，同じ名前を持つ 2 つの停止がある場合は，対象の停止に含まれるノードを一覧表示して，さらに指定を絞り込むことができます。

-newName <*NEWNAME*>

計画停止の新しい名前です。

-node <*NODE*>

ノード名，ホスト名，管理アドレス，または UUID を指定します。

-retroactive

さかのぼって，過去に計画停止を作成します。これは，事前に計画されずに発生した，いくつかの過去の停止のために，データベース内の停止エントリを作成するために使用されます。

-removeNode <*NODE*>

指定した計画停止からノードを削除します。

-start <*TIME*>

計画停止の開始日時です（「FORMATTING RULES」項を参照してください）。

-tz <*TIMEZONE*>

タイムゾーンの指定です（「FORMATTING RULES」項を参照してください）。

-update

指定した計画停止を更新します。停止のタイミングも，含まれるノードと同様に変更できます。

-uuid

計画停止の UUID です。

FORMATTING RULES

開始日時と終了日時は，YYYY-MM-DDTHH:MM として指定します。例えば，2013-05-08T14:56 は 2013 年 5 月 8 日午後 2 時 56 分です。 開始日時と終了日時はまた，すぐに計画停止を開始するため，または，現在進行中の計画停止を終了するため，"now"として指定することができます。

上述したように，期間はそれぞれ，数字（分），または"m"，"h"，"d"に続く数字を指定することで，それぞれ分，時間，日となります。大文字小文字は無視されます。そのため，"1d"と"1D"は同じです。

タイムゾーンパラメータは，任意の有効な Java タイムゾーンパラメータを指定できます。例えば，"US/Mountain"や"Australia/Perth"です。 タイムゾーンのパラメータも"target"（大文字/小文字は区別しない）にできます。これは，計画停止の時間を対象ノードのタイムゾーンで指定することを意味します。"target"を指定した場合は，すべてのノードが同じタイムゾーンに存在する必要があります。さらに，

"target"を指定する場合は，そのノードがタイムゾーン属性を持っている必要があります。タイムゾーンには，"server"を指定することもできます。これは，ユーザーのシェルプロセスのタイムゾーンと異なるNNMi サーバーのタイムゾーンを使用する場合があることを意味します。 タイムゾーンが指定されていない場合，ユーザーのシェルプロセスのタイムゾーンが使用されます。

注：計画停止を作成するときは，開始日時と，終了日時または期間のどちらか（両方ではない）を指定しなければなりません。

注：計画停止の最小継続時間は 15 分です。15 分より短い停止は拒否されます。 また，開始日時から 15 分経過していない場合に，停止を"今すぐ(now)"終了させようとした場合も拒否されます。

EXAMPLES

ノードのタイムゾーンを使用して，1 日持続する，ノードの計画停止を作成します（2013 年 5 月 21 日を将来の日付と仮定します）。

```
nnmscheduledoutage.ovpl -create -node cisco6509 -start 2013-05-21T09:00 -tz TARGET -duration
1d -name ”Replace power supply”
```

同上ですが，遡及停止です（2013 年 5 月 21 日を，前の例の未来に代わり，過去の日付と仮定します）。

```
nnmscheduledoutage.ovpl -create -node cisco6509 -retroactive -start 2013-05-21T09:00 -tz
TARGET -duration 1d -name ”Replace power supply” -retroactive
```

ローカルの開始日時および終了日時を使用して，ノードの計画停止を作成します。

```
nnmscheduledoutage.ovpl -create -node cisco6509 -start 2013-05-21T10:00 -end 2013-05-21T14:00
-name ”Replace power supply”
```

指定されたタイムゾーンで計画停止を作成します。

```
nnmscheduledoutage.ovpl -create -node cisco6509 -start 2013-05-21T09:00 -end 2013-05-21T10:00
-tz Australia/Perth -name ”Replace power supply”
```

"nodes.txt"ファイルに記載されているノードのリストの停止を計画します。

```
nnmscheduledoutage.ovpl -create -file nodes.txt -start 2013-05-21T09:00 -tz TARGET -duration
1d -name ”Weekly Maintenance”
```

指定されたタイムゾーンで，経過時間で計画停止を作成します。ノードグループ名がこの例で使用されています。

```
nnmscheduledoutage.ovpl -create -nodeGroup Routers -start 2013-05-25T09:00 -duration 2h -tz
Australia/Perth -name ”Weekly Router Maintenance”
```

指定されたノードの計画停止を一覧表示します。計画停止の UUID，名前，開始時間，終了時間および説明がテーブルに表示されます。

```
nnmscheduledoutage.ovpl -list -node cisco6509
```

CSV 形式で計画停止を一覧表示します。

```
nnmscheduledoutage.ovpl -list -node cisco6509 -format csv
```

2013 年 7 月 1 日以降のすべての停止について，CSV 形式でノードの停止履歴をダンプします。

```
nnmscheduledoutage.ovpl -dump -node cisco6509 -format csv -start 2013-07-01T00:00
```

2013 年 12 月のすべての遡及停止を CSV 形式でダンプします。

```
nnmscheduledoutage.ovpl -dump -retroactive -start 2013-12-01T00:00 -end 2014-01-01T00:00 -
format csv
```

名前から計画停止を削除します。複数の計画停止が同じ名前を持つ場合，それらはすべて削除されます。

```
nnmscheduledoutage.ovpl -delete -name "Weekly Router Maintenance"
```

指定した計画停止だけが削除されるように，一意な計画停止の UUID によって停止を削除します。

```
nnmscheduledoutage.ovpl -delete -uuid 204846c0-a35b-4a92-9726-4ce0a8be596d
```

新たな停止終了日時により，指定した名前の計画停止の停止期間を拡張します。

```
nnmscheduledoutage.ovpl -update -name "Weekly Router Maintenance" -end 2013-05-21T12:00
```

計画停止の UUID と新しい期間（計画された開始日時からの相対）により，計画停止の停止期間を拡張します。

```
nnmscheduledoutage.ovpl -update -uuid 204846c0-a35b-4a92-9726-4ce0a8be596d -duration 2h
```

ノードと名前と新しい開始日時（ただし，同じ終了日時を維持）により，計画された停止を遅らせます。

```
nnmscheduledoutage.ovpl -update -node cisco6509 -name "Replace power supply" -start
2013-05-21T08:00
```

AUTHOR

`nnmscheduledoutage.ovpl` was developed by Micro Focus.

FILES

- Windows：`%NNM_BIN%¥nnmscheduledoutage.ovpl`
- Linux：`$NNM_BIN/nnmscheduledoutage.ovpl`

付録 G.48　nnmsecurity.ovpl

NNMi セキュリティ管理

SYNOPSIS

nnmsecurity.ovpl -help

nnmsecurity.ovpl -assignNodeToSecurityGroup ((-node <*name or hostname or management address or uuid*> -securityGroup <*name or uuid*>) | -file <*name*>) | -assignNodeToTenant ((-node <name or hostname or management address or uuid> -tenant <*name or uuid*>) | -file <*name*>) | -assignSecurityGroupToTenant (-tenant <*name or uuid*> -securityGroup <*name or uuid*>) | -assignUserGroupToSecurityGroup ((-userGroup <*name*> -securityGroup <*name or uuid*> -role <*role*>) | -file <*name*>) | -assignUserToGroup ((-user <*name*> -userGroup <*name*>) | -file <*name*>) [-u <*username*> -p <*password*>]

nnmsecurity.ovpl -createSecurityGroup ((<*name*> [-securityGroupUuid <*uuid*>] [-description <*description*>]) | -file <*name*>) | -createTenant (<*name*> [-tenantUuid <*uuid*>] [-securityGroupUuid <*uuid*>] [-description <*description*>]) | -createUserAccount ((<*username*> -role <*role*> [-password <*password*>] [-directoryServiceAccount <*true|false*>]) | -file <*name*>) | -createUserGroup ((<*name*> [-displayName <*user friendly group name*>] [-description <*description*>] [-directoryServiceName <*dn*>]) | -file <*name*>) [-u <*username*> -p <*password*>]

nnmsecurity.ovpl -deleteSecurityGroup (<*groupName or uuid*> | -file <*name*>) | -deleteUserAccount (<*name*> | -file <*name*>) | -deleteUserGroup <*name*> [-u <*username*> -p <*password*>]

nnmsecurity.ovpl -displayConfigReport [<*report*>[, <*report*>]] [-u <*username*> -p <*password*>]

nnmsecurity.ovpl -listNode <*nodeName*> | -listNodesInSecurityGroup <*groupName or uuid*> | -listSecurityGroupForTenant <*tenantName or uuid*> | -listSecurityGroups | -listTenants | -listUserGroupMembers <*groupName*> | -listUserGroups | -listUserGroupsForSecurityGroup <*groupName or uuid*> [-u <*username*> -p <*password*>

nnmsecurity.ovpl -removeUserFromGroup ((-user <*name*> -userGroup <*name*>) | -file <*name*>) | -deleteUserGroup (<*name*> | -file <*name*>) | -removeUserGroupFromSecurityGroup ((-userGroup <*groupName*> -securityGroup <*groupName or uuid*> [-role <*role*>]) | -file <*file*>) | -updateUserGroup ((<*name*> [-displayName <*user friendly group name*>] [-description <*description*>] [-directoryServiceName <*dn*>]) | -file <*name*>) [-u <*username*> -p <*password*>]

DESCRIPTION

nnmsecurity.ovpl は，NNMi セキュリティ構成を管理するために使用します。これにより，ユーザーアカウント，ユーザーグループ，セキュリティグループなどのセキュリティオブジェクトを作成，更新，削除するため，およびこれらのオブジェクトの関係を構成するためのコマンドが提供されます。

Parameters

nnmsecurity.ovpl コマンドは，次のパラメータをサポートします。

-help

コマンドの使用方法を表示します。

-assignNodeToSecurityGroup (-node <*name or hostname or management address or uuid*> -securityGroup <*name or uuid*>) | -file <*name*>

コマンドライン引数または入力ファイルを使用して，セキュリティグループにノードを割り当てます。

-node

名前，ホスト名，管理アドレス，または UUID によってノードを識別します。

-securityGroup

名前または UUID によってセキュリティグループを識別します。

-file

セキュリティグループに割り当てる，securitygroup，node という形式のノードのリストが含まれる CSV 形式のファイルへのパスです。

-assignNodeToTenant (-node <*name or hostname or management address or uuid*> -tenant <*name or uuid*>) | -file <*name*>

コマンドライン引数または入力ファイルを使用して，テナントにノードを割り当てます。ノードからテナントへの割り当ては，両方のオブジェクトを直接管理する NNMi 管理サーバーで行う必要があります。グローバルなノードからテナントへの割り当てはサポートされていません。

-node

名前，ホスト名，管理アドレス，または UUID によってノードを識別します。

-tenant

名前または UUID によってテナントを識別します。

-file

テナントに割り当てる，node，tenant という形式のノードのリストが含まれる CSV 形式のファイルへのパスです。

-assignSecurityGroupToTenant -tenant <*name or uuid*> -securityGroup <*name or uuid*>

テナントのデフォルトのセキュリティグループを変更します。新しいノードがテナントに対して設定された場合，どのセキュリティグループを使用するかを指定するため，テナントのデフォルトのセキュリティグループを使用します。この値を変更しても，既存のノードは影響を受けません。

-tenant

修正するテナントの名前または UUID です。

-securityGroup

テナントのデフォルトとして設定するセキュリティグループの名前または UUID です。

-assignUserGroupToSecurityGroup (-userGroup <*name*> -securityGroup <*name or uuid*> -role <*role*>) | -file <*name*>

ユーザーグループをセキュリティグループに割り当てます。セキュリティグループ内のノードへのアクセスをグループ内のユーザーに付与するために，ユーザーグループはセキュリティグループに割り当てられます。各割り当ては，ノードのユーザーがどのアクションを利用できるかを制御する割り当ての一部としてロールを含みます。

-userGroup

名前によって割り当てるユーザーグループを識別します。

-securityGroup

ユーザーグループを受け入れるために，名前または UUID によってセキュリティグループを識別します。

-role

キーによって割り当てに使用するロールを識別します。利用できるロールは，admin，level2，level1，guest です。

-file

userGroup，securityGroup，role という形式の割り当てのリストを含む CSV 形式のファイルへのパスです。

-assignUserToGroup (-user <*name*> -userGroup <*name*>) | -file <*name*>

ユーザーをユーザーグループに割り当てます。ユーザーは，オブジェクトへのアクセスができるグループに割り当てられます。ユーザーは，複数のグループに割り当てられ，そのグループのすべてからすべてのオブジェクトにアクセスできます。また，admin，client，level2，level1，および guest のデフォルトグループは，それらに割り当てられたユーザーに NNMi 自体で同じ名前の一致するロールを与えます。

-user

名前によって割り当てるユーザーを識別します。

-userGroup

名前によって割り当てるユーザーグループを識別します。

-file

user，userGroup という形式の割り当てのリストを含む CSV 形式のファイルへのパスです。

-createSecurityGroup (<*name*> [-securityGroupUuid <*uuid*>] [-description <*description*>]) | -file <*name*>

新しいセキュリティグループを作成します。セキュリティグループは，セキュリティ構成を簡単にするために類似するトポロジオブジェクトをグループ化します。各セキュリティグループは，名前，UUID，および説明で構成されています。

-securityGroupUuid

(オプション) 新しいセキュリティグループの UUID です。このパラメータを指定しない場合，NNMi が値を生成します。

-description

(オプション) 新しいセキュリティグループの説明です。

-file

name, uuid, description という形式のセキュリティグループのリストを含む CSV 形式のファイルへのパスです。

-createTenant <*name*> [-tenantUuid <*uuid*>] [-securityGroupUuid <*uuid*>] [-description <*description*>]

同じ名前の一致するセキュリティグループと共に新しいテナントを作成します。

-tenantUuid

(オプション) 新しいテナントの UUID です。このパラメータを指定しない場合，NNMi が値を生成します。

-securityGroupUuid

(オプション) 新しいセキュリティグループの UUID です。このパラメータを指定しない場合，NNMi が値を生成します。

-description

(オプション) 新しいテナントの説明です。

-createUserAccount (<*username*> -role <*role*> [-password <*password*>] [-directoryServiceAccount <*true/false*>]) | -file <*name*>

新しいユーザーアカウントを作成します。

-role

内部アカウントでは，ロールが指定されている必要があります。NNMi は，一致するユーザーグループに，新しいユーザーを自動的に割り当てます。ディレクトリサービスがロールを提供することがあるため，外部アカウントはロールを必要としません。

-password

新しいユーザーのパスワードです。内部アカウントにだけ使用されます。

-directoryServiceAccount

外部ディレクトリサービスが，このユーザーアカウントを管理するかどうかを指定します。NNMi データベース内部に格納されているアカウントにはfalse を使用します。 ディレクトリサービスに格納されている外部アカウントにはtrue を使用します。デフォルト値はfalse です。

-file

username, password, role, directoryServiceAccount という形式のユーザーアカウントのリストを含む CSV 形式のファイルへのパスです。

-createUserGroup (<*name*> [-displayName <*user friendly group name*>] [-description <*description*>] [-directoryServiceName <*dn*>]) | -file <*name*>

新しいユーザーグループを作成します。

-displayName

(オプション) ユーザーグループの表示名です。

-description

(オプション) 新しいグループの説明です。

-directoryServiceName

(オプション) ディレクトリサービスユーザーのためのオプションです。ディレクトリサービスで区別された名前をユーザーグループと組み合わせるために，このオプションを使用します。

-file

name, displayName, description, directoryServiceName という形式のユーザーグループのリストを含む CSV 形式のファイルへのパスです。

-deleteSecurityGroup <*groupName or uuid*> | -file <*name*>

名前または UUID によってセキュリティグループを削除します。セキュリティグループは，割り当てられたノードまたはテナントを持つことはできません。

-file

name, uuid, description という形式のセキュリティグループのリストを含む CSV 形式のファイルへのパスです。この形式は createSecurityGroup のものと同じですが，名前だけ（または存在する場合は UUID）が使用されます。

-deleteUserAccount <*name*> | -file <*name*>

名前によってユーザーアカウントを削除します。

-file

username, password, role, directoryServiceAccount という形式のユーザーアカウントのリストを含む CSV 形式のファイルへのパスです。この形式は createUserAccount のものと同じですが，削除するアカウントを一致させるために，ユーザー名だけを使用します。

-deleteUserGroup <*name*>

名前によってユーザーグループを削除します。

-displayConfigReport [<*report*>[, <*report*>]]

セキュリティ構成レポートを表示します。利用できるレポートは，emptySecurityGroups, emptyUserGroups, securityGroupsWithSameName, usersWithoutGroups, tenantsWithSameName, usersWithoutRoles です。

レポートを指定しない場合，利用可能なすべてのレポートが実行されます。

-listNode <*node name*>

セキュリティグループの UUID および指定したノードに関連するテナントを表示します。ノードは，名前または UUID として指定することができます。出力には，ノードの UUID およびホスト名，セキュリティグループの UUID および名前，テナントの UUID および名前が別の行に表示されます。

-listNodesInSecurityGroup <*groupName or uuid*>

セキュリティグループ名または UUID によってセキュリティグループ内のノードを一覧表示します。

-listSecurityGroupForTenant <*tenantName or uuid*>

指定したテナント名または UUID に対して構成された，初期検出セキュリティグループを表示します。

-listSecurityGroups

すべての構成されたセキュリティグループの名前を一覧表示します。

-listTenants

すべての構成されたテナントの名前を一覧表示します。

-listUserGroupMembers <*groupName*>

指定したユーザーグループのユーザーを一覧表示します。

-listUserGroups

すべての構成されたユーザーグループを一覧表示します。

-listUserGroupsForSecurityGroup <*groupName or uuid*>

指定したセキュリティグループ名または UUID に関連するユーザーグループを一覧表示します。

-removeUserFromGroup (-user <*name*> -userGroup <*name*>) | -file <*filename*>

ユーザーアカウントとユーザーグループ間のマッピングを削除します。

-user

修正するユーザーアカウントのユーザー名です。

-userGroup

指定したユーザーアカウントからマッピングを解除するユーザーグループの名前です。

-file

user, userGroup という形式のユーザーグループマッピングへのユーザーのリストを含む CSV 形式のファイルへのパスです。

-deleteUserGroup <*name*> | -file <*name*>

名前によってユーザーグループを削除します。ユーザーグループとユーザーアカウントの間のマッピングおよびセキュリティグループも削除されます。

-file

name, displayName, description, directoryServiceName という形式のユーザーグループマッピングへのユーザーのリストを含む CSV 形式のファイルへのパスです。この形式は createUserGroup のものと同じですが，削除するグループに一致させるために，名前だけを使用します。

-removeUserGroupFromSecurityGroup (-userGroup <*groupName*> -securityGroup <*groupName or uuid*> [-role <*role*>]) | -file <*name*>

ユーザーグループとセキュリティグループの間のマッピングを削除します。

-userGroup

ユーザーグループの名前です。

-securityGroup

セキュリティグループの名前または UUID です。

-role

オプションのロールです。ロールが指定されていない場合，すべてのロールのマッピングが削除されます。

-file

userGroup，securityGroup，role という形式のユーザーグループマッピングへのユーザーのリストを含む CSV 形式のファイルへのパスです。

-updateUserGroup <*name*> ([-displayName <*user friendly group name*>] [-description <*description*>] [-directoryServiceName <*dn*>]) | -file <*name*>

ユーザーグループを更新します。名前を除くすべてのユーザーグループ属性を更新できます。

-displayName

(オプション) ユーザーグループの表示名です。

-description

(オプション) ユーザーグループの説明です。

-directoryServiceName

(オプション) ディレクトリサービスユーザーのためのオプションです。ディレクトリサービスで区別された名前をユーザーグループと組み合わせるために，このオプションを使用します。

-file

name，displayName，description，directoryServiceName という形式のユーザーグループのリストを含む CSV 形式のファイルへのパスです。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

nnmsecurity.ovpl -createTenant myTenant

"myTenant"という名前のテナントを作成します。

nnmsecurity.ovpl -listTenants

すべての構成されたテナントを一覧表示します。

nnmsecurity.ovpl -createTenant ”Tenant with a space”

nnmsecurity.ovpl -createTenant ¥!Tenant

使用するシェルに基づいて，名前にスペースが含まれるテナントを作成するために，テナント名の前後に引用符を使用したり，名前内に特殊文字があるテナントを作成するために，エスケープ文字を使用したりすることができます。

nnmsecurity.ovpl -createSecurityGroup mySecurityGroup

セキュリティグループ"mySecurityGroup"を作成します。

nnmsecurity.ovpl -createSecurityGroup ”Group with a space”

nnmsecurity.ovpl -createSecurityGroup ¥!MyGroup

使用するシェルに基づいて，名前にスペースが含まれるセキュリティグループを作成するために，セキュリティグループ名の前後に引用符を使用したり，名前内に特殊文字があるセキュリティグループを作成するために，エスケープ文字を使用したりすることができます。

nnmsecurity.ovpl -listSecurityGroups

すべての構成されたセキュリティグループを一覧表示します。

nnmsecurity.ovpl -listNode myNode

関連するセキュリティグループおよび提供されたノードのテナントを一覧表示します。

DIAGNOSTICS

nnmsecurity.ovpl コマンドは，次の終了コードを返します。

0

処理は成功しました。

1

エラーが発生しました。詳細はエラーメッセージを参照してください。

AUTHOR

nnmsecurity.ovpl was developed by Micro Focus.

FILES

次の環境変数は，使用するシェルおよびプラットフォームの要件に従って設定される共通パスです。

- Windows：%NNM_BIN%¥nnmsecurity.ovpl
- Linux：$NNM_BIN/nnmsecurity.ovpl

SEE ALSO

nnm.properties.

付録 G.49　nnmseeddelete.ovpl

NNMi トポロジデータベースからシードを削除します。

SYNOPSIS

nnmseeddelete.ovpl -help | -f <*seedFile*> | -seed <*seed*> | -all [-u <*username*> -p <*password*>]

DESCRIPTION

nnmseeddelete.ovpl コマンドは，システムからシードを削除します。

Parameters

nnmseeddelete.ovpl コマンドは，次のオプションをサポートします。

-help

コマンドの使用方法を表示します。

-seed <*seed*>

削除するシードを指定します。シードはホスト名または IP アドレスで，シードリストに記載されている内容に一致する必要があります。

-f <*seedFile*>

シードの読み込み元であるテキストファイルを指定します。

-all

すべてのシードを削除します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

nnmseeddelete.ovpl -u username -p password -seed 10.1.2.3

シード10.1.2.3 が削除されます。（NNMi 管理者のユーザー名とパスワードを指定する必要があります。）

nnmseeddelete.ovpl -f /tmp/seeds_to_delete.txt

ファイルに記載されている各シードが削除されます。

Diagnostics

nnmseeddelete.ovpl コマンドは，次の終了コードを返します。

0

処理は成功しました。

1

エラーが発生しました。詳細はエラーメッセージを参照してください。

2

エラーが発生しました。詳細はエラーメッセージを参照してください。

AUTHOR

nnmseeddelete.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmseeddelete.ovpl
- Linux：$NNM_BIN/nnmseeddelete.ovpl

SEE ALSO

nnmloadseeds.ovpl, nnm.properties.

付録 G.50　nnmsetdampenedinterval.ovpl

すべてのインシデントの設定に対するダンプニングの期間を設定します。

SYNOPSIS

nnmsetdampenedinterval.ovpl [[-hours *hours*] [-minutes *minutes*] [-seconds *seconds*] [-u *username*] [-p *password*]]

DESCRIPTION

nnmsetdampenedinterval.ovpl コマンドは，すべてのインシデントの設定に対するダンプニングの期間を設定します。設定できるダンプニングの期間の最大は 60 分です。設定する場合，ダンプニングの期間は少なくとも 6 分を推奨します。少なくとも，hours, minutes, seconds の一つを指定する必要があります。ダンプニングを無効にするには，hours, minutes, seconds に 0 を設定します。

Parameters

-hours *hours*

ダンプニング期間の時間を指定します。指定する場合，値は0 以上を指定してください。

-minutes *minutes*

ダンプニング期間の分を指定します。指定する場合，値は0 以上を指定してください。

-seconds *seconds*

ダンプニング期間の秒を指定します。指定する場合，値は0 以上を指定してください。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

EXAMPLES

すべてのインシデント設定のダンプニングの期間を 10 分 10 秒に設定します。

```
nnmsetdampenedinterval.ovpl -hours 0 -minutes 10 -seconds 10
```

すべてのインシデントの設定のダンプニングの期間を 1 時間に設定します。

```
nnmsetdampenedinterval.ovpl -hours 1
```

すべてのインシデントの設定のダンプニングの期間を 6 分に設定します。

```
nnmsetdampenedinterval.ovpl -minutes 6
```

すべてのインシデントの設定のダンプニングの期間を 30 秒に設定します。

```
nnmsetdampenedinterval.ovpl -seconds 30
```

すべてのインシデント設定のダンプニングの期間を 10 分 10 秒に設定します。

```
nnmsetdampenedinterval.ovpl -minutes 10 -seconds 10
```

すべてのインシデント設定のダンプニングを無効にします。

```
nnmsetdampenedinterval.ovpl -hours 0 -minutes 0 -seconds 0
```

AUTHOR

nnmsetdampenedinterval.ovpl was developed by Micro Focus.

SEE ALSO

nnm.properties

付録 G.51　nnmsetofficialfqdn.ovpl

NNMi 管理サーバーの正式な完全修飾名（FQDN）を設定します。

SYNOPSIS

```
nnmsetofficialfqdn.ovpl [-f | -force] <fqdn>
```

DESCRIPTION

インストール後に NNMi 管理サーバーの正式な完全修飾ドメイン名（FQDN）を変更するには，nnmsetofficialfqdn.ovpl スクリプトを使用します。 このスクリプトを引数なしで実行すると，正式な FQDN をデフォルト値に設定します。デフォルト値は，ホスト名のルックアップの実行により取得されます。 デフォルトのホスト名ルックアップについては，-m オプションを使用するnnmofficialfqdn.ovpl スクリプトのリファレンスページを参照してください。

FQDN が変更されると，ユーザーは新しい自己署名 SSL 証明書を新しいホスト用に生成するよう求められます。また，CA 署名の証明書を使用している場合は，新しいホスト名で，CA から新しい証明書を入手する必要があります。

Parameters

nnmsetofficialfqdn.ovpl コマンドに任意で指定できる引数は 2 種類あります。

<fqdn>

ユーザーに確認した上で，正式な完全修飾名を指定値*<fqdn>*に変更します。

-force|-f

確認メッセージを表示せず，強制的に正式 FQDN を変更するときに指定するフラグです。-f オプションを単独で指定した場合は，デフォルト値が正式 FQDN として設定されます。このオプションの後に*<fqdn>*を指定した場合は，指定値が正式 FQDN として設定されます。

EXAMPLES

FQDN をデフォルト値に設定する場合：

```
nnmsetofficialfqdn.ovpl
```

FQDN を"somehost.somedomain"に変更する場合：

```
nnmsetofficialfqdn.ovpl somehost.somedomain
```

正式 FQDN を強制的にデフォルト値に設定する場合：

```
nnmsetofficialfqdn.ovpl -f
```

AUTHOR

nnmsetofficialfqdn.ovpl was developed by Micro Focus.

FILES

nnmsetofficialfqdn.ovpl は%NNM_BIN%（Windows）または$NNM_BIN（Linux）ディレクトリにあります。

SEE ALSO

nnmofficialfqdn.ovpl.

付録 G.52　nnmsnmpbulk.ovpl

SNMPv2c GetBulk リクエストでノード情報について問い合わせを行います。

SYNOPSIS

nnmsnmpbulk.ovpl -u username -p password [*options*] *node object-id* [*object-id*]...

options: [-d] [-v *version*] [-c *community*] [-port *port(default:161)*] [-t *timeout(default:5000)*] [-r *retries(default:1)*] [-T] [-n *non-repeaters*] [-m *max-repetitions*] [-pp *Proxy Port*] [-pa *Proxy Address*] [-a *Authentication Protocol*] [-A *Authentication Pass phrase*] [-x *Privacy Protocol*] [-X *Privacy Passphrase*] [-N *Context Name*] [-oen *OID and Output Encoding*] [-oex *OIDs that are not encoded*] [-v3u *SNMPv3 user name*]

DESCRIPTION

nnmsnmpbulk.ovpl コマンドは，SNMP エージェントから情報を取得するため，SNMPv2c/v3 GetBulk リクエストを使用します。SNMP GetBulk リクエストは，大量の情報を取得するときに必要なプロトコル交換の回数を最小限に抑えます。リモートノードから管理情報を取得するときに必要なリクエストの回数が減少するため，性能が向上します。

node が SNMPv1 のみのエージェントの場合，このコマンドは，GetBulk リクエストを SNMPv1 対応の GetNext リクエストに自動的にダウングレードします。

node は，IP アドレスを持つ SNMP をサポートしているシステムです。IP アドレスまたはホスト名により指定できます。

コマンドに対する引数として，単数または複数のOID を指定することができます。各 OID は，オブジェクト識別子を 10 進ドット形式またはニモニック名で表したものです。ニモニック名で指定した場合，OID を定義する MIB をnnmloadmib.ovpl コマンドを使用してロードする必要があります。

このコマンドを実行できるのは，ロールが System，管理者，または Web サービス クライアントのユーザーに限定されます。ロールがレベル 1 オペレータ，レベル 2 オペレータ，またはゲストのユーザーは，このコマンドを実行することができません。

Parameters

-d

SNMP パケットを 16 進数形式と ASN.1 形式で標準出力にダンプします。

-v *version*

リモートノードとの通信に使用する SNMP のバージョンを指定します。*version* に有効な値は，1，2，2c または3 です。

このトポロジに存在しないノードで，値が指定されていない場合，デフォルトは2c になります。

-c *community*

リモートノード上で認証に使用するコミュニティ文字列を指定します。

注記：シェルに影響する文字がコミュニティ文字列に含まれている場合は，必要に応じて一つ以上のエスケープ文字または引用符を使用してください。

-port *port*

リモートノードと通信するときに使用するポートを指定します。

-t *timeout*

リモートノードと通信するときに使用するタイムアウト期間をミリ秒単位で指定します。

-r *retries*

リモートノードと通信するときに使用するリトライ数を指定します。

-T

10 進ドット形式の OID およびテキスト形式の規定が適用されていない MIB 変数値を出力します。

-n *non-repeaters*

non-repeaters は辞書的な順序で次候補を一つ返す変数の数を指定します。この値は，nnmsnmpbulk.ovpl コマンドで取得する繰り返しのない varbind（値）の数を示しています。

-m *max-repetitions*

max-repetitions は，残りの変数に対して，辞書的な順序で次候補をいくつ返すかを指定します。この値は，繰り返しのある varbind（値）に対して取得する行の数を示します。繰り返しのある varbind は，各テーブル行にあります。

-pp *Proxy Port*

ノードと通信するときに使用するプロキシポートを指定します。

-pa *Proxy Address*

ノードと通信するときに使用するプロキシ IP アドレスを指定します。

-a *Authentication Protocol*

SNMPv3 認証プロトコル(MD5|SHA|SHA224|SHA256|SHA384|SHA512)

-A *Authentication Passphrase*

SNMPv3 認証パスフレーズ

-x *Privacy Protocol*

SNMPv3 プライバシプロトコル(DES|3DES|AES|AES192|AES256)

-X *Privacy Passphrase*

SNMPv3 プライバシパスフレーズ

-N *Context Name*

SNMPv3 コンテキスト名 (例えば vlan1)

-oen <*OID*> : <*encoding*>

OID および出力エンコード（例えば 1.3.6.1.2.1.1.4:UTF-8）

-oex <*OID1,OID2,...*>

エンコードされていない OID

-v3u *SNMPv3 user name*

SNMPv3 ユーザー名(例えば testV3user)

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

nnmsnmpbulk.ovpl コマンドは，応答が受信されなかった場合に，線形のバックオフアルゴリズムを使用し，*timeout* と*retries* に基づいて SNMP リクエストを再送信します。 例えば，*timeout* が2000（2 秒）で*retries* が3 の場合，初期リクエストは 2 秒後にタイムアウトし，最初のリトライは 4 秒後にタイムアウトし，2 番目のリトライは 6 秒後にタイムアウトし，最後のリトライは 8 秒後にタイムアウトします。設定を解決するには，さらに時間が必要になることもあります。

EXAMPLES

次のコマンドは，システム MIB オブジェクト識別子の下にあるすべての情報をノードtestnode から取得します。

```
nnmsnmpbulk.ovpl -c community testnode .1.3.6.1.2.1.1.0
```

AUTHOR

nnmsnmpbulk.ovpl was developed by Micro Focus.

FILES

- Windows：%NNM_BIN%¥nnmsnmpbulk.ovpl
- Linux：$NNM_BIN/nnmsnmpbulk.ovpl

SEE ALSO

nnmloadmib.ovpl, nnmsnmpnotify.ovpl, nnmsnmpset.ovpl, nnmsnmpwalk.ovpl.

RFC 1155, 1157, 1212: SNMP Version 1.

RFC 1901 - 1908, 2576, 2578, 3416 - 3418: SNMP Version 2.

RFC 3411 - 3415: SNMP Version 3.

EXTERNAL INFLUENCES

Environmental Variables

$LANG は，メッセージを表示するときの言語を決定します。$LANG が指定されていない場合，または空の文字列に設定された場合，$LANG ではなくC がデフォルトに使用されます。国際化変数のどれかに無効な設定値がある場合，nnmsnmpbulk.ovpl は，国際化変数のすべてがC に設定されているように処理します。

International Code Set Support

シングルバイトまたはマルチバイトの文字コードセットをサポートします。

注記：DISPLAY STRING タイプの SNMP MIB 値は VT-ASCII に限定されています。

付録 G.53　nnmsnmpset.ovpl

SNMP セットリクエストを発行します。

SYNOPSIS

nnmsnmpset.ovpl -u *username* -p *password* [*options*] *node object-id asnType value* [*object-id asnType value*]...

options: [-d] [-v *version*] [-c *write community*] [-port *port(default:161)*] [-t *timeout(default:5000)*] [-r *retries(default:1)*(] [-T] [-pp *Proxy Port*] [-pa *Proxy Address*] [-a *Authentication Protocol*] [-A *Authentication Pass phrase*] [-x *Privacy Protocol*] [-X *Privacy Passphrase*] [-N *Context Name*] [-v3u *SNMPv3 user name*]

DESCRIPTION

nnmsnmpset.ovpl コマンドは，リモート*node* 上の MIB オブジェクトを変更するために SNMP セットリクエストを発行します。

object-id , *asnType* , *value* の三つを一組としてnnmsnmpset.ovpl コマンドからリモートノードに渡されるデータを指定します。ユーザーは，この組を少なくとも一つコマンドラインの引数で指定する必要があります。

各*object-id* は，10 進ドット形式のオブジェクトインスタンス識別子（例えば，.1.3.6.1.4.1.11.2.17.2.1.0）またはニモニックの文字列（例えば，openViewSourceId.0）です。

各*asnType* は，次の asnTypes のいずれかとします。

integer
integer32
unsigned32
octetstring
octetstringhex
octetstringoctal
octetstringascii
objectidentifier
null
ipaddress
counter
counter32
counter64(SNMPv2c または v3 に対応するリモートノードの場合)
gauge
gauge32
timeticks
opaque
opaquehex
opaqueoctal
opaqueascii

各*asnType* の詳細説明は，*RFC 1155* および *RFC 1902* を参照してください。

このときの*value* パラメータは，指定された asnType で有効である必要があります。16 進数または 8 進数の値が必要な asnType を使用する場合，ユーザーがこの値の各バイトを完全に定義する必要があります。 例えば，fff(または17377)を指定すると，1 バイト足りないため動作しません。代わりに0fff(または017377)を使用してください。asnType が*null* の場合，ユーザーはコマンドライン上で*value* を指定する

必要があります。リクエストの生成時，この*value* が無視されます。この*value* は，512 バイトを超えてはなりません。

このコマンドを実行できるのは，ロールが System，管理者，または Web サービスクライアントのユーザーに限定されます。ロールがレベル 1 オペレータ，レベル 2 オペレータ，またはゲストのユーザーは，このコマンドを実行することができません。

Parameters

-d

SNMP パケットを 16 進数形式と ASN.1 形式で標準出力にダンプします。

-v *version*

リモートノードとの通信に使用する SNMP のバージョンを指定します。*version* に有効な値は，1，2c または 3 です。

このトポロジに存在しないノードで，値が指定されていない場合，デフォルトは 2c になります。

-c *write community*

リモートノード上で認証に使用する書き込みコミュニティ文字列を指定します。

注記：シェルに影響する文字がコミュニティ文字列に含まれている場合は，必要に応じて一つ以上のエスケープ文字または引用符を使用してください。

-port *port*

リモートノードと通信するときに使用するポートを指定します。

-t *timeout*

リモートノードと通信するときに使用するタイムアウト期間をミリ秒単位で指定します。

-r *retries*

リモートノードと通信するときに使用するリトライ数を指定します。

-T

10 進ドット形式の OID およびテキスト形式の規定が適用されていない MIB 変数値を出力します。

-pp *Proxy Port*

ノードと通信するときに使用するプロキシポートを指定します。

-pa *Proxy Address*

ノードと通信するときに使用するプロキシ IP アドレスを指定します。

-a *Authentication Protocol*

SNMPv3 認証プロトコル(MD5|SHA|SHA224|SHA256|SHA384|SHA512)

-A *Authentication Passphrase*

SNMPv3 認証パスフレーズ

-x *Privacy Protocol*

SNMPv3 プライバシプロトコル(DES|3DES|AES|AES192|AES256)

-X *Privacy Passphrase*

SNMPv3 プライバシパスフレーズ

-N *Context Name*

SNMPv3 コンテキスト名(例えば vlan1)

-v3u *SNMPv3 user name*

SNMPv3 ユーザー名(例えば testV3user)

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

nnmsnmpset.ovpl コマンドは，応答が受信されなかった場合に，線形のバックオフアルゴリズムを使用し，*timeout* と *retries* に基づいて SNMP リクエストを再送信します。 例えば，*timeout* が 2000(2 秒)で *retries* が3 の場合，初期リクエストは 2 秒後にタイムアウトし，最初のリトライは 4 秒後にタイムアウトし，2 番目のリトライは 6 秒後にタイムアウトし，最後のリトライは 8 秒後にタイムアウトします。設定を解決するには，さらに時間が必要になることもあります。

EXAMPLES

次のコマンドは，ノード testnode の sysContact の値を Bob Jones に設定します。

```
nnmsnmpset.ovpl -c writeCommunity testnode system.sysContact.0 octetstring "Bob Jones"
```

このコマンドの出力は次のとおりです。

```
sysContact.0 : OCTET STRING- (ascii): Bob Jones
```

AUTHOR

nnmsnmpset.ovpl was developed by Micro Focus.

FILES

Windows：

%NNM_BIN%¥nnmsnmpset.ovpl

%NnmInstallDir%¥doc¥rfc*.txt

Linux：

$NNM_BIN/nnmsnmpset.ovpl

$NnmInstallDir/doc/rfc*.txt

SEE ALSO

nnmsnmpwalk.ovpl, nnmsnmpget.ovpl, nnmsnmpnext.ovpl, nnmsnmpbulk.ovpl, nnmsnmpnotify.ovpl.

RFC 1155, 1157, 1212: SNMP Version 1.

RFC 1901 - 1908, 2576, 2578, 3416 - 3418: SNMP Version 2.

RFC 3411 - 3415: SNMP Version 3.

EXTERNAL INFLUENCES

Environmental Variables

`$LANG` は，メッセージを表示するときの言語を決定します。`$LANG` が指定されていない場合，または空の文字列に設定された場合，`$LANG` ではなくC がデフォルトに使用されます。国際化変数のどれかに無効な設定値がある場合，`nnmsnmpset.ovpl` は，国際化変数のすべてがC に設定されているように処理します。

International Code Set Support

シングルバイトまたはマルチバイトの文字コードセットをサポートします。

注記：asnType が`octetstringascii` の SNMP MIB 値は VT-ASCII に限定されています。

付録 G.54　nnmsnmpwalk.ovpl, nnmsnmpget.ovpl, nnmsnmpnext.ovpl

SNMP GET リクエストまたは SNMP GETNEXT リクエストを使用してノードについて問い合わせを行います。

SYNOPSIS

`nnmsnmpwalk.ovpl -u` *`username`* `-p` *`password`* [*`options`*] *`node object-id`*

`nnmsnmpget.ovpl -u` *`username`* `-p` *`password`* [*`options`*] *`node object-id`* [*`object-id`*]...

`nnmsnmpnext.ovpl -u` *`username`* `-p` *`password`* [*`options`*] *`node object-id`* [*`object-id`*]...

options: [-d] [-v *version*] [-c *community*] [-port *port(default:161)*] [-t *timeout(default:5000)*] [-r *retries(default:1)*] [-T] [-pp *Proxy Port*] [-pa *Proxy Address*] [-a *Authentication Protocol*] [-A *Authentication Pass phrase*] [-x *Privacy Protocol*] [-X *Privacy Passphrase*] [-N *Context Name*] [-oen *OID and Output Encoding*] [-oex *OIDs that are not encoded*] [-v3u *SNMPv3 user name*]

DESCRIPTION

nnmsnmpwalk.ovpl コマンドは，SNMP GETNEXT リクエストを繰り返し発行し，node で登録されている MIB オブジェクトのすべてのインスタンスの値を取得します。nnmsnmpwalk.ovpl コマンドは，-v オプションの値およびリモートノードのタイプに基づき，SNMP Version1 または Community 対応の SNMP Version2（SNMPv2c）または version3 のいずれを使用するかを自動的に決定します。変数を指定しなかった場合，nnmsnmpwalk.ovpl コマンドは，object.iso.org の下にあるすべての値を取得します。その他の場合，検索対象のオブジェクト識別子空間の開始点が変数の値で決定されます。指定変数下のオブジェクト情報がすべて返ってきたとき， nnmsnmpwalk.ovpl が終了します。例えば，システムグループ全体を検索するには.1.3.6.1.2.1.1 を使用します。

nnmsnmpget.ovpl コマンドは，SNMP Get リクエストを使用してnode についての情報を照会します。

一般に，SNMP インスタンス番号を追加する必要があります（例えば，system.sysDescr.0 値の取得には.1.3.6.1.2.1.1.1.0 を使用します）。

nnmsnmpnext.ovpl コマンドはnnmsnmpwalk.ovpl コマンドと同様の動作をしますが，nnmsnmpnext.ovpl コマンドが単一の値のみを返す点が異なります。

node は，IP アドレスを持つ SNMP をサポートしているシステム，または SNMP プロキシ構成を定義するターゲット名です。IP アドレスまたはホスト名により指定できます。

コマンドに対する引数として，単数または複数の変数を指定することができます。各変数は，オブジェクト識別子を 10 進ドット形式またはニモニック名で表したものです。ニモニック名で指定した場合，オブジェクト識別子を定義する MIB はnnmloadmib.ovpl コマンドを使用してロードする必要があります。

nnmsnmpwalk.ovpl またはnnmsnmpnext.ovpl でリモートノードの MIB を超えて検索すると，「MIB ビューの終了。」というメッセージが返ってきます。

このコマンドを実行できるのは，ロールが System，管理者，または Web サービス クライアントのユーザーに限定されます。ロールがレベル 1 オペレータ，レベル 2 オペレータ，またはゲストのユーザーは，このコマンドを実行することができません。

Parameters

-d

SNMP パケットを 16 進数形式と ASN.1 形式で標準出力にダンプします。

-v *version*

リモートノードとの通信に使用する SNMP のバージョンを指定します。*version* に有効な値は，1，2c または3 です。

このトポロジに存在しないノードで，値が指定されていない場合，デフォルトは2c になります。

-c *community*

リモートノード上で認証に使用するコミュニティ文字列を指定します。

注記：シェルに影響する文字がコミュニティ文字列に含まれている場合は，必要に応じて一つ以上のエスケープ文字または引用符を使用してください。

-port *port*

リモートノードと通信するときに使用するポートを指定します。

-t *timeout*

リモートノードと通信するときに使用するタイムアウト期間をミリ秒単位で指定します。

-r *retries*

リモートノードと通信するときに使用するリトライ数を指定します。

-T

10 進ドット形式の OID およびテキスト形式の規定が適用されていない MIB 変数値を出力します。

-pp *Proxy Port*

ノードと通信するときに使用するプロキシポートを指定します。

-pa *Proxy Address*

ノードと通信するときに使用するプロキシ IP アドレスを指定します。

-a *Authentication Protocol*

SNMPv3 認証プロトコル(MD5|SHA|SHA224|SHA256|SHA384|SHA512)

-A *Authentication Passphrase*

SNMPv3 認証パスフレーズ

-x *Privacy Protocol*

SNMPv3 プライバシプロトコル(DES|3DES|AES|AES192|AES256)

-X *Privacy Passphrase*

SNMPv3 プライバシパスフレーズ

-N *Context Name*

SNMPv3 コンテキスト名 (例えば vlan1)

-oen *<OID>* : *<encoding>*

OID および出力エンコード

収集された MIB のオクテット文字列のエンコードされた文字列が，そのオクテット文字列が提供されたとおりにエンコードされた場合にのみ，出力に示されます。 オクテット文字列が指定されたエンコーディングでエンコードされない場合，コマンドは 16 進数の文字列を示します。

-oex *<OID1,OID2,...>*

エンコードされていない OID

エンコード対象から除外された OID を指定します。このオプションで指定された OID はエンコードされず，16 進数として出力されます。

-v3u *SNMPv3 user name*

SNMPv3 ユーザー名(例えば testV3user)

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

nnmsnmpget.ovpl，nnmsnmpnext.ovpl，およびnnmsnmpwalk.ovpl コマンドは，応答が受信されなかった場合に，線形のバックオフアルゴリズムを使用し，*timeout* と*retries* に基づいて SNMP リクエストを再送信します。例えば，*timeout* が2000（2 秒）で *retries* が 3 の場合，初期リクエストは 2 秒後にタイムアウトし，最初のリトライは 4 秒後にタイムアウトし，2 番目のリトライは 6 秒後にタイムアウトし，最後のリトライは 8 秒後にタイムアウトします。設定を解決するには，さらに時間が必要になることもあります。

EXAMPLES

次のコマンドで，*testnode* に対するシステムサブツリーを要求します。

```
nnmsnmpwalk.ovpl -c community testnode system
```

通常，上記コマンドの出力は下記のとおりです。

```
sysDescr.0 : OCTET STRING- (ascii): Ethernet Switch 470-24T-PWR
sysObjectID.0 : OBJECT IDENTIFIER: .1.3.6.1.4.1.45.3.63.1
sysUpTime.0 : Timeticks: (2975913) 8:15:59.13
sysContact.0 : OCTET STRING- (ascii): Bob Jones 933-558-3453
sysName.0 : OCTET STRING- (ascii): wr3-2-front-storage-n91-60-2
sysLocation.0 : OCTET STRING- (ascii): Woods Run 3 2nd floor
sysServices.0 : INTEGER: 3
```

以下に，ノード 192.168.50.1 にオプション-oen と-oex を使用する例を示します。

```
nnmsnmpwalk.ovpl -v 2c -c public -oen .1.3.6.1.2.1.1.4:shift-jis 192.168.50.1 .1.3.6.1.2.1.1
```

収集された MIB のオクテット文字列の Shift-JIS にエンコードされた文字列が，そのオクテット文字列が有効な Shift-JIS である場合にのみ，出力に示されます。 オクテット文字列が指定されたエンコーディングでエンコードされない場合，コマンドは 16 進数の文字列を示します。

```
nnmsnmpwalk.ovpl -v 2c -c public -oen .1.3.6.1.2.1.1.4:shift-jis -oex .1.3.6.1.2.1.1.4.0
192.168.50.1 .1.3.6.1.2.1.1
```

.1.3.6.1.2.1.1 の MIB の戻り値と，1.3.6.1.2.1.1.4.0 を除く 1.3.6.1.2.1.1.4 の子である 16 進数の値を Shift-JIS でエンコードした文字列が出力に示されます。

AUTHOR

nnmsnmpwalk.ovpl, nnmsnmpget.ovpl, and nnmsnmpnext.ovpl were developed by Micro Focus.

FILES

Windows：

```
%NNM_BIN%¥nnmsnmpwalk.ovpl
%NNM_BIN%¥nnmsnmpget.ovpl
%NNM_BIN%¥nnmsnmpnext.ovpl
```

Linux：

```
$NNM_BIN/nnmsnmpwalk.ovpl
$NNM_BIN/nnmsnmpget.ovpl
$NNM_BIN/nnmsnmpnext.ovpl
```

ユーザーのプラットフォームおよびシェルに対する一般的なパスの詳細は，付録 F.1　nnm.envvars のリファレンスページを参照してください。

SEE ALSO

nnmsnmpset.ovpl, nnmsnmpbulk.ovpl, nnmsnmpnotify.ovpl.

RFC 1155, 1157, 1212: SNMP Version 1.

RFC 1901 - 1908, 2576, 2578, 3416 - 3418: SNMP Version 2.

RFC 3411 - 3415: SNMP Version 3.

EXTERNAL INFLUENCES

Environmental Variables

$LANG は，メッセージを表示するときの言語を決定します。$LANG が指定されていない場合，または空の文字列に設定された場合，$LANG ではなくC がデフォルトに使用されます。国際化変数のどれかに無効な設定値がある場合，nnmsnmpget.ovpl，nnmsnmpnext.ovpl および nnmsnmpwalk.ovpl は，国際化変数のすべてが C に設定されているように処理します。

International Code Set Support

シングルバイトまたはマルチバイトの文字コードセットをサポートします。

注記：タイプDISPLAY STRING の SNMP MIB 値は VT-ASCII に限定されています。

付録 G.55　nnmstatuspoll.ovpl

このスクリプトは，状態ポーラーによりノードの状態を更新します。

SYNOPSIS

nnmstatuspoll.ovpl [-help | -node <*nodename*|*IP Address*> [-tenant *tenant name*] [-t *timeout in secs*] [-v] [-u <*username*> -p <*password*>]]

DESCRIPTION

nnmstatuspoll.ovpl コマンドを使用すると，監視対象のデバイスを動的にポーリングすることができます。この結果，キー収集された状態値が更新されます。状態要求ポーリングに対する情報がすべて収集・表示されると，nnmstatuspoll.ovpl コマンドは，ユーザーが要求したタスクが完了したことをユーザーに知らせます。

Parameters

-node <*nodename*|*IP Address*>

ターゲットノード名または IP アドレスを指定します。

-tenant <*tenant name*>

特定のノードと組み合わせるテナントを指定します。このオプションは，アドレスのドメイン環境が重複しているなどの理由で，ノードの名前および IP アドレスがトポロジにおいて一意でない場合に便利です。

-t <*timeout in secs*>

指定されたタイムアウト時間（秒単位）までクライアントが待ちます。

-v

詳細なログメッセージをコンソールに表示します。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-help

コマンドの使用方法を表示します。

RETURN VALUE

nnmstatuspoll.ovpl は，上記パラメータセクションに表示されている適当な出力を返してきます。

-v オプションを使用すると，下記のカラムに情報が表示されます。

カラム 1：データの収集にどのプロトコルを使用するかを表示します。

カラム 2：ポーリングの対象であったデバイス名を表示します。

カラム 3：ポーリングの対象であった MIB インスタンスを表示します。

カラム 4：ポーリングの結果を表示します。

カラム 5：マップされた値があれば，その値を表示します。

AUTHOR

`nnmstatuspoll.ovpl` was developed by Micro Focus.

SEE ALSO

nnm.properties

付録 G.56　nnmtopodump.ovpl

NNMi トポロジデータベースの内容を出力します。

SYNOPSIS

`nnmtopodump.ovpl -h | -u <username> -p <password> -type <type> [-legacy <format>] [-filter <filter>] [-http.host <host>] [-http.port <port>]`

DESCRIPTION

`nnmtopodump.ovpl` はトポロジデータベースの内容を出力します。 デフォルトでは，`-legacy` オプションを指定しない限り，xml 形式で出力されます。

Parameters

`nnmtopodump.ovpl` コマンドは，次のパラメータをサポートします。

`-h`

コマンドの使用方法を表示します。

`-u <username>`

コマンドの実行に必要な NNMi の管理者名を指定します。 `nnm.properties` ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

`-p <password>`

コマンドの実行に必要な NNMi の管理者パスワードを指定します。`nnm.properties` ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

`-http.host <host>`

サーバーのホスト。デフォルトは localhost です。

`-http.port <port>`

サーバーのポート。デフォルトは 80 です。

`-type <type>`

出力するオブジェクトのタイプ。次のいずれかが指定できます。
node|interface|incident|ip|subnet|rrp|vlan|nodeSensor|interfaceAggregation|card|l2connection|physcomp|physSensor

`-legacy [long]`

`-legacy` オプションを指定すると，データをテキスト形式で出力します。このオプションを指定しない場合，xml 形式で出力されます。このオプションは，次のタイプ値とだけ使用できます。
node, interface, ip, l2connection および interfaceAggregation。

`-filter <filter>`

出力をプロパティで絞り込みます。`nnmtopodump.ovpl` コマンドは，次に示すフィルターをサポートします。

node - node.name | node.shortname | node.id | node.uuid | node.status | node.snmpaddress | node.managementMode | node.deviceCategory | node.deviceDescription | node.deviceFamily | node.deviceVendor

interface - node.name | node.shortname | node.id | node.snmpaddress | node.managementMode | node.deviceCategory | node.deviceDescription | node.deviceFamily | node.deviceVendor | interface.ifType | interface.id | interface.uuid | interface.managementMode | interface.managementState

ip - interface.id | node.id | ip.value | ip.id

vlan - node.name | node.id | vlan.id | vlan.name | vlan.value

nodeSensor - node.name | node.hostname | node.id | nodeSensor.id |nodeSensor.name | nodeSensor.type

interfaceAggregation - master.id | master.uuid | master.index | master.alias |slave.id | slave.uuid | slave.index | slave.alias

card - node.name | node.hostname | node.uuid | comp.uuid | comp.name

l2connection - connection.name | connection.id | connection.uuid

physcomp - node.name | node.hostname | node.uuid | comp.uuid | comp.name | comp.type

physSensor - node.name | node.hostname | node.id | physcomp.name |physcomp.id | physSensor.id | physSensor.name | physSensor.type

EXAMPLES

`nnmtopodump.ovpl -u username -p password -type node`

トポロジデータベースにあるすべてのノードの情報を xml 形式で表示します。（NNMi 管理者のユーザー名とパスワードを指定する必要があります。）

`nnmtopodump.ovpl -u username -p password -legacy long -type node`

ノードをテキスト形式で出力する場合，`legacy` オプションを使用する必要があります。すべてのノードとそのインタフェースの情報を表示するには，`-legacy` と `-type node` オプションを使用します。

NNM 6.x/7.x での同等コマンド：`ovtopodump -l`

`nnmtopodump.ovpl -u username -p password -type node -filter node.name=foo.microfocus.com`

ノード foo.microfocus.com の情報を xml 形式で出力します。

`nnmtopodump.ovpl -u username -p password -legacy long -type node -filter node.name=foo.microfocus.com`

ノード foo.microfocus.com の情報をテキスト形式で出力します。ノードとそのインタフェースの情報を表示するには，`-legacy` と `-type node` オプションを使用します。

NNM 6.x/7.x での同等コマンド：`ovtopodump -lr foo.microfocus.com`

`nnmtopodump.ovpl -u username -p password -legacy long -type node -filter node.id=2345`

ノードの id が 2345 のノードの情報をテキスト形式で出力します。NNMi はノードのすべてのインタフェースの情報も表示します。

NNM 6.x/7.x での同等コマンド：`ovtopodump -lr 2345`

`nnmtopodump.ovpl -u username -p password -type interface -filter interface.managementState=MANAGED`

トポロジデータベースにあるすべての管理されているインタフェースの情報を xml 形式で出力します。フィルターには，MANAGED，NOTMANAGED および OUTOFSERVICE が指定できます。

FILTER

`nnmtopodump.ovpl` コマンドは以下のフィルターをサポートします。

-type node

node.name - ノードのホスト名。

node.shortname - ノード名。

node.id - ノードの ID。

node.uuid - ノードの UUID。

node.status - ノードのステータス。フィルターの値は NORMAL，WARNING，MINOR，MAJOR および CRITICAL が使用できます。

node.snmpaddress - ノードの管理アドレス。

node.managementMode - ノードのノード管理モード。フィルターの値は MANAGED，NOTMANAGED および OUTOFSERVICE が使用できます。

node.deviceCategory - ノードのデバイスのプロファイルのカテゴリ。

node.deviceDescription - ノードのデバイスのプロファイルの説明。

node.deviceFamily - ノードのデバイスのプロファイルのファミリ。

node.deviceVendor - ノードのデバイスのプロファイルのベンダー。

-type interface

node.name - ホスト元ノードのホスト名。

node.shortname - ホスト元ノードのノード名。

node.id - ホスト元ノードの ID。

node.snmpaddress - ホスト元ノードの管理アドレス。

node.managementMode - ホスト元ノードのノード管理モード。フィルターの値は MANAGED, NOTMANAGED および OUTOFSERVICE が使用できます。

node.deviceCategory - ホスト元ノードのデバイスのプロファイルのカテゴリ。

node.deviceDescription - ホスト元ノードのデバイスのプロファイルの説明。

node.deviceFamily - ホスト元ノードのデバイスのプロファイルのファミリ。

node.deviceVendor - ホスト元ノードのデバイスのプロファイルのベンダー。

interface.ifType - インターフェイスの ifType。

interface.id - インターフェイスの ID。

interface.uuid - インターフェイスの UUID。

interface.managementMode - インターフェイスの直接管理モード。フィルターの値は INHERITED, NOTMANAGED および OUTOFSERVICE が使用できます。

interface.managementState - インターフェイスの管理モード。フィルターの値は MANAGED, NOTMANAGED および OUTOFSERVICE が使用できます。

-type ip

interface.id - インターフェイスの ID。

node.id - ホスト元ノードの ID。

ip.value - IP アドレスのアドレス。

ip.id - IP アドレスの ID。

-type vlan

node.name - ポートのホスト元ノードのホスト名。

node.id - ポートのホスト元ノードの ID。

vlan.id - VLAN のデータベース上での ID。

vlan.name - VLAN の名前。

vlan.value - VLAN の VLAN ID。

-type nodeSensor

node.name - ホスト元ノードのノード名。

node.hostname - ホスト元ノードのホスト名。

node.id - ホスト元ノードの ID。

nodeSensor.id - ノードセンサーの ID。

nodeSensor.name - ノードセンサーの名前。

nodeSensor.type - ノードセンサーのタイプ。フィルターの値は com.hp.nnm.sensor.CPU, com.hp.nnm.sensor.MEMORY，com.hp.nnm.sensor.BUFFERS，com.hp.nnm.sensor.DISK, com.hp.nnm.sensor.BGP_PEER および com.hp.nnm.sensor.WLAN が使用できます。

-type interfaceAggregation

master.id - 集約インターフェイスの ID。

master.uuid - 集約インターフェイスの UUID。

master.index - 集約インターフェイスの ifIndex。

master.alias - 集約インターフェイスの ifAlias。

slave.id - 集約メンバーインターフェイスの ID。

slave.uuid - 集約メンバーインターフェイスの UUID。

slave.index - 集約メンバーインターフェイスの ifIndex。

slave.alias - 集約メンバーインターフェイスの ifAlias。

-type card

node.name - 管理ノードのノード名。

node.hostname - 管理ノードのホスト名。

node.uuid - 管理ノードの UUID。

comp.uuid - カードの UUID。

comp.name - カードの名前。

-type l2connection

connection.name - レイヤー 2 の接続の名前。

connection.id - レイヤー 2 の接続の ID。

connection.uuid - レイヤー 2 の接続の UUID。

-type physcomp

node.name - 管理ノードのノード名。

node.hostname - 管理ノードのホスト名。

node.uuid - 管理ノードの UUID。

comp.uuid - 物理コンポーネントの UUID。

comp.name - 物理コンポーネントの名前。

comp.type - 物理コンポーネントのタイプ。フィルターの値は com.hp.nnm.CHASSIS, com.hp.nnm.CARD，com.hp.nnm.BACKPLANE，com.hp.nnm.POWER，com.hp.nnm.FAN, com.hp.nnm.SENSOR および com.hp.nnm.CPU が使用できます。

-type physSensor

node.name - 管理ノードのノード名。

node.hostname - 管理ノードのホスト名。

node.id - 管理ノードの ID。

physComp.id - コンポーネントの ID。

physComp.name - コンポーネントの名前。

physSensor.id - 物理センサーの ID。

physSensor.name - 物理センサーの名前。

physSensor.type - 物理センサーのタイプ。フィルターの値は com.hp.nnm.physSensor.FAN, com.hp.nnm.physSensor.POWER_SUPPLY, com.hp.nnm.physSensor.TEMPERATURE, com.hp.nnm.physSensor.VOLTAGE, com.hp.nnm.physSensor.BACKPLANE および com.hp.nnm.physSensor.RADIO が使用できます。

OUTPUT FIELDS

-type node

/topo/node/id - ノードの ID。

/topo/node/uuid - ノードの UUID。

/topo/node/shortname - ノード名。

/topo/node/name - ノードのホスト名。

/topo/node/description - ノードのシステムの説明。

/topo/node/status - ノードのステータス。

/topo/node/managementMode - ノードのノード管理モード。

/topo/node/contact - ノードのシステムの連絡先。

/topo/node/location - ノードのシステムのロケーション。

/topo/node/nodesnmpsysname - ノードのシステムの名前。

/topo/node/nodesnmpaddress - ノードの管理アドレス。

/topo/node/systemObjectId - ノードのシステムのオブジェクト ID。

/topo/node/notes - ノードの注。

/topo/node/nodecreatetime - ノードの作成日時。

/topo/node/nodemodifiedtime - ノードの最終変更日時。

/topo/node/nodelaststatuschange - ノードのステータスの最終変更日時。

/topo/node/protocolversion - このフィールドは非サポートです。

/topo/node/snmpAgent - このフィールドは非サポートです。

/topo/node/discoveryState - ノードの検出状態。

/topo/node/deviceDescription - ノードのデバイスのプロファイルの説明。

/topo/node/deviceCategory - ノードのデバイスのプロファイルのカテゴリ。

/topo/node/deviceFamily - ノードのデバイスのプロファイルのファミリー。

/topo/node/deviceVendor - ノードのデバイスのプロファイルのベンダー。

/topo/node/capabilities - ノードのケーパビリティの一覧。

/topo/node/capabilities/capability - ノードのケーパビリティ。

/topo/node/extendedAttributes - ノードのカスタム属性の一覧。

/topo/node/extendedAttributes/attribute - ノードのカスタム属性。

/topo/node/extendedAttributes/attribute/name - カスタム属性の名前。

/topo/node/extendedAttributes/attribute/value - カスタム属性の値。

/topo/node/hostedOnId - ノードのホスト元ノードの ID。 このフィールドは仮想インスタンスか仮想マシンでのみ表示されます。

/topo/node/hostedOnName - ノードホスト元ノードのノード名。 このフィールドは仮想インスタンスか仮想マシンでのみ表示されます。

/topo/node/hostedOnHostname - ノードホスト元ノードのホスト名。 このフィールドは仮想インスタンスか仮想マシンでのみ表示されます。

/topo/node/powerState - ノードの電源状態。 このフィールドは VMWare 仮想マシンでのみ表示されます。

/topo/node/lastStateChange - ノードの状態の最終変更日時。 このフィールドは VMWare 仮想マシンでのみ表示されます。

-type node -legacy long

ホスト名： - ノードのホスト名。

短縮名： - ノード名。

ノードID： - ノードの ID。

ノードUUID： - ノードの UUID。

作成時間： - ノードの作成日時。

修正時間： - ノードの最終変更日時。

ステータス： - ノードのステータス。

管理モード： - ノード管理モード。

再度のステータス変更： - ノードのステータスの最終変更日時。

説明： - ノードのシステムの説明。

場所： - ノードのシステムのロケーション。

連絡先： - ノードのシステムの連絡先。

SNMP SYSNAME：- ノードのシステムの名前。

SNMP オブジェクトID： - ノードのシステムのオブジェクト ID。

サポートされているSNMPバージョン： - このフィールドは非サポートです。

SNMPアドレス： - ノードの管理アドレス。

ノードの説明： - ノードのデバイスのプロファイルの説明。

ノードカテゴリ： - ノードのデバイスのプロファイルのカテゴリ。

ノードファミリ： - ノードのデバイスのプロファイルのファミリー。

ノードベンダー： - ノードのデバイスのプロファイルのベンダー。

ノードエージェント： - このフィールドは非サポートです。

ルートグループ/ポート： - このフィールドは非サポートです。

ノードラベル： - ノードのホスト名。

ケーパビリティ： - ノードのケーパビリティ。

CUSTOM-ATTRIBUTE： - ノードのカスタム属性。

ノードのホスト元ID： - ノードのホスト元ノードの ID。

ノードのホスト元名： - ノードのホスト元ノードのノード名。

ノードのホスト元ホスト名： - ノードのホスト元ノードのホスト名。

インターフェイスの数： - ノードのインターフェイスの数。

-type interface

/topo/interface/id - インターフェイスの ID。

/topo/interface/uuid - インターフェイスの UUID。

/topo/interface/ifName - インターフェイスの ifName。

/topo/interface/ifAlias - インターフェイスの ifAlias。

/topo/interface/ifType - インターフェイスの ifType。

/topo/interface/ifIndex - インターフェイスの ifIndex。

/topo/interface/ifDescr - インターフェイスの ifDescr。

/topo/interface/physicalAddress - インターフェイスの物理アドレス。

/topo/interface/cdp - このフィールドは非サポートです。

/topo/interface/speed - インターフェイスの ifSpeed。

/topo/interface/ifcreatetime - インターフェイスの作成日時。

/topo/interface/ifmodtime - インターフェイスの最終変更日時。

/topo/interface/status - インターフェイスのステータス。

/topo/interface/managementMode - インターフェイスの直接管理モード。

/topo/interface/managementState - インターフェイスの管理モード。

/topo/interface/hostedOnId - インターフェイスのホスト元ノードの ID。

/topo/interface/hostedOnName - インターフェイスのホスト元ノードのホスト名。

/topo/interface/capabilities - インターフェイスのケーパビリティの一覧。

/topo/interface/capabilities/capability - インターフェイスのケーパビリティ。

/topo/interface/extendedAttributes - インターフェイスのカスタム属性の一覧。

/topo/interface/extendedAttributes/attribute -インターフェイスのカスタム属性。

/topo/interface/extendedAttributes/attribute/name - カスタム属性の名前。

/topo/interface/extendedAttributes/attribute/value - カスタム属性の値。

-type interface -legacy long

インターフェイス名： - インターフェイスの ifName。

インターフェイスの説明： - インターフェイスの ifDescr。

IF_ALIAS： - インターフェイスの ifAlias。

インターフェイスID： - インターフェイスの ID。

インターフェイスUUID： - インターフェイスの UUID。

作成時間： - インターフェイスの作成日時。

修正時間： - インターフェイスの最終変更日時。

ステータス： - インターフェイスのステータス。

管理モード： - インターフェイスの直接管理モード。

管理状態： - インターフェイスの管理モード。

IF 番号： - インターフェイスの ifIndex。

IF タイプ： - インターフェイスの ifType。

物理アドレス： - インターフェイスの物理アドレス。

ノードID： - インターフェイスのホスト元ノードの ID。

CDP： - このフィールドは非サポートです。

速度： - インターフェイスの ifSpeed。

ノード名： - インターフェイスのホスト元ノードのホスト名。

ケーパビリティ： - インターフェイスのケーパビリティ。

CUSTOM-ATTRIBUTE： - インターフェイスのカスタム属性。

-type incident

/topo/incident/id - インシデントの ID。

/topo/incident/uuid - インシデントの UUID。

/topo/incident/name - インシデントの名前。

/topo/incident/severity - インシデントの重大度。

/topo/incident/formattedMessage - インシデントのメッセージ。

/topo/incident/sourceName - インシデントのソースオブジェクトの名前。

/topo/incident/sourceType - インシデントのソースオブジェクトのタイプ。

/topo/incident/sourceUuid - インシデントのソースオブジェクトの UUID。

/topo/incident/nodeName - インシデントのソースノードの名前。

/topo/incident/nodeUuid - インシデントのソースノードの UUID。

/topo/incident/nature - インシデントの相関特性。

/topo/incident/assignedTo - インシデントの割り当て先。

/topo/incident/origin - インシデントの発生元。

/topo/incident/originOccurrenceTime - インシデントの元の発生日時。

/topo/incident/firstOccurrenceTime - インシデントの最初の発生日時。

/topo/incident/lastOccurrenceTime - インシデントの最後の発生日時。

/topo/incident/notes - インシデントの相関の注。

/topo/incident/rcaActive - インシデントの RCA アクティブ。

/topo/incident/duplicateCount - インシデントの重複数。

/topo/incident/lifecycleState - インシデントのライフサイクル状態。

/topo/incident/priority - インシデントの優先度。

/topo/incident/category - インシデントのカテゴリ。

/topo/incident/family - インシデントのファミリ。

/topo/incident/cias - インシデントのカスタムインシデント属性の一覧。

/topo/incident/cias/cia - インシデントのカスタムインシデント属性。

/topo/incident/cias/cia/id - カスタムインシデント属性の ID。

/topo/incident/cias/cia/name - カスタムインシデント属性の名前。

/topo/incident/cias/cia/type - カスタムインシデント属性のタイプ。

/topo/incident/cias/cia/value - カスタムインシデント属性の値。

-type ip

/topo/ip/id - IP アドレスの ID。

/topo/ip/uuid - IP アドレスの UUID。

/topo/ip/name - IP アドレス。

/topo/ip/status - IP アドレスのステータス。

/topo/ip/value - IP アドレス。

/topo/ip/prefixLength - IP アドレスのプレフィックス長。

/topo/ip/interface - IP アドレスのインターフェイスの ID。

/topo/ip/node - IP アドレスのホスト元ノードの ID。

/topo/ip/subnet - IP アドレスのサブネットの ID。

/topo/ip/extendedAttributes - このフィールドは非サポートです。

-type ip -legacy long

IPAddress ID： - IP アドレスの ID。

UUID： - IP アドレスの UUID。

名前： - IP アドレスのアドレス。

ステータス： - IP アドレスのステータス。

IP アドレス： - IP アドレスのアドレス。

プレフィックス長： - IP アドレスのプレフィックス長。

インターフェイス ID： - IP アドレスのインターフェイスの ID。

ノード ID： - IP アドレスのホスト元ノードの ID。

サブネット ID： - IP アドレスのサブネットの ID。

-type subnet

/topo/subnet/id - IP サブネットの ID。

/topo/subnet/uuid - IP サブネットの UUID。

/topo/subnet/name - IP サブネットの名前。

/topo/subnet/status - このフィールドは非サポートです。

/topo/subnet/prefix - IP サブネットのプレフィックス。

/topo/subnet/prefixLength - IP サブネットのプレフィックス長。

/topo/subnet/extendedAttributes - このフィールドは非サポートです。

-type rrp

/topo/routerRedundancyGroup/id - ルーター冗長グループの ID。

/topo/routerRedundancyGroup/name - ルーター冗長グループの名前。

/topo/routerRedundancyGroup/groupNumber - ルーター冗長グループのグループ番号。

/topo/routerRedundancyGroup/protocol - ルーター冗長グループのプロトコル。

/topo/routerRedundancyGroup/extendedAttributes - このフィールドは非サポートです。

/topo/routerRedundancyGroup/protectedIPs - ルーター冗長グループの仮想 IP アドレスの一覧。

/topo/routerRedundancyGroup/protectedIPs/value - ルーター冗長グループの仮想 IP アドレスの値

/topo/routerRedundancyGroup/protectedIPs/extendedAttributes - このフィールドは非サポートです。

/topo/routerRedundancyGroup/status - ルーター冗長グループのステータス。

/topo/routerRedundancyGroup/members - ルーター冗長グループのルーター冗長メンバーの一覧。

/topo/routerRedundancyGroup/members/member - ルーター冗長グループのルーター冗長メンバー。

/topo/routerRedundancyGroup/members/member/id - ルーター冗長メンバーの ID

/topo/routerRedundancyGroup/members/member/name - ルーター冗長メンバーの名前。

/topo/routerRedundancyGroup/members/member/isOwner - ルーター冗長メンバーが所有者であるかどうか。

/topo/routerRedundancyGroup/members/member/hostedOnId - ルーター冗長メンバーのホスト元ノードの ID。

/topo/routerRedundancyGroup/members/member/hostedOnName - ルーター冗長メンバーのホスト元ノードの名前

/topo/routerRedundancyGroup/members/member/redundancyInterfaceId - ルーター冗長メンバーの冗長インターフェイスの ID。

/topo/routerRedundancyGroup/members/member/currentState - ルーター冗長メンバーの現在の状態。

/topo/routerRedundancyGroup/members/member/previousState - ルーター冗長メンバーの以前の状態。

/topo/routerRedundancyGroup/members/member/priority - ルーター冗長メンバーの優先度。

/topo/routerRedundancyGroup/members/member/extendedAttributes - このフィールドは非サポートです。

/topo/routerRedundancyGroup/members/member/trackedObjects - ルーター冗長メンバーの追跡対象オブジェクトの一覧。

/topo/routerRedundancyGroup/members/member/trackedObjects/trackedObject - ルーター冗長メンバーの追跡対象オブジェクト。

/topo/routerRedundancyGroup/members/member/trackedObjects/trackedObject/id - 追跡対象オブジェクトの ID。

/topo/routerRedundancyGroup/members/member/trackedObjects/trackedObject/type - 追跡対象オブジェクトのタイプ。

/topo/routerRedundancyGroup/members/member/trackedObjects/trackedObject/objectUuid - 追跡対象オブジェクトの UUID。

/topo/routerRedundancyGroup/members/member/trackedObjects/trackedObject/hostedOnId - 追跡対象オブジェクトのホスト元ノードの ID。

/topo/routerRedundancyGroup/members/member/trackedObjects/trackedObject/hostedOnName - 追跡対象オブジェクトのホスト元ノードの名前。

/topo/routerRedundancyGroup/members/member/trackedObjects/trackedObject/trackPriority - 追跡対象オブジェクトの追跡の優先度。

/topo/routerRedundancyGroup/members/member/trackedObjects/trackedObject/extendedAttributes - このフィールドは非サポートです。

-type vlan

/topo/vlan/vlanId - VLAN のデータベース上での ID。

/topo/vlan/vlanValue - VLAN の VLAN ID。

/topo/vlan/vlanName - VLAN の名前。

/topo/vlan/portId - VLAN のポートの ID。

/topo/vlan/portName - VLAN のポートの名前。

/topo/vlan/nodeId - VLAN のポートのホスト元ノードの ID。

/topo/vlan/nodeName - VLAN のポートのホスト元ノードのホスト名。

-type nodeSensor

/topo/nodeSensor/id - ノードセンサーの ID。

/topo/nodeSensor/uuid - ノードセンサーの UUID。

/topo/nodeSensor/name - ノードセンサーの名前。

/topo/nodeSensor/type - ノードセンサーのタイプ。

/topo/nodeSensor/node - ノードセンサーのホスト元ノード。

/topo/nodeSensor/node/id - ホスト元ノードの ID。

/topo/nodeSensor/node/uuid - ホスト元ノードの UUID。

/topo/nodeSensor/node/name - ホスト元ノードのノード名。

/topo/nodeSensor/node/hostname - ホスト元ノードのホスト名。

/topo/nodeSensor/monitoredAttributes - ノードセンサーの監視対象属性の一覧。

/topo/nodeSensor/monitoredAttributes/attribute - ノードセンサーの監視対象属性。

/topo/nodeSensor/monitoredAttributes/attribute/id - 監視対象属性の ID。

/topo/nodeSensor/monitoredAttributes/attribute/name - 監視対象属性の名前。

/topo/nodeSensor/monitoredAttributes/attribute/state - 監視対象属性の状態。

/topo/nodeSensor/monitoredAttributes/attribute/modified - 監視対象属性の最終変更日時。

/topo/nodeSensor/extendedAttributes - このフィールドは非サポートです。

-type interfaceAggregation

/topo/interfaceAggregation - 集約インターフェイスと集約メンバーインターフェイスの関連。

/topo/interfaceAggregation/master - 集約インターフェイス。

/topo/interfaceAggregation/master/id - 集約インターフェイスの ID。

/topo/interfaceAggregation/master/uuid - 集約インターフェイスの UUID。

/topo/interfaceAggregation/master/index - 集約インターフェイスの ifIndex。

/topo/interfaceAggregation/master/alias - 集約インターフェイスの ifAlias。

/topo/interfaceAggregation/slave - 集約メンバーインターフェイス。

/topo/interfaceAggregation/slave/id - 集約メンバーインターフェイスの ID。

/topo/interfaceAggregation/slave/uuid - 集約メンバーインターフェイスの UUID。

/topo/interfaceAggregation/slave/index - 集約メンバーインターフェイスの ifIndex。

/topo/interfaceAggregation/slave/alias - 集約メンバーインターフェイスの ifAlias。

-type interfaceAggregation -legacy long

マスター - 集約インターフェイス。

ID： - 集約インターフェイスの ID。

UUID： - 集約インターフェイスの UUID。

インデックス： - 集約インターフェイスの ifIndex。

エイリアス： - 集約インターフェイスの ifAlias。

スレーブ - 集約メンバーインターフェイス。

ID： - 集約メンバーインターフェイスの ID。

UUID： - 集約メンバーインターフェイスの UUID。

インデックス： - 集約メンバーインターフェイスの ifIndex。

エイリアス： - 集約メンバーインターフェイスの ifAlias。

-type l2connection

/topo/l2connection/id - レイヤー 2 の接続の ID。

/topo/l2connection/name - レイヤー 2 の接続の名前。

/topo/l2connection/status - レイヤー 2 の接続のステータス。

/topo/l2connection/source - レイヤー 2 の接続のトポロジソース。

/topo/l2connection/uuid - レイヤー 2 の接続の UUID。

-type l2connection -legacy long

L2Connection ID： - レイヤー 2 の接続の ID。

L2Connection 名： - レイヤー 2 の接続の名前。

L2Connection ステータス： - レイヤー 2 の接続のステータス。

L2Connection ソース： - レイヤー 2 の接続のトポロジソース。

L2Connection UUID： - レイヤー 2 の接続の UUID。

-type physcomp , -type card

/topo/physcomp/uuid - 物理コンポーネントの UUID。

/topo/physcomp/name - 物理コンポーネントの名前。

/topo/physcomp/type - 物理コンポーネントのタイプ。

/topo/physcomp/index - 物理コンポーネントのコンポーネント ID。

/topo/physcomp/slots - このフィールドは非サポートです。

/topo/physcomp/modelName - 物理コンポーネントのモデル名。

/topo/physcomp/modelType - 物理コンポーネントのモデルタイプ。

/topo/physcomp/serialNumber - 物理コンポーネントのシリアル番号。

/topo/physcomp/firmwareVersion - 物理コンポーネントのファームウェアバージョン。

/topo/physcomp/hardwareVersion - 物理コンポーネントのハードウェアバージョン。

/topo/physcomp/softwareVersion - 物理コンポーネントのソフトウェアバージョン。

/topo/physcomp/description - 物理コンポーネントの説明。

/topo/physcomp/parent - 物理コンポーネントの親コンポーネント。

/topo/physcomp/capabilities - 物理コンポーネントのケーパビリティの一覧。

/topo/physcomp/capabilities/capability - 物理コンポーネントのケーパビリティ。

/topo/physcomp/extendedAttributes - 物理コンポーネントのカスタム属性の一覧。

/topo/physcomp/extendedAttributes/attribute - 物理コンポーネントのカスタム属性。

/topo/physcomp/extendedAttributes/attribute/name - カスタム属性の名前。

/topo/physcomp/extendedAttributes/attribute/value - カスタム属性の値。

/topo/physcomp/node - 物理コンポーネントの管理ノード。

/topo/physcomp/node/uuid - 管理ノードの UUID。

/topo/physcomp/node/name - 管理ノードのノード名。

/topo/physcomp/node/hostname - 管理ノードのホスト名。

/topo/physcomp/hostedNodes - 物理コンポーネントのホスト対象ノードの一覧。

/topo/physcomp/hostedNodes/hostedNode - 物理コンポーネントのホスト対象ノード。

/topo/physcomp/hostedNodes/hostedNode/id ホスト対象ノードの ID

/topo/physcomp/hostedNodes/hostedNode/name ホスト対象ノードのノード名

/topo/physcomp/hostedNodes/hostedNode/hostname ホスト対象ノードのホスト名

/topo/physcomp/monitoredAttributes - 物理コンポーネントの状態。

/topo/physcomp/monitoredAttributes/administrativeState - 物理コンポーネントの管理状態。

/topo/physcomp/monitoredAttributes/operationalState - 物理コンポーネントの運用状態。

/topo/physcomp/monitoredAttributes/standByState - 物理コンポーネントのスタンバイ状態。

/topo/physcomp/monitoredAttributes/previousStandByState - 物理コンポーネントの以前のスタンバイ状態。

/topo/physcomp/monitoredAttributes/modified - 物理コンポーネントの状態の最終変更日時。

/topo/physcomp/redundancyGroup - 物理コンポーネントの冗長グループ。

/topo/physcomp/redundancyGroup/uuid - 物理コンポーネントの冗長グループの UUID。

/topo/physcomp/redundancyGroup/name - 物理コンポーネントの冗長グループの名前。

/topo/physcomp/status - 物理コンポーネントのステータス。

/topo/physcomp/status/timestamp - 物理コンポーネントのステータスの最終変更日時。

/topo/physcomp/status/value - 物理コンポーネントのステータスの値。

/topo/physcomp/status/conclusion - 物理コンポーネントの結果。

/topo/physcomp/status/conclusion/name - 物理コンポーネントの結果の名前。

/topo/physcomp/status/conclusion/status - 物理コンポーネントの結果のステータス。

/topo/physcomp/status/conclusion/timestamp - 物理コンポーネントの結果のタイムスタンプ。

/topo/physcomp/optStrings - このフィールドは非サポートです。

-type physSensor

/topo/physSensor/id - 物理センサーの ID。

/topo/physSensor/uuid - 物理センサーの UUID。

/topo/physSensor/name - 物理センサーの名前。

/topo/physSensor/type - 物理センサーのタイプ。

/topo/physSensor/physComp - 物理センサーのコンポーネント。

/topo/physSensor/physComp/id - 物理センサーのコンポーネントの ID。

/topo/physSensor/physComp/uuid - 物理センサーのコンポーネントの UUID。

/topo/physSensor/physComp/name - 物理センサーのコンポーネントの名前。

/topo/physSensor/node - 物理センサーの管理ノード。

/topo/physSensor/node/id - 物理センサーの管理ノードの ID。

/topo/physSensor/node/uuid - 物理センサーの管理ノードの UUID。

/topo/physSensor/node/name - 物理センサーの管理ノードの名前。

/topo/physSensor/node/hostname - 物理センサーの管理ノードのホスト名。

/topo/physSensor/monitoredAttributes - 物理センサーの監視対象属性の一覧。

/topo/physSensor/monitoredAttributes/attribute - 物理センサーの監視対象属性。

/topo/physSensor/monitoredAttributes/attribute/id - 監視対象属性の ID。

/topo/physSensor/monitoredAttributes/attribute/name - 監視対象属性の名前。

/topo/physSensor/monitoredAttributes/attribute/state - 監視対象属性の状態。

/topo/physSensor/monitoredAttributes/attribute/modified - 監視対象属性の最終変更日時。

/topo/physSensor/extendedAttributes - このフィールドは非サポートです。

AUTHOR

nnmtopodump.ovpl was developed by Micro Focus.

SEE ALSO

nnmnodedelete.ovpl, nnm.properties.

付録 G.57　nnmtopoquery.ovpl

NNMi トポロジ上のクエリーを実行します。

SYNOPSIS

`nnmtopoquery.ovpl findConnectedNeighborInterfacesByNode -node (<name>|<uuid>)`

`nnmtopoquery.ovpl findL2ConnectionsByNode -node (<name>|<uuid>)`

`nnmtopoquery.ovpl findWebAgentSettingsByNode -node (<name>|<uuid>)`

`nnmtopoquery.ovpl listWebAgentSettings`

DESCRIPTION

nnmtopoquery.ovpl は NNMi 管理サーバに対してクエリを実行して，選択可能な形式で結果を出力します。

Parameters

nnmtopoquery.ovpl 以下のオプションを認識します。

-h

使用方法を出力します。

-node (<name>|<uuid>)

入力として受け入れるクエリのノードを指定します。ノードはノード名，ホスト名または UUID によって参照することができます。

-format (csv|list|text|xml)

出力形式を変更します。可能な出力モードは，CSV，LIST，TEXT および XML です。

-fields <comma_separated_fields>

CSV ファイルで出力フィールドを選択します。この引数が提供されない場合，すべてのフィールドがデフォルトで出力されます。 この引数が提供された場合，フィールドが指定した順序で出力されます。

-u <username> -p <password>

スクリプトを実行するための資格情報を指定します。このスクリプトは，nnm.properties ファイルを使用しない限り，有効な NNMi 資格情報が必要です。詳細については，nnm.properties のリファレンスページを参照してください。

Listing connected neighbor interfaces for a node

findConnectedNeighborInterfacesByNode クエリーを使用して，特定のノードに接続されたすべての隣接インタフェースをリストします。

このクエリーで利用可能な出力フィールドは，次のとおりです。

uuid，name，NodeUUID，NodeName，ifIndex, ifAlias, speed, inSpeed, outSpeed, ifDescr, ifName, ifType, physicalAddress および managementMode

Example

```
nnmtopoquery.ovpl findConnectedNeighborInterfacesByNode -node mynode -fields
nodeName,name,physicalAddress,managementMode
```

Listing Layer 2 Connections for a Node

findL2ConnectionsByNode クエリーを使用して，特定のノードのすべてのレイヤー 2 接続をリストします。

このクエリーで利用可能な出力フィールドは，次のとおりです。

connUUID, connName, connStatus, connStatusTimestamp, localNodeUUID, localNodeName, localIfUUID, localIfIndex, localIfAlias, localIfName, localIfDesc, localIfAddr, remoteNodeUUID, remoteNodeName, remoteIfUUID, remoteIfIndex, remoteIfAlias, remoteIfName, remoteIfDesc および remoteIfAddr

Example

```
nnmtopoquery.ovpl findL2ConnectionsByNode -node mynode -fields
connUUID,connName,localIfIndex,remoteIfIndex
```

Listing Web Agent Settings of a Node

findWebAgentSettingsByNode クエリーを使用して，特定のノードの Web Agent 設定をリストします。

このクエリーで利用可能な出力フィールドは，次のとおりです。

agentUuid, hostname, name, nodeUuid, nodeName, mode, agentEnabled, username, port, scheme, timeout, lastModifiedBy, lastModified, validTo および subjectDN

Example

```
nnmtopoquery.ovpl findWebAgentSettingsByNode -node mynode -fields
agentUuid,hostName,nodeUuid,nodeName,mode,userName
```

Listing All Web Agent Settings

listWebAgentSettings クエリーを使用して，すべての Web Agent 設定をリストします。

このクエリーで利用可能な出力フィールドは，次のとおりです。

agentUuid, hostName, name, nodeUuid, nodeName, mode, agentEnabled, userName, port, scheme, timeout, lastModifiedBy および lastModified

Example

```
nnmtopoquery.ovpl listWebAgentSettings -node mynode -fields
agentUuid,hostName,nodeUuid,nodeName,mode,userName
```

RETURN VALUE

nnmtopoquery.ovpl エラーが発生しなければステータス0(ゼロ)，そうでなければ 1 で終了します。

AUTHOR

nnmtopoquery.ovpl was developed by Micro Focus.

付録 G.58　nnmtrapconfig.ovpl

NNMi トラップサービスを構成します

SYNOPSIS

nnmtrapconfig.ovpl -u *<username>* -p *<password>* [-showProp] [-start] [-stop] [-readFilter] [-dumpBlockList] [-resetBlockCache]

nnmtrapconfig.ovpl -setProp -u *<username>* -p *<password>* [trapInterface *<ip_addr>*] [unsetTrapInterface] [trapPort *<port>*] [recvSocketBufSize *<size>*] [disallowV1V2] [allowV1V2] [loopbackAddrOverride *<ip_addr>*] [resetLoopbackAddrOverride] [blockTraps] [unblockTraps] [thresholdRate *<rate>*] [rearmRate *<rate>*] [overallThresholdRate *<rate>*] [overallRearmRate *<rate>*] [minTrapCount *<count>*] [databaseQSize *<count>*] [pipelineQSize *<count>*] [databaseFileSize *<size>*] [databaseFileCount *<count>*] [trapLoggingMode *<log_mode>*] [trapLoggingCompression *<boolean>*] [trapLoggingMaxFileSize *<size>*] [trapLoggingTaskInterval *<time>*] [trapLoggingBatchSize *<size>*] [trapReceiverNettyPort *<port>*] [trapReceiverJmsTTL *<time miliseconds>*] [-persist]

DESCRIPTION

nnmtrapconfig.ovpl を使用すると，トラップサービスの現在のプロパティを表示したり変更したりできます。さらに，トラップサービスの起動や停止もできます。nnmtrapconfig.ovpl は，フィルター構成ファイルの読み込み，現在のフィルター構成およびブロッキングキャッシュの印刷，ブロッキングキャッシュのリセットなどのフィルター関連機能も提供します。

Parameters

nnmtrapconfig.ovpl コマンドは，次のパラメータをサポートします。

-u *<username>*

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。 詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-showProp

トラップサービスに関連付けられたプロパティ，および現在のプロパティ値を示します。

-start

トラップサービスを起動します。

注意：このコマンドは，JBoss の中で実行されているトラップサーバーを起動します。スタンドアロンのトラップレシーバーの起動については，『*NNMi NmsTrapReceiver プロセス*』の章を参照してください。

-stop

トラップサービスを停止します。

注意：このコマンドは，JBoss の中で実行されているトラップサーバーを停止します。スタンドアロンのトラップレシーバーの停止については，『*NNMi NmsTrapReceiver プロセス*』の章を参照してください。

-readFilter

トラップサービスにフィルター構成ファイルを読み込ませます。

-dumpBlockList

トラップサービスが作成するフィルター構成およびブロッキングキャッシュを印刷します。

-resetBlockCache

トラップサービスのブロッキングキャッシュをリセットします。

一つ以上のプロパティの値を設定する場合は，nnmtrapconfig.ovpl に次のオプションを付けて実行してください。

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-setProp

トラップサービスに関連付けられたプロパティの値を設定します。

`trapInterface` *ip_addr*

トラップサービスがトラップをリッスンする IP アドレスを設定します。デフォルトでは，すべてのインタフェース上でリッスンします。

`unsetTrapInterface`

トラップサービスがすべてのインタフェース上でリッスンするよう設定します。

`trapPort` *port*

トラップサービスがトラップをリッスンするポートを設定します。デフォルトのポートは 162 です。

`recvSocketBufSize` *size*

トラップサービスがトラップをリッスンするソケットバッファのサイズをキロバイト単位で設定します。

`disallowV1V2`

受信する SNMPv1 および v2 トラップをすべてブロックし，v3 トラップのみを転送します。デフォルトで無効にされています。

`allowV1V2`

受信する SNMP トラップを v1 および v2 を含めすべて受け付けます。これは，デフォルトです。

`loopbackAddrOverride` *ip_addr*

ソースアドレスがループバックアドレスである場合に，トラップを転送する前にトラップのソースアドレスと交換する IP アドレスを設定します。

`resetLoopbackAddrOverride`

ループバックオーバーライドアドレスをリセットします。このオプションを実行すると，ユーザー指定のループバックオーバーライドアドレスは削除されます。この場合，NNMi はサーバーのアドレスの一つをループバックオーバーライドアドレスとして選択します。

`blockTraps`

フィルター構成およびしきい値構成に基づいてトラップをブロックします。

`unblockTraps`

トラップをブロックしません。

`thresholdRate` *rate*

トラップの送信元やトラップの OID をブロックするレートをトラップ/秒の形式で設定します。

`rearmRate` *rate*

ブロックされたトラップの送信元やトラップの OID のブロックを解除するレートをトラップ/秒の形式で設定します。このレートは，thresholdRate の値以下である必要があります。

`overallThresholdRate` *rate*

すべての受信トラップをブロックする割合を，1 秒あたりのトラップ数で設定します。

`overallRearmRate` *rate*

すべての受信トラップをブロック解除する割合を，1 秒あたりのトラップ数で設定します。この値は `overallThresholdRate` の値以下にしてください。

`minTrapCount`*count*

最低何個のトラップをソースから受信したらブロッキングを検討するのかを設定します。また，同じトラップを最低何個受信したらブロッキングを検討するのかも設定してください。

`databaseQSize` *count*

トラップをデータベースに書き込む待ち行列に保持できる最大トラップ数を設定します。

`pipelineQSize` *count*

トラップパイプラインの各ステージの待ち行列に保持できる最大トラップ数を設定します。

`databaseFileSize` *size*

トラップデータベース内の一つのファイルの最大サイズを MB 単位で設定します。ファイルサイズがこの値に到達すると，ロールオーバーが発生します。

`databaseFileCount` *count*

トラップデータベース内の最大ファイル数を設定します。

`trapLoggingMode` *log_mode*

トラップロガーの動作モードです。有効な値は，OFF，CSV，LOG，BOTH です。デフォルトは CSV です。

```
OFF:  すべてのトラップのロギングをオフにします。
CSV:  トラップは CSV 形式で記録されます。
LOG:  トラップは trapd.log に似たテキスト形式で記録されます。
BOTH: トラップは CSV 形式とテキスト形式の両方で記録されます。
```

`trapLoggingCompression`*boolean*

true の場合，トラップは gz 圧縮形式で書き込まれます。デフォルトは false です。

`trapLoggingMaxFileSize` *size*

トラップログファイルのファイルサイズが MB 単位の最大値に近づいた場合は，拡張子が.old のファイルにまとめてアーカイブされます。拡張子が .old のファイルは，各ログ形式に 1 ファイルのみ保持されます。デフォルトは 5MB です。

`trapLoggingTaskInterval` *time*

ファイルシステムにトラップを書き込む前にトラップロガーが待機する時間を秒単位で設定します。デフォルトは 2 秒です。

`trapLoggingBatchSize` *size*

各間隔の間にファイルシステムに書き込まれるトラップの最大数を設定します。デフォルトは 2048 です。

`trapReceiverNettyPort`*port*

TrapReceiver で JBoss からの接続を待機するポートを設定します。デフォルトは 5447 です。

`trapReceiverJmsTTL` *time miliseconds*

JMS メッセージの TTL をミリ秒単位で設定します。負の値は使用できません。これはトラップが TrapReceiver によってキャッシュされる時間です。JBoss がこの時間以上ダウンした場合，データは失われます。デフォルトは 5 分（300000）です。

`-persist`

将来の再起動にプロパティ値を使用できるように，現在のプロパティを維持します。

EXAMPLES

トラップサービスに関連付けられたプロパティ，およびプロパティ値を示します。

```
nnmtrapconfig.ovpl -u user -p pass -showProp
```

トラップサービスを起動します。

```
nnmtrapconfig.ovpl -u user -p pass -start
```

トラップポートを 1162 に設定します。

```
nnmtrapconfig.ovpl -u user -p pass -setProp trapPort 1162
```

ログは両方のフォーマット，最大ファイルサイズは 32MB，タスクの実行間隔は 30 秒，バッチサイズは 1024 の設定を使用してトラップを受信します。将来のトラップサービス起動用に現在の値も維持します。

```
nnmtrapconfig.ovpl -u user -p pass -setProp trapLoggingMode BOTH trapLoggingMaxFileSize 32
trapLoggingTaskInterval 30 trapLoggingBatchSize 1024 -persist
```

ブロッキングを有効にし，将来のトラップサービス起動用に現在の値を維持します。

```
nnmtrapconfig.ovpl -u user -p pass -setProp blockTraps -persist
```

将来のトラップサービス起動用に現在の値を維持します。

```
nnmtrapconfig.ovpl -u user -p pass -setProp -persist
```

FILES

NNMi トラップサービスプロパティは次に示すファイルに格納されています。

- Windows：`%NNM_PROPS%¥nnmtrapserver.properties`
- Linux：`$NNM_PROPS/nnmtrapserver.properties`

ブロッキングフィルターは次に示すファイルに設定できます。

- Windows：%NNM_SHARED_CONF%¥nnmtrapd.conf
- Linux：$NNM_SHARED_CONF/nnmtrapd.conf

AUTHOR

nnmtrapconfig.ovpl was developed by Micro Focus.

付録 G.59　nnmtrapdump.ovpl

バイナリトラップストアにロギングされたトラップをコンソールに出力します

SYNOPSIS

nnmtrapdump.ovpl [-t] [-from *date*] [-to *date*] [-source *IP address*] [-trapid *Trap OID*] [-last *minutes*] [-short] [-nodns] [-hexDump]

DESCRIPTION

すべての受信トラップは NNMi トラップサービスによってバイナリトラップストアにロギングされます。nnmtrapdump.ovpl は，ロギングされたトラップの確認に使用できます。新規受信トラップの監視にも使用できます。トラップツールを使用する場合，ログやエラーメッセージは標準出力には表示されません。標準エラー出力を別のファイルにリダイレクトすることで，実際のトラップダンプ出力とそれらのメッセージが混ざることを防止できます。

Parameters

nnmtrapdump.ovpl コマンドは，次のパラメータをサポートします。

-t

受信トラップを連続して出力する場合に使用します。このオプションは-from オプションと一緒に使用できます。

-from *date*

トラップ出力の開始日を指定します。日付は ISO 8601 標準形式：*yyyy-mm-ddThh:mm:ss[+ or -]hh:mm* で指定します。このオプションは-last オプションと一緒には使用できません。

-to *date*

トラップ出力の終了日を指定します。日付は ISO 8601 標準形式：*yyyy-mm-ddThh:mm:ss[+ or -]hh:mm* で指定します。

-source *IP address*

出力するトラップのソース IP アドレスを指定します。指定されたソースのトラップだけが出力されます。

-trapid *Trap OID*

出力するトラップのトラップ OID を指定します。指定された OID を持つトラップだけが出力されます。

-last *minutes*

出力するトラップの期間を指定します。値は分単位です。指定した分数をさかのぼった時点から現在までのトラップが出力されます。このオプションは-from オプションと一緒には使用できません。

-short

-short を使用すると，受信したトラップを短い形式で表示します。OID と到着時刻，送信元アドレスのみが表示されます。

-nodns

-nodns を使用すると，IP アドレスの名前解決を抑止します。これは，トラップ出力を高速化します。

-hexDump

-hexDump を使用すると，16 進形式でトラップを表示します。

EXAMPLES

バイナリトラップストアのすべてのトラップをコンソールに出力する場合

```
nnmtrapdump.ovpl
```

ループ状態で待って，すべての受信トラップを出力する場合

```
nnmtrapdump.ovpl -t
```

2008 年 7 月 31 日，午前 9:00 から午前 9:05（MDT）までの 5 分間枠のトラップを出力する場合

```
nnmtrapdump.ovpl -from 2008-07-31T09:00:00-06:00 -to 2008-07-31T09:05:00-06:00
```

過去 5 分間に受信したトラップを出力してから受信トラップを待つ場合

```
nnmtrapdump.ovpl -last 5 -t
```

IP アドレス 192.168.0.1 からのトラップを出力する場合

```
nnmtrapdump.ovpl -source 192.168.0.1
```

FILES

%NNM_DB%¥traps（Windows）または$NNM_DB/traps（Linux）は，トラップデータベースを構成するファイルが格納されたディレクトリです。

%NNM_LOG%¥nnm¥trapanalytics.log（Windows）または$NNM_LOG/nnm/trapanalytics.log（Linux）は，最もトラップ送信が多いトラップ ID とソースに関する情報を含む分析ログファイルです。

AUTHOR

nnmtrapdump.ovpl was developed by Micro Focus.

付録 G.60　nnmtrimincidents.ovpl

インシデントの削除および（必要に応じて）アーカイブへの保存

SYNOPSIS

nnmtrimincidents.ovpl [[[-age *age* -incr *incr*] | -date *date* | -trimOldest *numberToTrim*] [-nature *nature*] [-lifecycle *lifecycleState*] [-severity *severity*] [-origin *origin*] [-name *name*] [-family *family*] [-sysobjectid *sysobjectid*] [-path *path*] [-archiveOnly] [-trimOnly] [-trimAndArchive] [-batch *batchSize*] [-u *username*] [-p *password*] [-quiet]]

DESCRIPTION

nnmtrimincidents.ovpl は，インシデントをインシデントテーブルから削除するために使用します。 削除されたインシデントは，（必要に応じて）圧縮されたアーカイブファイルに保存されます。

- Windows：%NNM_DATA%¥tmp¥incidentArchive.ISO 8601 Date.Time Ms.csv.gz
- Linux：$NNM_DATA/tmp/incidentArchive.ISO 8601 Date.Time Ms.csv.gz

デフォルトの動作では，アーカイブせずにインシデントを削除します。

ARCHIVE-FORMAT

インシデントは，csv 形式を使用してアーカイブされます。カラム名は，次のように並べられます。

```
LastOcurranceTimeStamp,Name,SourceNodeName,SourceObjectName,SysObjectID,FormattedMessage,Life
CycleState,Severity,Priority,AssignedTo,JournalNotes,Category,Family,Nature,Origin,Incident
Notes,DuplicateCount,FirstOccuranceTimeStamp,OriginOccuranceTimeStamp,PayloadList,ElementOID

LastOcurranceTimeStamp   - このインシデントが最後に発生した日時の読解可能な文字列形式のタイ
ムスタンプ
Name             - インシデント名
SourceNodeName   - ソースノードの短縮名
SourceObjectName - ソースオブジェクト名
SysObjectID      - 常に空のこのフィールドは下位互換性のために維持されます
FormattedMessage - インシデントを説明するフォーマットされたメッセージ
LifeCycleState   - インシデントのライフサイクル状態
Severity         - インシデントの重大度
Priority         - インシデントの優先度をローカライズしたラベル
AssignedTo       - インシデントを割り当てられているアカウント
JournalNotes     - インシデントの注
Category         - インシデントのカテゴリをローカライズしたラベル
Family           - インシデントのファミリーをローカライズしたラベル
Nature           - インシデントの相関特性処理
Origin           - インシデントの発生元
IncidentNotes    - 相関処理の注
DuplicateCount   - システムでインシデントが発生した回数
FirstOccuranceTimeStamp  - 重複したインシデントの場合の最初の発生日時
OriginOccuranceTimeStamp - トラップ/syslog がシステムに到達したときのタイムスタンプ（該当す
る場合）
PayloadList: すべての CIA（”Name[Type]=Value” の形式で，複数の CIA がある場合は「|」で区切られます）
```

```
       Name      - CIA の名前（最大 50 文字）
       Type      - CIA のタイプ
       Value     - CIA の値（最大 2000 文字）
ElementOID       - インシデントの OID（存在する場合）
```

Parameters

-age *age*

削除するインシデントの時間を指定します。このオプションは*incr* オプションと一緒に使用してください。*age* には必ず0 より大きい値を指定してください。

-incr *incr*

age オプションの増加単位を指定します。サポートされている増加単位はhours（時間），days（日），weeks（週），months（月）です。

-trimOldest *numberToTrim*

データベースから削除するインシデントの数を指定します。指定したオプションに基づいて，最も古い*numberToTrim* インシデントがデータベース内のすべてのインシデントから選択されます。

-date *date*

日付を指定し，その日付より古いインシデントが削除されます。日付は ISO 8601 標準形式*yyyy-mm-ddThh:mm:ss[+ or -]hh:mm* で指定します。

-archiveOnly

アーカイブファイルを作成しますが，インシデントは削除しません。このオプションは*age* オプションまたは *date* オプションと共に指定する必要があります。

-trimOnly

インシデントを削除しますが，削除されたインシデントはアーカイブに保存しません。これはデフォルトの動作です。

-trimAndArchive

削除対象のインシデントをアーカイブして，削除します。

-batch *batchSize*

インシデントを削除するときのバッチサイズを指定します。0 より大きく，1000 以下の値が指定できます。

-path *path*

アーカイブファイル名を完全パスで指定します。このパスはデフォルトのアーカイブファイルより優先されます。

- Windows： %NNM_DATA%¥tmp¥incidentArchive.ISO 8601 Date.Time Ms.csv.gz
- Linux：$NNM_DATA/tmp/incidentArchive.ISO 8601 Date.Time Ms.csv.gz

-lifecycle *lifecycleState*

削除対象の*age* または*date* に一致するインシデントのライフサイクル状態を指定します。

lifecycleState の例としては次のものがあります。

Registered（登録済み）

InProgress（進行中）

Completed（完了）

Closed（解決済み）

-severity *severity*

削除対象の*age* または*date* に一致するインシデントの重大度を指定します。

severity の例としては次のものがあります。

Critical（危険域）

Major（重要警戒域）

Minor（警戒域）

Warning（注意域）

Normal（正常域）

-name *name*

削除対象の*age* または*date* に一致するインシデント名を指定します。

-family *family*

削除対象の*age* または*date* に一致するインシデントのファミリーの「一意のキー」を指定します。

-sysobjectid *sysobjectid*

削除対象の*age* または*date* に一致するインシデントのデバイスのシステムオブジェクト ID を指定します。

-nature *nature*

削除対象の*age* または*date* に一致するインシデントの相関処理特性を指定します。

nature の例としては次のものがあります。

RootCause（根本原因）

SecondaryRootCause（二次的な根本原因）

Symptom（症状）

ServiceImpact（サービスインパクト）

StreamCorrelation（ストリームの相関処理）

None（なし）

Info（情報）

Dedup_Stream_Correlation（重複削除ストリームの相関処理）

Rate_Stream_Correlation（レートストリームの相関処理）

-origin *origin*

削除対象の*age* または*date* に一致するインシデントの発生元を指定します。

origin の例としては次のものがあります。

ManagementSoftware（NNMi）

ManuallyCreated（手動作成）

RemotelyGenerated（遠隔生成）

SnmpTrap（SNMP トラップ）

Syslog（システムログ）

Other（その他）

-u <*username*>

コマンドの実行に必要な NNMi の管理者名を指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-p <*password*>

コマンドの実行に必要な NNMi の管理者パスワードを指定します。nnm.properties ファイルが存在していない限り，これが必要になります。詳細は nnm.properties リファレンスページを参照してください。

-quiet

非プロンプトモードを指定します。

EXAMPLES

6 日前より古いインシデントを削除します。

```
nnmtrimincidents.ovpl -age 6 -incr days
```

6 週間より古い，相関処理特性が Symptom のインシデントを削除します。

```
nnmtrimincidents.ovpl -age 6 -incr weeks -nature Symptom
```

6 か月より古い，ライフサイクル状態が Closed のインシデントを削除します。

```
nnmtrimincidents.ovpl -age 6 -incr months -lifecycle Closed
```

6 か月より古い，重大度が Normal のインシデントを削除します。

```
nnmtrimincidents.ovpl -age 6 -incr months -severity Normal
```

指定日付より古いインシデントを削除します。

```
nnmtrimincidents.ovpl -date 2007-07-16T19:20:30
```

指定日付より古い，相関処理特性が Symptom のインシデントを削除します。

```
nnmtrimincidents.ovpl -date 2007-07-16T19:20:30+01:00 -nature Symptom
```

指定日付より古い，ライフサイクル状態が Closed のインシデントを削除します。

```
nnmtrimincidents.ovpl -date 2007-07-16T19:20:30-01:00 -lifecycle Closed
```

6 日前より古いインシデントを削除し，アーカイブ用に指定したファイル名を使って，アーカイブに保存します。

```
nnmtrimincidents.ovpl -trimAndArchive -age 6 -incr days -path "C:¥BkupDir¥saveIncidents.gz"
```

6 日前より古いインシデントを削除します（アーカイブへの保存はしません）。

```
nnmtrimincidents.ovpl -age 6 -incr days
```

相関処理特性が Symptom の最も古い 10,000 個の SNMP トラップを削除します。

```
nnmtrimincidents.ovpl -trimOldest 10000 -nature Symptom -origin SnmpTrap
```

6 日前より古いインシデントを削除します（アーカイブへの保存はしません）。

```
nnmtrimincidents.ovpl -trimOnly -age 6 -incr days
```

6 日前より古いインシデントをアーカイブに保存します（削除はしません）。

```
nnmtrimincidents.ovpl -archiveOnly -age 6 -incr days
```

6 日前より古いインシデントを削除し，アーカイブに保存します。

```
nnmtrimincidents.ovpl -trimAndArchive -age 6 -incr days
```

AUTHOR

nnmtrimincidents.ovpl was developed by Micro Focus.

SEE ALSO

nnm.properties

付録 G.61　ovjboss

jboss アプリケーションサーバーのラッパー

SYNOPSIS

ovjboss

DESCRIPTION

ovjboss はovspmd の管理下にあるサービスコンポーネントです。ovjboss は，%NNM_DATA%¥shared¥nnm¥conf¥props（Windows）または$NNM_DATA/shared/nnm/conf/props（Linux）ディレクトリにあるプロパティファイル（nms-support.properties，nms-jboss.properties およびovjboss.jvmargs）を使用して jboss アプリケーションサーバーに引数を渡します。各ファイルには，制御する設定の変更方法を記載した資料が格納されています。

このコマンドは絶対に直接実行してはいけませんが，ovspmd の管理下にあります。このコマンドはovstart またはovstart -c ovjboss の実行時に起動されます。このコマンドを停止するには，ovstop またはovstop -c ovjboss を実行してください。このコマンドが監視している内部サービスの状態を参照するには，ovstatus -v ovjboss を呼び出してください。

ovbjoss の起動に問題がある場合，ovjboss.log およびnnm.log ログファイルを参照して，問題解決に役立つ情報があるかどうか確認できます。

EXAMPLES

ovjboss を含めた*NNMi* を起動するには，次のコマンドを実行します。

```
ovstart
```

ovjboss だけを起動するには，次のコマンドを実行します。

```
ovstart -c ovjboss
```

ovjboss が起動したサービスの状態を参照するには，次のコマンドを実行します。

```
ovstatus -v ovjboss
```

AUTHOR

ovjboss was developed by Micro Focus.

FILES

- Windows：%NNM_DATA%¥shared¥nnm¥conf¥props¥nms-jboss.properties
- Linux：$NNM_DATA/shared/nnm/conf/props/nms-jboss.properties

ovjboss 内部で起動したサービスが使用するパラメータファイル

- Windows：%NNM_DATA%¥shared¥nnm¥conf¥props¥nms-support.properties
- Linux：$NNM_DATA/shared/nnm/conf/props/nms-support.properties

ovjboss 内部で起動したサービスが使用するパラメータファイル

- Windows：%NNM_DATA%¥shared¥nnm¥conf¥props¥ovjboss.jvmargs

- Linux：$NNM_DATA/shared/nnm/conf/props/ovjboss.jvmargs

jboss を実行する JVM に渡すパラメータ

- Windows：%NNM_DATA%¥Conf¥nnm¥props¥nms-local.properties
- Linux：$NNM_DATA/conf/nnm/props/nms-local.properties

ポートの設定を含むローカル設定ファイル

- Windows：%NNM_DATA%¥log¥nnm¥nnm.log
- Linux：$NNM_DATA/log/nnm/nnm.log

（あれば）例外を格納するログファイル

- Windows：%NNM_DATA%¥log¥nnm¥ovjboss.log
- Linux：$NNM_DATA/log/nnm/ovjboss.log

標準エラー出力メッセージを格納するログファイル

SEE ALSO

ovspmd

nnm.ports

付録 G.62　ovspmd

NNMi プロセス管理サービス

SYNOPSIS

ovspmd [[install] [start] [stop] [remove] [-W] [-d] [-V] [-f *startup_file*]]

DESCRIPTION

ovspmd は NNMi の一部である各サービスプロセスを管理します。ovspmd はovstart，ovstop，および ovstatus からの要求に応じてこれらプロセスの起動，停止，状態の報告を行います。通常，ovspmd は ovstart が自動的に起動します。Windows では，ovspmd はサービスとして登録されています。ovspmd は サービス名 HP OpenView Process Manager 配下に登録します。

ovstart は，NNMi 起動ファイル（SUF）に指定したオブジェクトマネージャプログラムの開始要求を ovspmd に送信します。SUF のデフォルトはovsuf です。NNMi 管理対象プロセスはローカル登録ファイル（LRF）で構成され，ovaddobj によって SUF に追加されます。引数なしでovstart を呼び出した場合，ovspmd は自動的に起動するように構成された（つまり初期開始フラグOVs_YES_START を LRF に設定した）すべての管理対象プロセスを起動します。

ovstop は構成済みの管理対象プロセスの停止要求をovspmd に送信します。ovstop を引数なしで呼び出す場合，ovspmd は現在実行中のすべての管理対象プロセスを停止，終了します。

ovstatus は，構成済みの管理対象プロセスの現在の実行状態の報告要求をovspmd に送信します。

管理対象プロセスはovspmd によってサービスとして（つまり標準入力，標準出力，標準エラー出力のすべてを無視した状態のバックグラウンドで）起動されます。

各管理対象プロセスは，依存関係リスト（プロセスの起動を成功させるためにあらかじめ起動しておく必要がある他プロセスのリスト）を付けて構成できます。ovspmd は管理対象プロセスが依存しているすべてのプロセスの初期化が成功していなければ，その管理対象プロセスを起動しません。起動時には，ovspmd は LRF 指定の依存関係が循環形式になっていないことを確認します。(循環例は A -> B -> C -> A です。) この依存関係で，起動の相対的な順序付けも停止の逆順序も決定します。

ovspmd には，予期せず失敗したプロセスを自動的に再起動する仕組みがあります。このプロセスでは，%NNM_SHARED_CONF%¥ovspmd.restart.properties ファイル（Windows），または，$NNM_SHARED_CONF/ovspmd.restart.properties ファイル（Linux）に一覧表示しているようにデーモンプロセスのリトライ数を追加する必要があります。デフォルトでは，リトライ数は 0 です。プロセスが予期せず終了すると，この数が 0 に達するまで一つずつ減ります。その時点では，プロセスは自動的に再起動されません。

ovstart でプロセスを起動しようとすると，リトライ数がリセットされ，プロセスが再度起動されます。プロセスが 2 時間実行を続けた場合，そのプロセスはリトライカウンターをリセットします。エントリを削除すると，ovspmd は再起動を行わなくなります。これはリトライ数が 0 である場合にも当てはまります。

ovspmd はオブジェクトマネージャの三つのクラスを区別します。

OVs_WELL_BEHAVED

well-behaved なプロセスは OVsPMD API（OVsPMD_API(3)参照）を使ってovspmd と通信します。このプロセスは，初期化の成否や正常終了と異常終了に関するovspmd 状態情報を送信するように構成されている場合は，これらの情報を送信します。well-behaved なプロセスから初期化の成功が明示的に報告された場合にのみ，ovspmd はこのプロセスの初期化が成功したと見なします。また，well-behaved なプロセスは，ovspmd からOVS_CMD_EXIT コマンドを受信したときに終了します。

管理対象プロセスがovspmd に渡した状態情報は，ovstart，ovstop，ovstatus のうち現在実行中のものに転送されます。各管理対象プロセスから受信した最後のメッセージは，保存されてから，要求があり次第ovstatus に転送されます。well-behaved なプロセスから受信したメッセージもアプリケーションイベントログ（イベントビューアで調べることができます）にロギングされます。

OVs_NON_WELL_BEHAVED

ovspmd はOVsPMDAPI(non-well-behaved processes)を使わないオブジェクトマネージャ（non-well-behaved なプロセス）が自発的にバックグラウンドに突入しない場合に限り，これらも管理できます（下記OVs_DAEMON を参照)。non-well-behaved なプロセスは状態メッセージを返さないので，ovspmd はこのようなプロセスが LRF 指定のタイムアウト時間内に終了しなくても，プロセスの初期化が成功したものと見なします。

設定されたタイムアウト内に non-well-behaved なプロセスが終了しない場合は，Terminal Process で終了されます。

OVs_DAEMON

バックグラウンドに突入する管理対象プロセスは，通信チャネルまたは信号を使って管理できません。ovspmd はこのようなプロセスを起動できますが，必要な通信チャネルもプロセス ID もないので，プロセスを停止したり意味のある状態を報告したりできません。

Parameters

install

ovspmd をサービスとしてインストールします。

start

ovspmd サービスを起動します。

stop

ovspmd サービスを停止します。

remove

ovspmd サービスを削除します。

-W

ovspmd の起動時に管理対象プロセスを起動しません。ovstart からの起動要求を待ちます。

-d

デバッグ用に使用します。このオプションを使用した場合，ovspmd はサービスになりません。

-V

超冗長モードで実行します。このモードでは，ovspmd は管理対象プロセスの構成に関する非常に詳しい情報を出力します。これは通常の使用では詳細過ぎる情報です。

-f *startup_file*

デフォルトではなく*startup_file* を起動ファイル（SUF）として読み込みます。なお，*startup_file* は絶対パスである必要があります。

Application Authorization

ovspmd は NNMi サービスの管理を統制します。ovspmd はどのホスト，ユーザー，アプリケーションが NNMi サービスを起動および停止できるのかをovspmd.auth ファイルを使って制御します。ovspmd.auth ファイルの格納場所は%NNM_SHARED_CONF%（Windows）または$NNM_SHARED_CONF（Linux）です。

ovspmd はovspmd.auth ファイルのエントリを先頭から終わりまで検索します。検討中のアクセスを明示的に許可または拒否しているエントリを見つけ次第，ovspmd は検索を中止します。従って，より具体的なエントリのほうが一般的なエントリより優先されます。

ファイルには，権限があるホスト，ユーザー，およびアプリケーションを指定する行が含まれています。各行には，ovspmd への接続権限がある単一のホスト，ユーザー，およびアプリケーションリストを表示します。ファイルの各行の形式は次のとおりです。

#comment

hostname [*username* [*appname1 appname2 appname3...*]]

ナンバー記号（#）およびそれに続く部分はコメントであり，無視されます。空白行も無視されます。

username と*appname* は任意指定です。アプリケーションを指定しない場合，その行はどのアプリケーションからのアクセスも許可（または拒否）します。ユーザー名を指定しない場合，その行はどのアプリケーションで動作しているどのユーザーからのアクセスも許可（または拒否）します。

hostname にプラス記号（+）を指定すると，その行はどのホストからのアクセスも参照します。*username* にプラス記号（+）を指定すると，その行はどのユーザーからのアクセスも参照します。*hostname* の先頭にマイナス記号（-）を指定すると，その行はそのホストからのすべてのアクセスを明示的に拒否します（同じ行にユーザー名やアプリケーション名が指定されていても無視されます）。*username* の先頭にマイナス記号（-）を指定すると，その行は指定ホストのユーザーによるアクセスを明示的に拒否します（同じ行にアプリケーション名が指定されていても無視されます）。

アプリケーションが一覧表示されている場合，その行は表示されたアプリケーションへの（指定ホストからの指定ユーザーによる）アクセスだけを許可します。登録アプリケーション名に含まれる空白はアンダースコアに置換する必要があるという点を除いて，権限ファイルに一覧表示されたアプリケーション名はアプリケーションの登録名に一致しなければならないことに注意してください。

インストール時に作成されるovspmd.auth ファイルには，ファイル形式の例がさらに格納されており，いくつかの例は「EXAMPLES」の項にも記載されています。

DIAGNOSTICS

ovspmd は構成エラーおよびシステム呼び出し障害に関するエラーメッセージを発行します。これらのメッセージは内容が一目瞭然であることを目的としています。ovspmd が現在ovstart，ovstop，ovstatus のいずれかとオープン通信チャネルを持っている場合，ovspmd はこれらのエラーメッセージをプログラムが出力する通信チャネル経由で転送します。

ovspmd は複数の要求（start，stop，status）を一度に処理できます。追加の要求は現在の要求が完了するまでタイプ別に待ち行列に入れられます。

さらに，ovspmd はERROR レベルの OVS サブシステム内の nettl を使って，処理，構成，システムエラーなどをロギングします。初期化成功などの通常イベントを示すメッセージはINFORMATIVE レベルでロギングされます。初期化失敗または異常終了を示すメッセージはWARNING レベルでロギングされます。

EXAMPLES

ovspmd.auth ファイルの内容例を次に示します。

```
# Normally, you should authorize any application
# run by any user on the same host on which ovspmd is running.
# To do so, use a single line listing the
# name of the host on which this file is located
```

```
# (for example, "thishost"):

thishost

# Similarly, if you are running Management
# Consoles, you should authorize any application
# run by any user on all the client hosts and on
# the server host. For example, if your server
# system named "bigsystem" has one client named
# "hohum", list each of them on a separate line in
# this file on bigsystem:

bigsystem
hohum

# It is possible to permit specific users to run
# specific applications from a remote system. The
# following line permits the user "shem" from host
# "blimp" to run the applications "Toaster Manager"
# and "Blender". Note that, because the application's
# registered name "Toaster Manager" contains white
# space, you must replace the whitespace with the
# underscore character in the authorization file:

shem blimp Toaster_Manager Blender

# It is not possible to exclude specific applications,
# except by explicitly permitting all non-excluded
# applications.

# The following line denies access by the user "fred"
# from any host:

+ -fred

# The following line denies any application access
# from the host "badguy":

-badguy
```

AUTHOR

ovspmd was developed by Micro Focus.

FILES

次に示すファイルの環境変数の使用については，nnm.envvars リファレンスページを参照してください。

Windows：

%NNM_BIN%¥ovspmd

%NNM_SHARED_CONF%¥ovsuf

Linux：

$NNM_BIN/ovspmd

$NNM_SHARED_CONF/ovsuf

再起動プロパティ構成については，%NNM_SHARED_CONF%¥ovspmd.restart.properties（Windows）または$NNM_SHARED_CONF/ovspmd.restart.properties（Linux）を参照してください。

EXTERNAL INFLUENCES

Environmental Variables

$LANG は，国際化変数 LC_ALL，LC_CTYPE，LC_MESSAGES が未設定，NULL，または無効な場合にデフォルト値を提供します。

$LANG が未設定，NULL，または無効な場合，デフォルト値のC（Windows の場合English_UnitedStates.1252）が使用されます。

LC_ALL（または$LANG）はovspmd が起動したほかのすべてのプロセスのロケールを決定します。

LC_CTYPE は，テキストをシングルバイト文字，マルチバイト文字，あるいはその両方として解釈すること，文字の分類は印刷可能とすること，正規表現の文字クラス表現によって文字を一致させることを決定します。

LC_MESSAGES は，メッセージを表示する言語を決定します。

他のすべての環境変数はovspmd を実行しているシェル（またはovspmd を起動した初回 ovstart）から引き継がれます。ovspmd およびすべてのサービスプロセスはこの環境を共有します。結果として，環境変更を有効にするためには ovspmd を停止し，再起動する必要があります（ovstart 参照）。

SEE ALSO

ovstatus, ovstart, ovstop, nnmcluster.

付録 G.63　ovstart

NNMi 管理対象プロセスの起動

SYNOPSIS

ovstart [[-c] [-d] [-o *ovspmd_path*] [-v] [--] [*ovspmd_options...*]
[*managed_process_names...*]]

DESCRIPTION

ovstart は NNMi 監視対象プロセスを起動します。一つ以上の*managed_process_name* 引数を指定して呼び出された場合，ovstart は指定した管理対象プロセスが依存している他の管理対象プロセスを最初に起動し

てから，指定した管理対象プロセスを起動します。引数なしで呼び出された場合，ovstart はデフォルトで起動するように構成されているすべての管理対象プロセスを起動します。

ovstart は，起動しようとしたすべての管理対象プロセスが応答またはタイムアウト（LRF 指定のタイムアウト時間内の応答に失敗）するまで終了しません。デフォルトでは，ovstart は，管理対象プロセスが失敗するまで出力を作成しません。ovstart をコマンドラインから実行した場合は，操作の進捗を把握するために-c または-v オプションを使用するとよいでしょう。前回のが成功した後でovstart を再度起動してもまったく害はありません。

ovstart は，起動要求（OVS_REQ_START）をプロセス管理サービスであるovspmd に送信します。ovspmd がまだ起動中でない場合，ovstart はまずこれを起動します。

ovstart は Windows システムの管理者または Linux システムの root が実行する必要があります。

管理対象プロセスはovaddobj によってローカル登録ファイルの情報から構成されます。管理対象プロセスは，その管理対象プロセスを記述している LRF の第 1 フィールドによって命名されます。

NNMi クラスタ（nnmcluster 参照）用に構成されたノードでovstart を使用した場合，ovstart の動作は上記とは異なります。具体的に言うと，ovstart は”nnmcluster -daemon”コマンドとまったく同じ動作をします。

NNMi クラスタ環境では，ovstart は（バックグラウンドの NNMi クラスタを起動後）すぐに返ります。その代わりに，nnmcluster コマンドがほかの NNMi プロセスを起動するかどうか，またはいつ起動するのかを決定します。ovstatus 出力を監視して，NNMi プロセスの起動が完了したかどうか判断してください。

NNMi クラスタ環境では，ovstart のほかのコマンドラインオプションはサポートしていません。

なお，NNMi クラスタ属性をきめ細かく制御するために，nnmcluster コマンドを直接使用してください。NNMi クラスタ環境でのovstart コマンドは，よく知られたコマンドを使って NNMi を起動する場合の便宜のために提供されています。

Parameters

ovstart コマンドは，次のオプションをサポートします。サポートされないオプションを指定した場合，使用方法のメッセージが表示されます。

-c

各管理対象プロセスの成否に関する 1 行分の情報を作成します。

-d

ovspmd の起動，連絡，および起動要求の送信，および通信チャネルの閉鎖を含む処理の重要な段階を報告します。

-o *ovspmd_path*

ovspmd の実行可能ファイルをデフォルト位置（Windows の場合：%NNM_BIN%，Linux の場合：$NNM_BIN）ではなくovspmd_path に格納するように指定します。ovspmd が既に起動中の場合は，このオプションは無視されます。

-v

各管理対象プロセスの成否に関する複数行の情報を作成します。

- *ovspmd_options*

ovstart に知られていないオプションがあればovspmd に渡します。-d オプションは両方のプログラムに対して有効なので，ovstart オプションと解釈され，ovspmd には渡されません。同様に，-V オプションはovstart に対して有効ではないので，ovspmd に渡されます。オプションがどちらにも認識されない場合，使用方法メッセージは ovstart からではなくovspmd から出力されます。

--

ovstart コマンドラインの options セクションを終了します。コメントトークン（--）の後のすべての引数は，起動する管理対象プロセス名と解釈され，ovspmd に渡されます。

RETURN VALUE

非 NNMi クラスタ環境では，ovstart は起動に成功しなかったオブジェクトマネージャ数を開始行から示す状態で終了します。要求されたすべての管理対象プロセスが起動に成功した場合，ovstart はステータス0（ゼロ）で終了します。

NNMi クラスタ環境では，ovstart は常にステータス0（ゼロ）で即時終了します。

DIAGNOSTICS

ovstart は特定のコマンドラインエラー（特に引数過多）およびシステムエラーを報告します。メッセージの先頭には「ovstart:」が付与され，内容が一目瞭然であることを目的としています。ovstart はovspmd から受信したエラーメッセージも出力します。これらのメッセージの先頭には「ovspmd:」が付与されます。ovstart はサポートしていないオプションをエラーとして扱いませんが，ovspmd はこれらをエラーとして扱います。

ovspmd は複数の要求（ovstart，ovstop，ovstatus）を一度に処理できることに注意してください。これらのコマンドのどれかが処理中の場合，新規要求は前回のコマンドが完了するまでタイプ別に待ち行列に入れられます。

EXAMPLES

```
ovstart
```

デフォルトで起動するように構成されているすべての管理対象プロセスの起動をovspmd に要求します。ovspmd がまだ起動中でない場合，オプションなしでこれを起動します。失敗のみが報告されます。

```
ovstart -v -V -- ovjboss
```

ovjboss プロセスの起動をovspmd に要求します。これにより，ovjboss プロセスが依存しているほかの管理対象プロセスを最初に起動してから，Jboss アプリケーションサーバーおよび Jboss 内で一緒に展開されているすべての NNMi サービスが起動されます。ovspmd がまだ起動中でない場合，冗長モードで起動します（-V オプション）。プログラム起動の成否を報告します（-v オプション）。ovstart がovjboss をサポートされていない-V オプションの引数と解釈しないように，コメントトークン（--）オプションが必要であることに注意してください。

AUTHOR

ovstart was developed by Micro Focus.

FILES

次に示すファイルの環境変数の使用については，nnm.envvars リファレンスページを参照してください。

Windows：

```
%NNM_BIN%¥ovstart
%NNM_BIN%¥ovspmd
```

Linux：

```
$NNM_BIN/ovstart
$NNM_BIN/ovspmd
```

EXTERNAL INFLUENCES

Environmental Variables

$LANG は，国際化変数LC_ALL, LC_CTYPE, LC_MESSAGES が未設定，NULL，または無効な場合にデフォルト値を提供します。

$LANG が未設定，NULL，または無効な場合，デフォルト値のC（Windows の場合 English_UnitedStates.1252）が使用されます。

LC_ALL（または$LANG）はovspmd が起動したほかのすべてのプロセスのロケールを決定します。

LC_CTYPE は，テキストをシングルバイト文字，マルチバイト文字，あるいはその両方として解釈すること，文字の分類は印刷可能とすること，正規表現の文字クラス表現によって文字を一致させることを決定します。

LC_MESSAGES は，メッセージを表示する言語を決定します。

ovstart が実行され，かつ現在実行中のovspmd プロセスがない場合，ovspmd は実行シェルの環境を引き継ぎます。ovspmd が起動したすべての管理対象プロセスはこの同じ環境を引き継ぎます。

ovspmd または管理対象プロセスの環境を変更するには，正しい環境でovspmd を再起動する必要があります。このためには，すべての管理対象プロセスを停止する必要があります（ovspmd はすべての管理対象プロセスがシャットダウンされるまで終了しません）。

結果として，ovstart またはovspmd から起動した管理対象プロセスの環境を変更するには，下記を実行する必要があります。

1. 引数なしでovstop を実行して，すべての管理対象プロセスおよびovspmd をシャットダウンします。
2. 正しい環境変数を設定します。
3. ovstart を実行して，ovspmd および一部またはすべての管理対象プロセスを再起動します。

NNMi Cluster

com.hp.ov.nms.cluster.name が%NNM_DATA%¥shared¥nnm¥conf¥props¥nms-cluster.properties ファイル（Windows の場合）または，$NNM_DATA/shared/nnm/conf/props/nms-cluster.properties ファイル（Linux の場合）で定義されている場合，ovstart は起動をnnmcluster コマンドに任せます。

SEE ALSO

ovstatus, ovstop, ovspmd, nnmcluster.

付録 G.64　ovstop

NNMi 管理対象プロセスの停止

SYNOPSIS

ovstop [[-c] [-d] [-v] [*managed_process_names*...]] [[-failover|-nofailover|-cluster]]

DESCRIPTION

ovstop は NNMi 監視対象プロセスを停止します。ovstop は停止要求（OVS_REQ_STOP）をプロセス管理プロセス（Linux オペレーティングシステム）またはサービス（Windows オペレーティングシステム）であるovspmd に送信します。一つ以上の *managed_process_name* 引数で呼び出された場合，ovstop は最初に依存プロセスの一つを停止してから，指定した管理対象プロセスを停止します。引数なしで呼び出された場合，または名前付き引数の一つがovspmd である場合，ovstop はovspmd 自身を含む現在実行中のすべての管理対象プロセスを停止します。

LRF で指定したタイムアウト時間内に管理対象プロセスがovstop 要求に対して応答しない場合，ovspmd はプロセスに終了信号をSIGTERM，SIGKILL（kill 参照）の順で送信することによってプロセスを強制終了します。ovstop が強制終了を報告するのは，-v または-c オプションを使用している時（例えばovstop -v [*managed_process_name*]）だけですので注意してください。停止要求時に管理対象プロセスがタイムアウトした場合は，すぐにタイムアウト値を大きくするとよいでしょう。プロセスがovstop 要求に応答するまでovspmd が待つ時間（秒）を大きくするには，%NNM_LRF%¥ov*（Windows オペレーティングシステム）または$NNM_LRF/ov*（Linux オペレーティングシステム）の指示に従ってください。

ovstart とは異なり，ovstop はovspmd が既に起動中ではなくてもこれを起動しません。

管理対象プロセスはovaddobj によってローカル登録ファイルの情報から構成されます。管理対象プロセスは，その管理対象プロセスを記述しているLRF の第1フィールドによって命名されます。ovstart と同様，ovstop はLRF にある依存情報を使用します。ほかの管理対象プロセスが，停止している管理対象プロセスに依存する場合，ovspmd は依存関係を記録し，適切なすべての管理対象プロセスをLRF の依存順序とは逆の順序で終了します。

ovstop はWindows オペレーティングシステムの管理者またはLinux オペレーティングシステムのroot が実行する必要があります。

OVs_DAEMON プロセスがLRF エントリ内のStop Command を用いて構成される場合，ovstop はコマンドを実行します。この機能はovspmd とはもう連絡していないプロセスの停止に使用します。Stop Command は必要に応じてプロセスの開発者が提供し，構成します。

前回のovstart 操作で起動されたNNMi 管理対象プロセスの名前は，ovstatus -c コマンドの実行によって取得できます。

ovstop ovjboss コマンドはJboss アプリケーションサーバーおよびJboss 内で一緒に展開されたすべてのNNMi サービスを停止します。Jboss 展開されたNNMi サービスの名前はovstatus -v ovjboss コマンドの実行によって取得できます。ovstop ovjboss の実行によって，すべてのNNMi サービスの停止のみが可能です。一部のNNMi サービスをほかのNNMi サービスとは無関係に個別に停止するということはサポートしていません。

NNMi クラスタ（nnmcluster 参照）用に構成されたノードでovstop を使用した場合，ovstop の動作は上記とは異なります。具体的に言うと，ovstop（パラメータ指定なし）は”nnmcluster -disable -shutdown”コマンドとまったく同じ動作をします。

NNMi クラスタ環境では，ovstop は（バックグラウンドのNNMi クラスタにシャットダウン信号を送信後）すぐに返ります。次にnnmcluster コマンドが，スタンバイクラスタノードへのNNMi サービスのフェイルオーバーの契機となり得るNNMi プロセスをシャットダウンします。ovstatus の出力を監視して，NNMi プロセスがシャットダウンを完了したかどうか判断してください。

NNMi クラスタ環境では，ovstop が認識するコマンドラインオプションは-failover,-nofailover,-cluster だけです。

なお，NNMi クラスタ属性をきめ細かく制御するために，nnmcluster コマンドを直接使用してください。NNMi クラスタ環境でのovstop コマンドは，よく知られたコマンドを使ってNNMi サービスをシャットダウンする場合の便宜のために提供されています。

Parameters

ovstop コマンドは，次のオプションをサポートします。オプションではない第1引数と，以降のすべての引数は停止させる管理対象プロセスの名前として解釈され，停止要求でovspmd に渡されます。

-c

各管理対象プロセスの成否に関する1行分の情報を作成します。

-d

ovspmd への連絡と停止要求の送信，および通信チャネルの閉鎖を含む処理の重要な段階を報告します。

-v

各管理対象プロセスの成否に関する複数行の情報を作成します。

-failover

（NNMi クラスタのみ）ローカル NNMi ノード（アクティブノードである場合）に各 NNMi プロセスをシャットダウンさせ，NNMi クラスタプロセスを終了します。同時に，自動フェイルオーバーを有効にして NNMi サービスがスタンバイノードに移動するようにします。

-nofailover

（NNMi クラスタのみ）ローカル NNMi ノード（アクティブノードである場合）に各 NNMi プロセスをシャットダウンさせ，NNMi クラスタプロセスを終了します。同時に，自動フェイルオーバーを無効にして NNMi サービスがスタンバイノードに移動しないようにします。

-cluster

（NNMi クラスタのみ）NNMi クラスタ内の全ノードをシャットダウンさせます。スタンバイノードの NNMi クラスタプロセスは最初にシャットダウンされ，次にアクティブノードが NNMi サービスを停止し，最後に現用ノードの NNMi クラスタプロセスがシャットダウンします。

RETURN VALUE

ovstop は，停止に成功しなかった管理対象プロセス数を示す状態で終了します。要求されたすべての管理対象プロセスが停止に成功した場合，ovstop はステータス0（ゼロ）で終了します。

DIAGNOSTICS

ovstop は特定のコマンドラインエラー（特に引数過多）およびシステムエラーを報告します。メッセージの先頭には「ovstop:」が付与され，内容が一目瞭然であることを目的としています。ovstop はovspmd から受信したエラーメッセージも出力します。これらのメッセージの先頭には「ovspmd:」が付与されます。ovstop はサポートしていないオプションは無視します。

管理対象プロセスの状態がPAUSED,PAUSE_ERROR,PAUSE_TIMEOUT,RESUME_ERROR,RESUME_TIMEOUT,または DEPENDENCY_ERR のいずれかである場合，その管理対象プロセスは停止されます。ただし，警告メッセージが印刷されて，実行中状態ではないプロセスに対してovstop が使われたことを通知します。

ovspmd は複数の要求(ovstart,ovstop,ovstatus)を一度に処理できることに注意してください。これらのコマンドのどれかが処理中の場合，新規要求は前回のコマンドが完了するまでタイプ別に待ち行列に入れられます。

AUTHOR

ovstop was developed by Micro Focus.

FILES

次に示す環境変数は，お使いのシェルおよびプラットフォーム要件に応じて確立された汎用パス名を表します。お使いのプラットフォームおよびシェルの汎用パス名については，nnm.envvars リファレンスページを参照してください。

次に示すファイルの環境変数の使用については，nnm.envvars リファレンスページを参照してください。

Windows：

```
%NNM_BIN%¥ovstop
%NNM_BIN%¥ovspmd
```

Linux：

```
$NNM_BIN/ovstop
$NNM_BIN/ovspmd
```

EXTERNAL INFLUENCES

Environmental Variables

$LANG は，国際化変数LC_ALL，LC_CTYPE，LC_MESSAGES が未設定，NULL，または無効な場合にデフォルト値を提供します。

$LANG が未設定，NULL，または無効な場合，デフォルト値のC（Windows の場合 English_UnitedStates.1252）が使用されます。

LC_ALL（または$LANG）はovspmd が起動したほかのすべてのプロセスのロケールを決定します。

LC_CTYPE は，テキストをシングルバイト文字，マルチバイト文字，あるいはその両方として解釈すること，文字の分類は印刷可能とすること，正規表現の文字クラス表現によって文字を一致させることを決定します。

LC_MESSAGES は，メッセージを表示する言語を決定します。

NNMi Cluster

com.hp.ov.nms.cluster.name が %NNM_DATA%¥shared¥nnm¥conf¥props¥nms-cluster.properties ファイル（Windows オペレーティングシステムの場合）または，$NNM_DATA/shared/nnm/conf/props/nms-cluster.properties ファイル（Linux オペレーティングシステムの場合）で定義されている場合，ovstop は停止を nnmcluster コマンドに任せます。

SEE ALSO

ovstatus, ovstart, ovspmd, nnmcluster.

付録 H　リファレンスページ（File Formats）

付録 H.1　disco.NoVLANIndexing

検出のポーリング時に VLAN Indexing を省略するノードを指定します。

SYNOPSIS

disco.NoVLANIndexing

DESCRIPTION

NNMi が管理対象ネットワーク内のスイッチ間のレイヤー 2 接続の状況を知る方法の一つは，各スイッチからdot1dTpFdbTable（FDB）を読み出すことです。しかし，Cisco 製スイッチの場合は，FDB 全体を読み出すためにVLAN Indexing を使用する必要があります。VLAN Indexing を使用することで，NNMi は Cisco 製スイッチに設定された各 VLAN の FDB をその都度読み出します。各デバイス上に設定されている VLAN の数が多い場合は，VLAN Indexing を使用した FDB の読み出しは，完了までに非常に長い時間がかかり，場合によっては数時間かかることもあります。

Cisco 製スイッチは，多くの場合，Cisco Discovery Protocol（CDP）を使用する設定になっています。CDP は，レイヤー 2 接続の状況を知るためのより優れた方法と考えられています。ネットワークのコアに位置する大規模なスイッチには，多数の VLAN が設定されている場合があります。こうしたスイッチでは，通常はスイッチ自身に直接接続されているエンドノードがありません。管理対象スイッチにエンドノードが直接接続されていない場合，こうした大規模なスイッチでは FDB の収集を抑制する設定にした方が良いかもしれません。FDB の収集を抑制しても，NNMi は CDP から収集したデータを使用してレイヤー 2 の検出を実行できます。VLAN Indexing を抑制する対象としては，まず，こうした大規模なスイッチが最初の候補となります。一方，ネットワークの端に位置し，多数のエンドノードが接続されている小規模なスイッチ（多くの場合，アクセススイッチと呼ばれます）では，VLAN Indexing を抑制しないでください。

VLAN Indexing を抑制するように NNMi を設定できます。これを行うには，NNMi の管理者は，disco.NoVLANIndexing ファイルを作成する必要があります。このファイルの名前は大文字と小文字を区別します。disco.NoVLANIndexing ファイルは，ovjboss プロセスの起動時に読み込まれます。NNMi 管理者がovjboss プロセスの起動後にdisco.NoVLANIndexing ファイルを変更した場合，その変更は次にovjboss プロセスが起動されるまで有効になりません。disco.NoVLANIndexing ファイルは初期状態では用意されていません。disco.NoVLANIndexing ファイルが存在しない場合はこの機能が無効になるため，NNMi はVLAN Indexing を使用し，すべてのデバイスについて FDB テーブル全体を収集しようとします。

disco.NoVLANIndexing ファイルには，IP アドレス，IP アドレスの範囲およびコメントを記述できます。コメントは，ポンド（ハッシュ）記号（#）と，#から行末までのすべての文字で構成されます。NNMi では空白行はコメントとして扱われます。IP アドレスは，IP バージョン 4 の場合は標準のドット付き 10 進数表記，IP バージョン 6 の場合は標準形式（RFC 2373）で指定します。

IP アドレス範囲のフォーマットの詳細については，NNMi ヘルプ［領域のアドレス範囲を設定する］を参照してください。

リストされている IP アドレスのどれかとノードの管理アドレスが一致する場合，そのノードは一致ノードと見なされます。ノードに設定されているほかの IP アドレスは考慮されません。ノードが disco.NoVLANIndexing ファイルに含まれるアドレスのどれかに一致すると，NNMi はデフォルトの FDB（@vlan-id のサフィックスが追加されていないコミュニティ文字列を使用してアクセス可能な FDB）だけを収集します。

FDB 全体の収集を無効にすると，管理ネットワークのレイヤー 2 マップのレイアウトが不正確になる場合があります。どのスイッチをdisco.NoVLANIndexing ファイルに含めるかについて，慎重に検討してください。

EXAMPLES

disco.NoVLANIndexing ファイルの例を次に示します。

```
#This entry suppresses VLAN-indexing for the node whose management address is 10.2.37.149
10.2.37.149

192.168.100-101.1 #This entry causes the nodes 192.168.100.1 and 192.168.101.1 to be skip
ped, too

# Here are some examples of IPv6 addresses and ranges:
   2136::8:800:200C:417a
   fd01::a352:1245:fc4B
   2001:D88:2:0:a07:ffff:0a01:3200-37ff
```

AUTHOR

disco.NoVLANIndexing was developed by Micro Focus.

FILES

- Windows：%NnmDataDir%shared¥nnm¥conf¥disco¥disco.NoVLANIndexing
- Linux：$NnmDataDir/shared/nnm/conf/disco/disco.NoVLANIndexing

SEE ALSO

詳細については，『NNMi の保守』の章および NNMi ヘルプ［領域のアドレス範囲を設定する］を参照してください。

付録 H.2　disco.SkipXdpProcessing

NNMi が検出プロトコル情報を問い合わせるべきでないノードの，管理 IP アドレスのリストが含まれます。

SYNOPSIS

disco.SkipXdpProcessing

DESCRIPTION

NNMi が管理ネットワークに含まれるネットワークデバイス間のレイヤー 2 接続を検出するとき，一つの方法として，検出プロトコル関連の情報をデバイスから収集する方法を使用します。検出プロトコルには，さまざまなプロトコルが定義されています。例えば，Link Layer Discovery Protocol（LLDP）は業界標準のプロトコルですが，Cisco 製デバイスに使用する Cisco Discovery Protocol（CDP）のように，ベンダー固有のプロトコルが多数存在します。これらのプロトコルはすべて，NNMi の検出機能のXdpAnalyzer によって処理されます。

指定したデバイスの検出プロトコル収集を行わないように NNMi を設定できます。 この機能では，NNMi 管理者が作成する設定ファイルdisco.SkipXdpProcessing を使用します。このファイルの名前は大文字と小文字を区別します。disco.SkipXdpProcessing は，ovjboss プロセスの起動時に読み込まれます。NNMi 管理者がovjboss プロセスの起動後にこのファイルを変更した場合，その変更は次にovjboss プロセスが起動されるまで有効になりません。disco.SkipXdpProcessing ファイルは初期状態では用意されていません。disco.SkipXdpProcessing が存在しない場合はこの機能が無効になるため，NNMi はすべての管理ノードから検出プロトコル情報を収集しようとします。

この機能によって修正される既知の問題の詳細については，下記の「SEE ALSO」の項を参照してください。

disco.SkipXdpProcessing ファイルには，IP アドレスとコメントを記述できます。コメントは，ポンド（ハッシュ）記号（#）と，#から行末までのすべての文字で構成されます。NNMi では空白行はコメントとして扱われます。IP アドレスは，IP バージョン 4 の場合は標準のドット付き 10 進数表記，IP バージョン 6 の場合は標準形式（RFC 2373）で指定します。

リストされている IP アドレスのどれかとノードの管理アドレスが一致する場合，そのノードは一致ノードと見なされます。ノードに設定されているほかの IP アドレスは考慮されません。ノードの管理アドレスとdisco.SkipXdpProcessing ファイルに含まれるアドレスのどれかが一致すると，そのノードのXdpAnalyzer サービスは省略され，検出プロトコル情報は収集されません。

ノードの検出プロトコル処理を無効にすると，管理ネットワークのレイヤー 2 マップのレイアウトが不正確になる場合があります。

EXAMPLES

disco.SkipXdpProcessing ファイルの例を次に示します。

```
#このエントリにより，管理アドレスが 10.2.37.149 のノードの XdpAnalyzer 処理が行われなくなります。
10.2.37.149

192.168.100.1 #このエントリにより，ノード 192.168.100.1 も省略対象となります。

#IPv6 アドレスの例を次に示します。：
```

```
2136::8:800:200C:417a
fd01::a352:1245:fc4B
```

AUTHOR

disco.SkipXdpProcessing was developed by Micro Focus.

FILES

- Windows：%NnmDataDir%shared¥nnm¥conf¥disco¥disco.SkipXdpProcessing
- Linux：$NnmDataDir/shared/nnm/conf/disco/disco.SkipXdpProcessing

SEE ALSO

詳細については，『NNMi の保守』の章を参照してください。

付録 H.3　hostnolookup.conf

システム IP ネームサーバーを使用した IP アドレス解決の対象から除外するホスト名またはワイルドカード指定のホスト名を格納したファイル

SYNOPSIS

hostnolookup.conf

DESCRIPTION

hostnolookup.conf は，ovjboss プロセスが，システム IP ネームサーバーを使用して IP アドレスを解決するかどうかを決定するのに使用するファイルです。NNMi プロセスovjboss は，ホスト名から IP アドレスを解決する前に，ホスト名とこのファイルの各エントリとを突き合わせチェックします。一致するエントリが見つかった場合，ovjboss プロセスはシステム IP ネームサーバーを使用した IP アドレス解決を行いません。

このファイルには，一つのホスト名またはホスト名のワイルドカードを含むエントリを追加します。一つのエントリを複数の行に記述することはできません。コメントを追加するには，コメントの前に記号（#）を追加します。このようにすると，その行の#以降の部分が無視されます。hostnolookup.conf ファイルには空白行を追加できます。

hostnolookup.conf ファイルは，特定のホスト名（またはホスト名の範囲）をシステム IP ネームサーバーを使用した IP アドレス解決から除外するときに使用します。

hostnolookup.conf ファイルは管理者が作成する必要があります。初期状態ではこのファイルは用意されていません。

NNMi プロセスが起動しているときにhostnolookup.conf ファイルの変更を反映する場合は，%NnmInstallDir%support¥nmsdnssync.ovpl コマンド（Windows の場合）または$NnmInstallDir/support/nmsdnssync.ovpl コマンド（Linux の場合）を引数なしで実行します。

EXAMPLES

hostnolookup.conf ファイルの例を次に示します。

```
# 特定のホスト名を指定する例
badsys.mydomain.mycorp.com
# ワイルドカードを使用したホスト名の指定例
*.baddomain.mycorp.com
```

最初の例では，DNS サーバーが予期しない結果を返すホスト名を指定しています。二番目の例では，解決できないドメイン名を指定しています。上記のエントリをhostnolookup.conf ファイルに追加すると，NNMi でホスト名解決が行われなくなります。

AUTHOR

hostnolookup.conf was developed by Micro Focus.

FILES

- Windows：%NNM_DATA%¥shared¥nnm¥conf¥hostnolookup.conf
- Linux：$NNM_DATA/shared/nnm/conf/hostnolookup.conf

SEE ALSO

ipnolookup.conf.

付録 H.4　ipnolookup.conf

システム IP ネームサーバーを使用したホスト名解決の対象から除外する IP アドレスまたはワイルドカード指定の IP アドレスを格納したファイル

SYNOPSIS

ipnolookup.conf

DESCRIPTION

ipnolookup.conf は，すべての NNMi プロセスが，システム IP ネームサーバーを使用してホスト名を解決するかどうかを決定するのに使用するファイルです。各 NNMi プロセスは，IP アドレスからホスト名を解決する前に，IP アドレスとこのファイルの各エントリとを突き合わせチェックします。一致するエントリが見つかった場合，NNMi プロセスはシステム IP ネームサーバーを使用したホスト解決を行いません。

このファイルには，各行に一つの IP アドレスまたは IP アドレスのワイルドカードを含むエントリを追加します。一つのエントリを複数の行に記述することはできません。コメントを追加するには，コメントの前に記号（#）を追加します。このようにすると，その行の#以降の部分が無視されます。ipnolookup.conf ファイルには空白行を追加できます。

ipnolookup.conf ファイルは，特定の IP アドレス（または IP アドレスの範囲）をシステム IP ネームサーバーを使用したホスト名解決から除外するときに使用します。

ipnolookup.conf ファイルは管理者が作成する必要があります。初期状態ではこのファイルは用意されていません。

NNMi プロセスが起動しているときにipnolookup.conf ファイルの変更を反映する場合は，%NnmInstallDir%support¥nmsdnssync.ovpl コマンド（Windows の場合）または$NnmInstallDir/support/nmsdnssync.ovpl コマンド（Linux の場合）を引数なしで実行します。

EXAMPLES

ipnolookup.conf ファイルの例を次に示します。

```
# 特定の IP アドレスを指定する例
192.168.1.100
# ワイルドカードを使用した IP アドレスの指定例
10.*.*.*
# IP アドレスの範囲指定例
192.168.1.101-255
```

最初の例では，多くの Web サイトが IP アドレスに192.168.*.*を使用するため，この単独 IP アドレスはインターネットにルーティングされる可能性があります。二番目の例に示すワイルドカードを使用した IP アドレス範囲指定は，NAT アドレスとなる可能性があります。したがって，この指定方法は通信には不向きです。三番目の例では，IP のワイルドカード範囲が，プライマリ IP アドレス以外の目的で使用されるアドレスの組である可能性があります。

AUTHOR

ipnolookup.conf was developed by Micro Focus.

FILES

- Windows：%NNM_DATA%¥shared¥nnm¥conf¥ipnolookup.conf
- Linux：$NNM_DATA/shared/nnm/conf/ipnolookup.conf

SEE ALSO

hostnolookup.conf.

付録 H.5　macdedupexceptions.txt

MAC アドレスベースのノードの重複削除ロジックの例外と見なされているノードタイプの sysObjectID の値を含むファイル

SYNOPSIS

```
macdedupexceptions.txt
```

DESCRIPTION

NNMi は，ノードが実際にはデータベース内の他のノードと重複していることを検出するために，様々な複雑なアルゴリズムを使用します。 場合によっては，NNMi は MAC アドレスを比較し，DHCP リースの期限切れのためにノードが新しい IP アドレスを割り当てられたかどうか判断します。これは，ファイアウォールやロードバランサーなどの一部のネットワークデバイスで問題が発生する場合があります。これらのデバイスは，複数の異なるデバイス間で共通の IP アドレスと MAC アドレスを使用することがあります。通常，NNMi は SNMP sysName の違いにより，これらのデバイスを区別します。しかし，異なる SNMP sysName を設定できない場合もあります。これらのケースでは，NNMi は，デバイスが重複していると判断して，データベースからいずれかのデバイスを削除することがあります。

このようなロードバランサーやファイアウォールのようなデバイスについては，`macdedupexceptions.txt` ファイルに，これらのデバイスの SNMP sysObjectID の値を記載することによって，重複削除アルゴリズムを修正するよう NNMi に指示することができます。このファイルに含めるための良い候補となるデバイスには，次の特性があります。

- デバイスは，DHCP サーバから IP アドレスを取得してはいけません。IP アドレスは静的に割り当てられる必要があります。
- デバイスは，一意の管理 IP アドレスを使用する必要があります。

この設定ファイルに有用なデバイスの例を次に示します。

- デバイスは，いくつかの共通 IP アドレスと MAC アドレスを使用し，同一の SNMP sysName を共有している別のデバイスと共に，冗長構成として設定されています。
- デバイスは，いくつかの仮想インスタンスをサポートする物理デバイスであり，そこでは各インスタンスが同様の IP アドレスと MAC アドレスを使用し，同一の SNMP sysName を共用している可能性があります。

正しく検出されることが必要なデバイスに関するエントリのみを，NNMi の管理者がこのファイルに追加することを推奨します。不要なエントリを追加すると，予期しない結果になるおそれがあります。

ファイルには，一つ以上の SNMP sysObjectID の値（各行に一つの値）を含めることができます。「#」で始まる行は，空行と同様にコメント行として扱われます。コメントは sysObjectID に続けて，「#」で開始し行末まで指定することもできます。sysObjectID の前後の空白は無視されます。 また，sysObjectID 値の先頭のドット（.）はオプションです。

初期状態ではこのファイルは用意されていません。必要な場合，NNMi の管理者が作成する必要があります。 ファイルは NNMi の起動時に読み込まれます。NNMi の起動後に行われた変更は，NNMi が再起動されるまで有効になりません。

EXAMPLES

macdedupexceptions.txt ファイルの例を次に示します。

```
# F5 BIG-IP Pb200 ロードバランサー機器
.1.3.6.1.4.1.3375.2.1.3.4.19

1.3.6.1.4.1.9.1.1291  #Cisco ACE サービスモジュール
```

AUTHOR

macdedupexceptions.txt was developed by Micro Focus.

FILES

- Windows：%NNM_DATA%¥shared¥nnm¥conf¥disco¥macdedupexceptions.txt
- Linux：$NNM_DATA/shared/nnm/conf/disco/macdedupexceptions.txt

付録 H.6　nnm.ports

次の情報は，NNMi 管理サーバーのリッスンポートを示します。ポート競合の場合には，これらのポート番号の多くを変更してください。

SYNOPSIS

nnm.ports

コマンドの概要はありません。*nnm.ports* リファレンスページには，nms-local.properties ファイルの修正によって変更できるウェルノウンポートが記載されています。

DESCRIPTION

ポート番号を変更するには，次の手順を実施します。

1. NNMi が使用するポートを変更するために，次のファイルを編集します。
   - Windows：%NnmDataDir%Conf¥nnm¥props¥nms-local.properties
   - Linux：$NnmDataDir/conf/nnm/props/nms-local.properties
2. 変更するポートを含む行を特定します。
3. 必要に応じて行の先頭の#!を削除します。

4. ポート番号を修正し，変更を保存します。
5. コマンドプロンプトで`ovstop`と`ovstart`を実行し，NNMi を再起動します。

現在定義されているポートを次に示します。

nmsas.server.port.web.http=80

この TCP ポートは，Web UI と Web サービスで，デフォルトの HTTP ポートとして使用されます。NNMi のインストール時に，この値を設定するよう求められます。このポートの値は，NNMi のインストール中，または`nms-local.properties`ファイルを修正することで変更できます。

nmsas.server.port.web.https=443

この TCP ポートは，Web UI と Web サービスで，デフォルトのセキュア HTTPS ポート(SSL)として使用されます。

nmsas.server.port.naming.rmi=1098

RMI ネーミングサービスのデフォルトの TCP ポートです。

nmsas.server.port.naming.port=1099

JNP サービス(JNDI プロバイダ)のデフォルトの TCP ポートです。

nmsas.server.port.remoting.ejb3=3873

この TCP ポートは，グローバル NNMi 管理サーバー上で実行されるコマンドラインツールが，リモートアクセスのために使用します。

nmsas.server.port.jmx.jrmp=4444

この TCP ポートは，RMI がデータ転送に使用する RMI オブジェクトポート(Java Remote Method Protocol)です。

nmsas.server.port.jmx.rmi=4445

この TCP ポートは，RMI 要求をプールするときに使用されるデフォルトのポートです。

nmsas.server.port.invoker.unified=4446

この TCP ポートは，デフォルトの RMI リモーティングサーバーコネクタポートです。jboss リモーティングサービスによって使用されます。

nmsas.server.port.hq=4457

この TCP ポートは，グローバルネットワーク管理の暗号化されていない通信のために使用します。

nmsas.server.port.hq.ssl=4459

この TCP ポートは，グローバルネットワーク管理の暗号化された通信のために使用します。

nmsas.server.port.ts.recovery=4712

この TCP ポートは，NNMi の内部で使用されるトランザクション・サービス・ポートです。

nmsas.server.port.ts.status=4713

この TCP ポートは，NNMi の内部で使用されるトランザクション・サービス・ポートです。

nmsas.server.port.ts.id=4714

この TCP ポートは，NNMi の内部で使用されるトランザクション・サービス・ポートです。

com.hp.ov.nms.postgres.port=5432

この TCP ポートは，組み込みデータベースが NNMi 管理サーバーとの通信のためにリッスンするポート(PostgreSQL のポート)です。

AUTHOR

nnm.ports was developed by Micro Focus.

FILES

- Windows：%NnmDataDir%Conf¥nnm¥props¥nms-local.properties
- Linux：$NnmDataDir/conf/nnm/props/nms-local.properties

SEE ALSO

ovstatus, ovstart, ovstop.

付録 H.7　nnm.properties

コマンドラインツールの実行に必要なユーザー名およびパスワードを格納するファイル。

SYNOPSIS

nnm.properties

DESCRIPTION

NNMi コマンドラインツールを頻繁に実行する場合は，nnm.properties ファイルを作成しておくことを推奨します。このファイルには，コマンドラインオプションの-u および-p に代わって使用されるユーザー名およびパスワードが格納されます。nnm.properties ファイルを使用することで，パスワードを入力せずに多数のコマンドを実行できます。nnm.properties ファイルには，パスワードが暗号化されて格納されます。このファイルは，nnmsetcmduserpw.ovpl コマンドを実行して作成する必要があります。nnm.properties ファイルは決して手動で編集しないでください。

nnmsetcmduserpw.ovpl コマンドでファイルを作成または編集すると，次のホームディレクトリの配下の”.nnm”サブディレクトリにファイルが配置されます。

- Windows：*drive*:¥Users¥*username*¥.nnm¥
- Linux：~/.nnm/

このファイルの内容は次のように定義する必要があります。

nnm.username

アカウントのユーザー名

nnm.password

アカウントの暗号化されたパスワード

ほとんどのコマンドラインツールは，管理者ロールが割り当てられた管理者権限のユーザーとして実行する必要があります。

AUTHOR

nnm.properties was developed by Micro Focus.

FILES

次の環境変数は，使用するシェルおよびプラットフォームの要件に従って設定される共通パスです。

- Windows：*drive*:¥Users¥*username*¥.nnm¥nnm.properties
- Linux：*$HOME*/.nnm/nnm.properties

SEE ALSO

nnmsetcmduserpw.ovpl.

付録 H.8　incidentconfiguration.format

NNMi データベースにロードできるインシデント構成を格納するファイル。このファイルは nnmincidentcfgdump.ovpl によって作成され，nnmincidentcfgload.ovpl によって NNMi データベースにロードされます。

SYNOPSIS

incidentconfiguration.format

DESCRIPTION

incidentconfiguration.format ファイルは NNMi データベースにロードできるインシデント構成を格納します。このファイルでは，構成内容を表現するために規定されたタグを使用します

それぞれのインシデント構成は，次の４つの構成種別タグから始まる必要があります。

```
*ConfigurationType=MgmtEventConfig
*ConfigurationType=PairwiseConfig
*ConfigurationType=SnmpTrapConfig
*ConfigurationType=SyslogMessageConfig
```

インシデント構成を修正する場合，次の点に注意してください。

```
*(#) 記号はコメントを示します。
*コメントは構成種別タグの前に記載する必要があります。
*(#) 記号が構成データの中に記載された場合，それらは記載されたタグの値の一部として扱われます。
*コメントは NNMi データベースに保存されません。そのため，その後に実行された nnmincidentdump.ovpl コマンドの出力には含まれません。
*構成種別タグの後に記載されたすべてのタグは，そのインシデント構成種別の一部と判断されます。
*(-) で始まるタグは編集することができます。
*(*) で始まるタグは，一度インポートされた後は編集することはできません。
*(OPTIONAL) はそのタグがオプションであることを示します。
*([]) は，タグの値を，特定のフォーマットや，決められた一覧からの値として指定する必要があることを示します。
*”(Direct child tags may occur multiple times)” と注釈が付いたタグには，複数の子となるタグを指定することができます。
*UUID タグはオプションです。UUID は NNMi にデータベース上でのユニークな識別子として利用されます。
*Label タグを指定しない場合，NNMi が自動的に作成します。
*Label を NNMi が Key タグの値から決定できない場合，エラーとなります。
```

次の例外に注意してください。：NNMi は変更されたインシデントの構成に対して，"Customer"作成者キーと，そのラベルを割り当てます。

インシデント構成ファイルをロードする前に，次の作業を実施することを推奨します。

```
1. nnmincidentcfgdump.ovpl -name コマンドを使用して，編集したいインシデントの例を選択してください。
2. 1. の出力を基に，タグの階層を確認してください。
3. ファイルの書式を確認した後，本リファレンスに記載する書式のリストから編集する構成種別を見つけ，編集箇所を決定し，指定された場所に編集したい値を記載してください。
4. nnmincidentcfgload.ovpl -validate コマンドを使用して，編集内容を検証してください。
5. nnmincidentcfgload.ovpl -load コマンドを使用して NNMi データベースにロードし，編集内容をテストしてください。
```

注意：nnmincidentcfgload.ovpl コマンドは，書式に一致しない値に対して，エラーを出力します。

FILES

NNMi は，インシデント構成ファイルの例と，タグフォーマットファイルの正しい書式を次の場所で提供しています。

- Windows：%NnmInstallDir%examples¥nnm¥incidentcfg
- Linux：/opt/OV/examples/nnm/incidentcfg

EXAMPLES

管理イベントを必須タグのみで作成します。

```
*ConfigurationType=MgmtEventConfig
    *Name MinimnalistMgmtConfig
    *Oid .1.3.6.1.4.1.11.2.17.19.2.0.9999
    -Author
        -Key com.customer.author
```

```
    -Category
        -Key com.hp.nms.incident.category.Fault
    -Family
        -Key com.hp.nms.incident.family.Node
    -MessageFormat Custom message format
    -Severity MINOR
```

強化設定を管理イベントに追加します。

```
*ConfigurationType=MgmtEventConfig
    *Name MinimnalistMgmtConfig
    *Oid .1.3.6.1.4.1.11.2.17.19.2.0.9999
    -Author
        -Key com.customer.author
    -Category
        -Key com.hp.nms.incident.category.Fault
    -Family
        -Key com.hp.nms.incident.family.Node
    -MessageFormat Custom message format
    -EnrichConfiguration
        -Enable true
        -Enrichments
            -Enrichment
                -PayloadFilter
                    -Expression ciaName notEquals ”varArg”
```

syslog message インシデント構成の構成種別タグの前に，コメントを追加しています。

```
#
# Insert comments before the configuration type tag
#
# NNMi does not store comments in the NNMi database
#
# This example includes only the required tabs for the syslog message configuration
#
*ConfigurationType=SyslogMessageConfig
    *Name MinimalistSyslogConfig
    -Author
        -Key com.minimal.customer
        -Label MinimalCustomer
    -Category
        -Key com.hp.nms.incident.category.Fault
    -Family
        -Key com.hp.nms.incident.family.AggregatePort
    -MessageFormat $.1.3.6.1.4.1.11937.1.54.5: $.1.3.6.1.4.1.11937.1.4
    -Severity CRITICAL
```

PAIRWISE CONFIGURATION FORMAT

次の例は，PairwiseConfig 構成種別のための書式を示します。

```
*ConfigurationType=PairwiseConfig (ROOT TAG)
    *Name
    -SetOfPairItems (OPTIONAL TAG) (Direct child tags may occur multiple times)
```

```
        -SetOfPairItem (OPTIONAL TAG)
            -FirstInPair
            -FirstParamType
            -SecondInPair
            -SecondParamType
            *UUID (OPTIONAL TAG)
    -Author
        -Key
        -Label (OPTIONAL TAG)
    -DeleteWhenClosed (OPTIONAL TAG)
    -Description (OPTIONAL TAG)
    -Duration
    -Enable (OPTIONAL TAG)
    -FirstIncidentConfigRef
        -Key
        -Type = [MgmtEventConfig, SnmpTrapConfig, SyslogMessageConfig]
    -FirstIncidentName
    -FirstIncidentPayloadFilter (OPTIONAL TAG)
        -Expression [Format = Formatted Expression String]
        *UUID (OPTIONAL TAG)
    -SecondIncidentConfigRef
        -Key
        -Type = [MgmtEventConfig, SnmpTrapConfig, SyslogMessageConfig]
    -SecondIncidentName
    -SecondIncidentPayloadFilter (OPTIONAL TAG)
        -Expression [Format = Formatted Expression String]
        *UUID (OPTIONAL TAG)
```

MANAGEMENT EVENT CONFIGURATION FORMAT

次の例は，MgmtEventConfig 構成種別のための書式を示します。

```
*ConfigurationType=MgmtEventConfig (ROOT TAG)
    *Name
    *Oid
    -Author
        -Key
        -Label (OPTIONAL TAG)
    -Category
        -Key
        -Label (OPTIONAL TAG)
    -Enable (OPTIONAL TAG)
    -ActionConfiguration (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -Actions (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -Action (OPTIONAL TAG)
                -Command (OPTIONAL TAG)
                -CommandType
                -LifecycleState = [Registered, InProgress, Completed, Closed, Dampened]
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
    -DampenConfiguration (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
```

```
        -HourInterval
        -MinuteInterval
        -SecondInterval
        *UUID (OPTIONAL TAG)
        -PayloadFilter (OPTIONAL TAG)
            -Expression [Format = Formatted Expression String]
            *UUID (OPTIONAL TAG)
    -DedupConfiguration (OPTIONAL TAG)
        -ComparisonCriteria
        -CorrelationIncidentConfig (OPTIONAL TAG)
            *Name
        -DedupCount (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        -HourInterval (OPTIONAL TAG)
        -MinuteInterval (OPTIONAL TAG)
        -SecondInterval (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -ComparisonParamList (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -ComparisonParam (OPTIONAL TAG)
                -ParamType (OPTIONAL TAG)
                -ParamValue
                *UUID (OPTIONAL TAG)
    -Description (OPTIONAL TAG)
    -Family
        -Key
        -Label (OPTIONAL TAG)
    -MessageFormat
    -Severity
    -EnrichConfiguration (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -Enrichments (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -Enrichment (OPTIONAL TAG)
                -AssignedTo (OPTIONAL TAG)
                -Category (OPTIONAL TAG)
                    -Key
                    -Label (OPTIONAL TAG)
                -Description (OPTIONAL TAG)
                -Family (OPTIONAL TAG)
                    -Key
                    -Label (OPTIONAL TAG)
                -MessageFormat (OPTIONAL TAG)
                -Nature (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
                -Priority (OPTIONAL TAG)
                    -Key
                    -Label (OPTIONAL TAG)
                -EnrichCias (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -EnrichCia (OPTIONAL TAG)
                        -CiaName
                        -EnrichCiaType
                        -Expression [Format = Formatted Expression String]
                        *UUID (OPTIONAL TAG)
                -Severity (OPTIONAL TAG)
    -SuppressConfiguration (OPTIONAL TAG)
```

```
        -Enable (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -PayloadFilter (OPTIONAL TAG)
            -Expression [Format = Formatted Expression String]
            *UUID (OPTIONAL TAG)
    -InterfaceGroups (OPTIONAL TAG) (Direct child tags may occur multiple times)
        -InterfaceGroup (OPTIONAL TAG)
            -Enable
            *UUID (OPTIONAL TAG)
            -DampenConfiguration
                -Enable (OPTIONAL TAG)
                -HourInterval
                -MinuteInterval
                -SecondInterval
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
            -EnrichConfiguration
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Enrichments (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Enrichment (OPTIONAL TAG)
                        -AssignedTo (OPTIONAL TAG)
                        -Category (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -Description (OPTIONAL TAG)
                        -Family (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -MessageFormat (OPTIONAL TAG)
                        -Nature (OPTIONAL TAG)
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
                        -Priority (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -EnrichCias (OPTIONAL TAG) (Direct child tags may occur multiple tim
es)
                            -EnrichCia (OPTIONAL TAG)
                                -CiaName
                                -EnrichCiaType
                                -Expression [Format = Formatted Expression String]
                                *UUID (OPTIONAL TAG)
                        -Severity (OPTIONAL TAG)
            *InterfaceGroup
            -Ordering
            -ActionConfiguration
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Actions (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Action (OPTIONAL TAG)
                        -Command (OPTIONAL TAG)
                        -CommandType
                        -LifecycleState = [Registered, InProgress, Completed, Closed, Dampen
```

```
ed]
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
            -SuppressConfiguration
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
    -NodeGroups (OPTIONAL TAG) (Direct child tags may occur multiple times)
        -NodeGroup (OPTIONAL TAG)
            -Enable (OPTIONAL TAG)
            *UUID (OPTIONAL TAG)
            -DampenConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                -HourInterval
                -MinuteInterval
                -SecondInterval
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
            -EnrichConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Enrichments (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Enrichment (OPTIONAL TAG)
                        -AssignedTo (OPTIONAL TAG)
                        -Category (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -Description (OPTIONAL TAG)
                        -Family (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -MessageFormat (OPTIONAL TAG)
                        -Nature (OPTIONAL TAG)
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
                        -Priority (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -EnrichCias (OPTIONAL TAG) (Direct child tags may occur multiple tim
es)
                            -EnrichCia (OPTIONAL TAG)
                                -CiaName
                                -EnrichCiaType
                                -Expression [Format = Formatted Expression String]
                                *UUID (OPTIONAL TAG)
                        -Severity (OPTIONAL TAG)
            *NodeGroup
            -Ordering
            -ActionConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
```

```
                *UUID (OPTIONAL TAG)
                -Actions (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Action (OPTIONAL TAG)
                        -Command (OPTIONAL TAG)
                        -CommandType
                        -LifecycleState = [Registered, InProgress, Completed, Closed, Dampen
ed]
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
            -SuppressConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
    -RateConfiguration (OPTIONAL TAG)
        -ComparisonCriteria
        -CorrelationIncidentConfig (OPTIONAL TAG)
            *Name
        -Enable (OPTIONAL TAG)
        -HourInterval (OPTIONAL TAG)
        -MinuteInterval (OPTIONAL TAG)
        -RateCount (OPTIONAL TAG)
        -SecondInterval (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -ComparisonParamList (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -ComparisonParam (OPTIONAL TAG)
                -ParamType (OPTIONAL TAG)
                -ParamValue
                *UUID (OPTIONAL TAG)
```

SNMP TRAP CONFIGURATION FORMAT

次の例は，SnmpTrapConfig 構成種別のための書式を示します。

```
*ConfigurationType=SnmpTrapConfig (ROOT TAG)
    *Name
    *Oid
    -Author
        -Key
        -Label (OPTIONAL TAG)
    -Category
        -Key
        -Label (OPTIONAL TAG)
    -Enable (OPTIONAL TAG)
    -ActionConfiguration (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -Actions (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -Action (OPTIONAL TAG)
                -Command (OPTIONAL TAG)
                -CommandType
                -LifecycleState = [Registered, InProgress, Completed, Closed, Dampened]
                *UUID (OPTIONAL TAG)
```

```
            -PayloadFilter (OPTIONAL TAG)
                -Expression [Format = Formatted Expression String]
                *UUID (OPTIONAL TAG)
-DampenConfiguration (OPTIONAL TAG)
    -Enable (OPTIONAL TAG)
    -HourInterval
    -MinuteInterval
    -SecondInterval
    *UUID (OPTIONAL TAG)
    -PayloadFilter (OPTIONAL TAG)
        -Expression [Format = Formatted Expression String]
        *UUID (OPTIONAL TAG)
-DedupConfiguration (OPTIONAL TAG)
    -ComparisonCriteria
    -CorrelationIncidentConfig (OPTIONAL TAG)
        *Name
    -DedupCount (OPTIONAL TAG)
    -Enable (OPTIONAL TAG)
    -HourInterval (OPTIONAL TAG)
    -MinuteInterval (OPTIONAL TAG)
    -SecondInterval (OPTIONAL TAG)
    *UUID (OPTIONAL TAG)
    -ComparisonParamList (OPTIONAL TAG) (Direct child tags may occur multiple times)
        -ComparisonParam (OPTIONAL TAG)
            -ParamType (OPTIONAL TAG)
            -ParamValue
            *UUID (OPTIONAL TAG)
-Description (OPTIONAL TAG)
-Family
    -Key
    -Label (OPTIONAL TAG)
-GeoCentralForwardConfiguration (OPTIONAL TAG)
    -Enable (OPTIONAL TAG)
    *UUID (OPTIONAL TAG)
    -PayloadFilter (OPTIONAL TAG)
        -Expression [Format = Formatted Expression String]
        *UUID (OPTIONAL TAG)
-MessageFormat
-Severity
-EnrichConfiguration (OPTIONAL TAG)
    -Enable (OPTIONAL TAG)
    *UUID (OPTIONAL TAG)
    -Enrichments (OPTIONAL TAG) (Direct child tags may occur multiple times)
        -Enrichment (OPTIONAL TAG)
            -AssignedTo (OPTIONAL TAG)
            -Category (OPTIONAL TAG)
                -Key
                -Label (OPTIONAL TAG)
            -Description (OPTIONAL TAG)
            -Family (OPTIONAL TAG)
                -Key
                -Label (OPTIONAL TAG)
            -MessageFormat (OPTIONAL TAG)
            -Nature (OPTIONAL TAG)
            *UUID (OPTIONAL TAG)
            -PayloadFilter (OPTIONAL TAG)
                -Expression [Format = Formatted Expression String]
                *UUID (OPTIONAL TAG)
```

```
                -Priority (OPTIONAL TAG)
                    -Key
                    -Label (OPTIONAL TAG)
                -EnrichCias (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -EnrichCia (OPTIONAL TAG)
                        -CiaName
                        -EnrichCiaType
                        -Expression [Format = Formatted Expression String]
                        *UUID (OPTIONAL TAG)
                -Severity (OPTIONAL TAG)
    -SuppressConfiguration (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -PayloadFilter (OPTIONAL TAG)
            -Expression [Format = Formatted Expression String]
            *UUID (OPTIONAL TAG)
    -InterfaceGroups (OPTIONAL TAG) (Direct child tags may occur multiple times)
        -InterfaceGroup (OPTIONAL TAG)
            -Enable
            *UUID (OPTIONAL TAG)
            -DampenConfiguration
                -Enable (OPTIONAL TAG)
                -HourInterval
                -MinuteInterval
                -SecondInterval
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
            -EnrichConfiguration
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Enrichments (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Enrichment (OPTIONAL TAG)
                        -AssignedTo (OPTIONAL TAG)
                        -Category (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -Description (OPTIONAL TAG)
                        -Family (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -MessageFormat (OPTIONAL TAG)
                        -Nature (OPTIONAL TAG)
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
                        -Priority (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -EnrichCias (OPTIONAL TAG) (Direct child tags may occur multiple tim
es)
                            -EnrichCia (OPTIONAL TAG)
                                -CiaName
                                -EnrichCiaType
                                -Expression [Format = Formatted Expression String]
                                *UUID (OPTIONAL TAG)
```

```
                        -Severity (OPTIONAL TAG)
            *InterfaceGroup
            -Ordering
            -ActionConfiguration
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Actions (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Action (OPTIONAL TAG)
                        -Command (OPTIONAL TAG)
                        -CommandType
                        -LifecycleState = [Registered, InProgress, Completed, Closed, Dampen
ed]
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
            -SuppressConfiguration
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
    -NodeGroups (OPTIONAL TAG) (Direct child tags may occur multiple times)
        -NodeGroup (OPTIONAL TAG)
            -Enable (OPTIONAL TAG)
            *UUID (OPTIONAL TAG)
            -DampenConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                -HourInterval
                -MinuteInterval
                -SecondInterval
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
            -EnrichConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Enrichments (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Enrichment (OPTIONAL TAG)
                        -AssignedTo (OPTIONAL TAG)
                        -Category (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -Description (OPTIONAL TAG)
                        -Family (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -MessageFormat (OPTIONAL TAG)
                        -Nature (OPTIONAL TAG)
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
                        -Priority (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -EnrichCias (OPTIONAL TAG) (Direct child tags may occur multiple tim
```

```
es)
                            -EnrichCia (OPTIONAL TAG)
                                -CiaName
                                -EnrichCiaType
                                -Expression [Format = Formatted Expression String]
                                *UUID (OPTIONAL TAG)
                        -Severity (OPTIONAL TAG)
            *NodeGroup
            -Ordering
            -ActionConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Actions (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Action (OPTIONAL TAG)
                        -Command (OPTIONAL TAG)
                        -CommandType
                        -LifecycleState = [Registered, InProgress, Completed, Closed, Dampen
ed]
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
            -SuppressConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
    -RateConfiguration (OPTIONAL TAG)
        -ComparisonCriteria
        -CorrelationIncidentConfig (OPTIONAL TAG)
            *Name
        -Enable (OPTIONAL TAG)
        -HourInterval (OPTIONAL TAG)
        -MinuteInterval (OPTIONAL TAG)
        -RateCount (OPTIONAL TAG)
        -SecondInterval (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -ComparisonParamList (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -ComparisonParam (OPTIONAL TAG)
                -ParamType (OPTIONAL TAG)
                -ParamValue
                *UUID (OPTIONAL TAG)
    -UserRootCause (OPTIONAL TAG)
```

SYSLOG MESSAGE CONFIGURATION FORMAT

次の例は，SyslogMessageConfig 構成種別のための書式を示します。

```
*ConfigurationType=SyslogMessageConfig (ROOT TAG)
    *Name
    -Author
        -Key
        -Label (OPTIONAL TAG)
    -Category
        -Key
```

```
        -Label (OPTIONAL TAG)
    -Enable (OPTIONAL TAG)
    -ActionConfiguration (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -Actions (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -Action (OPTIONAL TAG)
                -Command (OPTIONAL TAG)
                -CommandType
                -LifecycleState = [Registered, InProgress, Completed, Closed, Dampened]
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
    -DampenConfiguration (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        -HourInterval
        -MinuteInterval
        -SecondInterval
        *UUID (OPTIONAL TAG)
        -PayloadFilter (OPTIONAL TAG)
            -Expression [Format = Formatted Expression String]
            *UUID (OPTIONAL TAG)
    -DedupConfiguration (OPTIONAL TAG)
        -ComparisonCriteria
        -CorrelationIncidentConfig (OPTIONAL TAG)
            *Name
        -DedupCount (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        -HourInterval (OPTIONAL TAG)
        -MinuteInterval (OPTIONAL TAG)
        -SecondInterval (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -ComparisonParamList (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -ComparisonParam (OPTIONAL TAG)
                -ParamType (OPTIONAL TAG)
                -ParamValue
                *UUID (OPTIONAL TAG)
    -Description (OPTIONAL TAG)
    -Family
        -Key
        -Label (OPTIONAL TAG)
    -GeoCentralForwardConfiguration (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -PayloadFilter (OPTIONAL TAG)
            -Expression [Format = Formatted Expression String]
            *UUID (OPTIONAL TAG)
    -MessageFormat
    -Severity
    -EnrichConfiguration (OPTIONAL TAG)
        -Enable (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -Enrichments (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -Enrichment (OPTIONAL TAG)
                -AssignedTo (OPTIONAL TAG)
                -Category (OPTIONAL TAG)
                    -Key
```

```
                -Label (OPTIONAL TAG)
            -Description (OPTIONAL TAG)
            -Family (OPTIONAL TAG)
                -Key
                -Label (OPTIONAL TAG)
            -MessageFormat (OPTIONAL TAG)
            -Nature (OPTIONAL TAG)
            *UUID (OPTIONAL TAG)
            -PayloadFilter (OPTIONAL TAG)
                -Expression [Format = Formatted Expression String]
                *UUID (OPTIONAL TAG)
            -Priority (OPTIONAL TAG)
                -Key
                -Label (OPTIONAL TAG)
            -EnrichCias (OPTIONAL TAG) (Direct child tags may occur multiple times)
                -EnrichCia (OPTIONAL TAG)
                    -CiaName
                    -EnrichCiaType
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
            -Severity (OPTIONAL TAG)
-SuppressConfiguration (OPTIONAL TAG)
    -Enable (OPTIONAL TAG)
    *UUID (OPTIONAL TAG)
    -PayloadFilter (OPTIONAL TAG)
        -Expression [Format = Formatted Expression String]
        *UUID (OPTIONAL TAG)
-InterfaceGroups (OPTIONAL TAG) (Direct child tags may occur multiple times)
    -InterfaceGroup (OPTIONAL TAG)
        -Enable
        *UUID (OPTIONAL TAG)
        -DampenConfiguration
            -Enable (OPTIONAL TAG)
            -HourInterval
            -MinuteInterval
            -SecondInterval
            *UUID (OPTIONAL TAG)
            -PayloadFilter (OPTIONAL TAG)
                -Expression [Format = Formatted Expression String]
                *UUID (OPTIONAL TAG)
        -EnrichConfiguration
            -Enable (OPTIONAL TAG)
            *UUID (OPTIONAL TAG)
            -Enrichments (OPTIONAL TAG) (Direct child tags may occur multiple times)
                -Enrichment (OPTIONAL TAG)
                    -AssignedTo (OPTIONAL TAG)
                    -Category (OPTIONAL TAG)
                        -Key
                        -Label (OPTIONAL TAG)
                    -Description (OPTIONAL TAG)
                    -Family (OPTIONAL TAG)
                        -Key
                        -Label (OPTIONAL TAG)
                    -MessageFormat (OPTIONAL TAG)
                    -Nature (OPTIONAL TAG)
                    *UUID (OPTIONAL TAG)
                    -PayloadFilter (OPTIONAL TAG)
                        -Expression [Format = Formatted Expression String]
```

```
                            *UUID (OPTIONAL TAG)
                        -Priority (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -EnrichCias (OPTIONAL TAG) (Direct child tags may occur multiple tim
es)
                            -EnrichCia (OPTIONAL TAG)
                                -CiaName
                                -EnrichCiaType
                                -Expression [Format = Formatted Expression String]
                                *UUID (OPTIONAL TAG)
                        -Severity (OPTIONAL TAG)
            *InterfaceGroup
            -Ordering
            -ActionConfiguration
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Actions (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Action (OPTIONAL TAG)
                        -Command (OPTIONAL TAG)
                        -CommandType
                        -LifecycleState = [Registered, InProgress, Completed, Closed, Dampen
ed]
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
            -SuppressConfiguration
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
    -NodeGroups (OPTIONAL TAG) (Direct child tags may occur multiple times)
        -NodeGroup (OPTIONAL TAG)
            -Enable (OPTIONAL TAG)
            *UUID (OPTIONAL TAG)
            -DampenConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                -HourInterval
                -MinuteInterval
                -SecondInterval
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
            -EnrichConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Enrichments (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Enrichment (OPTIONAL TAG)
                        -AssignedTo (OPTIONAL TAG)
                        -Category (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -Description (OPTIONAL TAG)
                        -Family (OPTIONAL TAG)
                            -Key
```

```
                            -Label (OPTIONAL TAG)
                        -MessageFormat (OPTIONAL TAG)
                        -Nature (OPTIONAL TAG)
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
                        -Priority (OPTIONAL TAG)
                            -Key
                            -Label (OPTIONAL TAG)
                        -EnrichCias (OPTIONAL TAG) (Direct child tags may occur multiple tim
es)
                            -EnrichCia (OPTIONAL TAG)
                                -CiaName
                                -EnrichCiaType
                                -Expression [Format = Formatted Expression String]
                                *UUID (OPTIONAL TAG)
                        -Severity (OPTIONAL TAG)
            *NodeGroup
            -Ordering
            -ActionConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -Actions (OPTIONAL TAG) (Direct child tags may occur multiple times)
                    -Action (OPTIONAL TAG)
                        -Command (OPTIONAL TAG)
                        -CommandType
                        -LifecycleState = [Registered, InProgress, Completed, Closed, Dampen
ed]
                        *UUID (OPTIONAL TAG)
                        -PayloadFilter (OPTIONAL TAG)
                            -Expression [Format = Formatted Expression String]
                            *UUID (OPTIONAL TAG)
            -SuppressConfiguration (OPTIONAL TAG)
                -Enable (OPTIONAL TAG)
                *UUID (OPTIONAL TAG)
                -PayloadFilter (OPTIONAL TAG)
                    -Expression [Format = Formatted Expression String]
                    *UUID (OPTIONAL TAG)
    -RateConfiguration (OPTIONAL TAG)
        -ComparisonCriteria
        -CorrelationIncidentConfig (OPTIONAL TAG)
            *Name
        -Enable (OPTIONAL TAG)
        -HourInterval (OPTIONAL TAG)
        -MinuteInterval (OPTIONAL TAG)
        -RateCount (OPTIONAL TAG)
        -SecondInterval (OPTIONAL TAG)
        *UUID (OPTIONAL TAG)
        -ComparisonParamList (OPTIONAL TAG) (Direct child tags may occur multiple times)
            -ComparisonParam (OPTIONAL TAG)
                -ParamType (OPTIONAL TAG)
                -ParamValue
                *UUID (OPTIONAL TAG)
    -UserRootCause (OPTIONAL TAG)
```

AUTHOR

`incidentconfiguration.format` was developed by Micro Focus.

SEE ALSO

nnmincidentcfgload.ovpl.

nnmincidentcfgdump.ovpl.

付録 H.9　nnmtrapd.conf

IP アドレスおよび OID に基づいてトラップをブロックするためのフィルターファイル

SYNOPSIS

`nnmtrapd.conf`

DESCRIPTION

IP アドレスおよびトラップ OID の両方に基づいてトラップをブロックするためのフィルターを構成するために，`nnmtrapd.conf` ファイルを使用します。

1 行に一つのフィルターを入力してください。各フィルターは，IP アドレス，アドレスの範囲またはワイルドカードと，一つ以上のトラップ OID，トラップ OID の範囲，またはワイルドカードがカンマで区切られて構成されています。

フィルターの形式は次のとおりです。

```
<IP Address, OID[,OID]*>
```

IP Address は，単一の IP アドレスでも，プレフィックス/プレフィックス長の表記法もしくは範囲のワイルドカードの表記法のパターンでもかまいません。 "*" という特別な表記法はすべてのアドレスを示します。同じアドレスでプレフィックス/プレフィックス長の表記法と，範囲のワイルドカード表記法を組み合わせないでください。アドレスの代わりにホスト名を指定しないでください。すべてのフィルターエントリには，一意のアドレス（単一，ワイルドカード，または範囲）が必要です。プレフィックス/プレフィックス長の表記法のアドレスの例は，次のとおりです。

```
10.2.112.0/20

1080:0:a00::/44
```

同じアドレスを，範囲のワイルドカード表記法で表すと次のようになります。

```
10.2.112-127.*

1080:0:a00-a0f:*:*:*:*:*
```

トラップ OID も，範囲またはワイルドカードとして指定することができます。一つの OID では，範囲またはワイルドカードのどちらかを使用します。OID の最後のサブ OID だけを範囲またはワイルドカードとして指定します。".*"という特別な表記法はすべての OID を示しています。例は次のとおりです。

```
.1.3.6.1.4.1.11.2.17.1.0.58915834-58915868
```

```
.1.3.6.1.4.1.11.*
```

linkUp のような一般トラップについては，特定のベンダーをブロックするために，トラップ OID にベンダーのエンタープライズ OID を追加することができます。反対に，すべてのベンダーからの一般トラップをブロックするためには，トラップ OID にワイルドカードを追加する必要があります。

すべてのアドレスからすべてのトラップをブロックすることは許可されていません。このため，次のエントリは無視されます。

```
<*, .*>
```

フィルターの変更を実行中の NNMi に適用するには，次のコマンドを実行します。

```
nnmtrapconfig.ovpl -readFilter
```

EXAMPLES

次の例は，10.2.120 から10.2.127 の範囲のすべてのサブネットからのすべての一般トラップをブロックします。

```
<10.2.120.0/21, .1.3.6.1.6.3.1.1.5.*>
```

次の例は，10.6.112/21 サブネットにあるエンタープライズ OID が.1.3.6.1.4.1.11.2.3.7.11.17 であるすべてのデバイスからの linkUp トラップをブロックします。

```
<10.6.112.0/21, .1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.11.2.3.7.11.17>
```

次の例は，10.6.112/21 サブネットにあるすべてのデバイスからの linkUp トラップをブロックします。

```
<10.6.112.0/21, .1.3.6.1.6.3.1.1.5.4.*>
```

次の例は，一つの IPv6 アドレスからの，OID が .1.3.6.1.4.1.11.2.17 で始まるすべてのトラップ，および認証失敗トラップをブロックします。

```
<1080::8:800:200c:417a, .1.3.6.1.4.1.11.2.17.*, .1.3.6.1.6.3.1.1.5.5.*>
```

AUTHOR

nnmtrapd.conf was developed by Micro Focus.

FILES

- Windows：%NnmDataDir%shared¥nnm¥conf¥nnmtrapd.conf

- Linux：$NnmDataDir/shared/nnm/conf/nnmtrapd.conf

SEE ALSO

nnmtrapconfig.ovpl.

付録 H.10　trapFilter.conf

IP アドレスおよび OID に基づいてトラップをブロックするためのフィルターファイル

SYNOPSIS

trapFilter.conf

DESCRIPTION

IP アドレスおよびトラップ OID の両方に基づいてトラップをブロックするためのフィルターを構成するために，trapFilter.conf ファイルを使用します。これはnnmtrapd.conf ファイルの使用に似ていますが，trapFilter.conf ファイルによってブロックされたトラップは，バイナリトラップストアに格納されず，トラップ率を分析するために使用されないところが違います。トラップ率は，trapFilter.conf ファイルによってブロックされたトラップの影響を受けません。NNMi は，trapFilter.conf ファイルによってブロックされたトラップを格納しません。

1 行に一つのフィルターを入力してください。各フィルターは，IP アドレス，アドレスの範囲またはワイルドカードと，一つ以上のトラップ OID，トラップ OID の範囲，またはワイルドカードがカンマで区切られて構成されています。

フィルターの形式は次のとおりです。

```
<IP Address, OID[,OID]*>
```

IP Address は，単一の IP アドレスでも，プレフィックス/プレフィックス長の表記法もしくは範囲のワイルドカードの表記法のパターンでもかまいません。 "*" という特別な表記法はすべてのアドレスを示します。同じアドレスでプレフィックス/プレフィックス長の表記法と，範囲のワイルドカード表記法を組み合わせないでください。アドレスの代わりにホスト名を指定しないでください。すべてのフィルターエントリには，一意のアドレス（単一，ワイルドカード，または範囲）が必要です。プレフィックス/プレフィックス長の表記法のアドレスの例は，次のとおりです。

```
10.2.112.0/20

1080:0:a00::/44
```

同じアドレスを，範囲のワイルドカード表記法で表すと次のようになります。

```
10.2.112-127.*
```

1080:0:a00-a0f:*:*:*:*:*

トラップ OID も，範囲またはワイルドカードとして指定することができます。一つの OID では，範囲またはワイルドカードのどちらかを使用します。OID の最後のサブ OID だけを範囲またはワイルドカードとして指定します。".*"という特別な表記法はすべての OID を示しています。例は次のとおりです。

.1.3.6.1.4.1.11.2.17.1.0.58915834-58915868

.1.3.6.1.4.1.11.*

linkUp のような一般トラップについては，特定のベンダーをブロックするために，トラップ OID にベンダーのエンタープライズ OID を追加することができます。反対に，すべてのベンダーからの一般トラップをブロックするためには，トラップ OID にワイルドカードを追加する必要があります。

すべてのアドレスからすべてのトラップをブロックすることは許可されていません。このため，次のエントリは無視されます。

<*, .*>

フィルターの変更を実行中の NNMi に適用するには，次のコマンドを実行します。

```
nnmtrapconfig.ovpl -readFilter
```

EXAMPLES

次の例は，10.2.120 から10.2.127 の範囲のすべてのサブネットからのすべての一般トラップをブロックします。

<10.2.120.0/21, .1.3.6.1.6.3.1.1.5.*>

次の例は，10.6.112/21 サブネットにあるエンタープライズ OID が.1.3.6.1.4.1.11.2.3.7.11.17 であるすべてのデバイスからの linkUp トラップをブロックします。

<10.6.112.0/21, .1.3.6.1.6.3.1.1.5.4.1.3.6.1.4.1.11.2.3.7.11.17>

次の例は，10.6.112/21 サブネットにあるすべてのデバイスからの linkUp トラップをブロックします。

<10.6.112.0/21, .1.3.6.1.6.3.1.1.5.4.*>

次の例は，一つの IPv6 アドレスからの，OID が .1.3.6.1.4.1.11.2.17 で始まるすべてのトラップ，および認証失敗トラップをブロックします。

<1080::8:800:200c:417a, .1.3.6.1.4.1.11.2.17.*, .1.3.6.1.6.3.1.1.5.5.*>

AUTHOR

trapFilter.conf was developed by Micro Focus.

FILES

- Windows：%NnmDataDir%shared¥nnm¥conf¥trapFilter.conf
- Linux：$NnmDataDir/shared/nnm/conf/trapFilter.conf

SEE ALSO

nnmtrapconfig.ovpl.

付録 I　各バージョンの変更内容

付録 I.1　13-10 の変更内容

(1)　資料番号（3021-3-L32-10）の変更内容

- Application Experience サービスに関する記述を削除した。
- 次のトラップサーバープロパティを削除した。
  - com.hp.nms.trapd.updateSourcesPeriod
  - com.hp.nms.trapd.notifySourcesPeriod
  - com.hp.ov.nms.trapd.hosted.object.trapstorm.enabled
  - com.hp.ov.nms.trapd.hosted.object.trapstorm.threshold
  - com.hp.ov.nms.trapd.analysis.windowSize
  - com.hp.ov.nms.trapd.analysis.numSources
  - com.hp.ov.nms.trapd.rmiPort
  - com.hp.ov.nms.trapd.connectToWinSNMP
- 高可用性（HA）に関する記述を次のとおり追加・変更した。
  - CLUSTERPRO の追加
  - Windows の場合の説明を，WSFC も含める説明に変更
  - 共有ディスクについての説明の追加
  - HA 構成を解除する順番の説明の追加
- SNMPv3 以外で管理されているノードまたは監視対象外のノードの SNMPv3 トラップを認証するための設定について記述を変更した。
- NNMi の再起動後も設定を維持する手順にあるプロパティファイル内の検索用コメントを変更した。
- Destination Unreachable を受信した場合でも，リトライ処理を継続するように設定する手順を追加した。
- スタンドアロンの NNMi 管理サーバーの IP アドレスを変更した後の説明を変更した。
- サポートする TLS プロトコルに TLSv1.3 を追加した。
- 「バージョン 9・10・11・12 の NNMi からの移行」の説明を，13-10 へバージョンアップするための説明に変更した。
- SNMPv3 通信の認証プロトコルの設定値を変更した。

- nnmsamlconfig.ovpl コマンドの説明を追加した。

付録 I.2　13-00 の変更内容

(1)　資料番号（3021-3-L32）の変更内容

- Linux サーバーに NNMi をインストールする際に必要なライブラリファイル，コマンド，およびパッケージを変更した。
- 次の適用 OS を追加した。
  - Linux 9.1
  - Oracle Linux 9.1

  また，次の適用 OS を削除した。
  - Windows Server 2012
  - CentOS
  - Linux 6.1
  - Oracle Linux 6.1
- Web ブラウザの Internet Explorer のサポート終了に伴い，関連する記述を変更または削除した。
- NNMi が使用する HTTP ポートおよび HTTPS ポートのデフォルト値を変更した。
- systemd を使用しないディストリビューションに関する記述を削除した。
- ノードグループのステータス設定時の［ステータスの設定］フォームのチェックボックスのタイトルを［最も重大なステータスに伝達］に変更した。
- NNMi のインストール時の設定ファイルの内容のコピー先フォルダを変更した。
- NNMi がサポートする SNMPv3 通信の認証プロトコルを追加した。
- ノードグループマップに背景イメージを追加する手順を変更した。
- Subject Alternative Name（SAN）に関する記述を追加した。
- Telnet クライアントが動作する Windows オペレーティングシステムに関する説明を変更した。
- HA 設定の関連製品を使用する場合の注意を追加した。
- NNMi ノードの FQDN 設定変更時に出力されるメッセージについての注意事項を変更した。
- NNMi の自己監視で，ヘルスインシデントを抑制するための設定方法の説明を追加した。
- NNMi 管理サーバーのホスト名またはドメイン名の変更時の注意事項を追加した。
- 「バージョン 9・10・11・12 の NNMi からの移行」の説明を，13-00 へバージョンアップするための説明に変更した。

- JP1/IM3 のサポートに伴い，関連する記述を追加または変更した。
- 環境変数（%jdkdir%および$jdkdir）のパス名を変更した。
- Administrator Commands のリファレンスnnmnotes.ovpl を追加した。
- JKS リポジトリを使った証明書の使用に関する記載を削除した。
- HP-UX または Solaris オペレーティングシステムからの NNMi の移行に関する記載を削除した。
- バージョン 8 以前の JP1/Cm2/Network Node Manager（NNM）に関する記述を削除した。
- JP1/Universal CMDB 10.3 Full に関する記述を削除した。

付録 I.3　12-60 の変更内容

(1)　資料番号（3021-3-E02-30）の変更内容

- Linux の場合に設定できるロケールを変更した。
- システムアカウントのパスワードを NNMi のインストール中に設定するように変更した。
- NNMi のライセンスキーのフォーマット変更についての説明を追加した。
- 恒久ライセンスキーの申請に必要な情報を追加した。
- NNMi 設定ファイルのモデルファイルについての説明を追加した。
- UndefinedSNMPTrap インシデントを複数回出すかどうかを指定する手順を追加した。
- SNMP トラップの MIB データの文字列を正しく解釈し表示する方法についての説明を変更した。
- 適用 OS に Windows Server 2022 を追加した。
- グローバルネットワーク管理の例で使用するスイッチ名を変更した。
- グローバルネットワーク管理のアップグレード手順についての説明を追加した。
- アプリケーションフェイルオーバー構成のアップグレードについて説明を追加した。
- リファレンスページを追加した。

付録 I.4　12-50 の変更内容

(1)　資料番号（3021-3-E02-20）の変更内容

- NNMi 管理サーバーのインストール前チェックリストに kernel.shmall に関する説明を追加した。

- NNMi をインストールする際に設定する環境変数として，LANG を追加した。
- サポートしている証明書の形式について説明を追加した。
- NNMi のインストール時に作成される証明書を置き換える手順を変更した。
- 複数の LDAP ディレクトリサーバーのルート CA 証明書をインポートする場合の説明を追加した。
- Telnet および SSH クライアントの設定対象として，Microsoft Edge を追加した。
- Internet Explorer または Firefox が動作する Windows オペレーティングシステムに関する説明を変更した。
- server エレメントに関する説明を変更した。
- NNMi インストールスクリプトで設定するデフォルトのポート番号を追加した。
- 共有ディスクの使用条件を確認する手順を変更した。
- インシデントの自動トリムを有効にする場合の式を変更した。
- SNMP トラップインシデントの自動トリムを有効にする場合の式を変更した。
- アーカイブファイルのローテーションに関する説明を追加した。
- 「バージョン 9・10・11 の NNMi からの移行」の説明を，12-50 へバージョンアップするための説明に変更した。
- NNMi 管理サーバーが使用するポートの一覧で，ポート 443 の設定の変更について説明を追加した。

付録 I.5　12-10 の変更内容

(1)　資料番号（3021-3-E02-10）の変更内容

- NNMi 管理サーバーのインストール前チェックリストの説明を変更した。
- Internet Explorer を使用する場合の互換表示設定について説明を追加した。
- NNMi が必要とするパッケージとして，次の項目を追加した。
  - fontconfig パッケージ
  - liberation-sans-fonts パッケージ
  - libnsl パッケージ
- Windows システムへの NNMi インストール完了の確認に関する説明を追加した。
- ウイルスチェック除外設定の対象から，次のディレクトリを削除した。
  - /etc/opt/OV
- systemd でサービスを管理しているディストリビューションの場合の説明を追加した。
- インシデントの自動トリム設定に関する説明を追加または変更した。

- NNMi をアンインストールしたあとに削除するディレクトリおよびファイルの対象を変更した。
- 自己署名証明書を生成する手順で，システムからプライベートキーを生成するコマンドを変更した。
- CA 署名証明書を生成する手順で，証明書をキーストアーファイルにインポートするコマンドを変更した。これに伴い，コマンドについてのメモを変更した。
- トラストストアー証明書の所有者 CN について説明を追加した
- Web ブラウザで使用するサードパーティ SSH クライアントを設定する手順を変更した。
- 「バージョン 9・10・11 の NNMi からの移行」の説明を，12-10 へバージョンアップするための説明に変更した。
- JP1/IM2 のインテリジェント統合管理基盤と連携について説明を追加した。
- マニュアルで使用する環境変数についての説明を追加または削除した。
- NodeOrConnectionDown インシデントに関する説明を変更した。
- 次の適用 OS を追加した。
  - CentOS 8.1 以降
  - Linux 8.1 以降
  - Oracle Linux 8.1 以降
  - Windows Server 2019

付録 I.6　12-00 の変更内容

(1)　資料番号（3021-3-E02）の変更内容

- 適用 OS から Windows Server 2008 R2 を削除した。
- NNMi 管理サーバーのインストール前チェックリストの説明を変更した。
- NNMi コンソール用の Web ブラウザの有効化で，Internet Explorer の設定についてのメモを変更した。
- 「NNMi をインストールする（Windows の場合）」の説明に，次の項目を追加した。
  - インストール結果の確認
- NNMi のインストール後に必要な作業として，次の項目を追加した。
  - ウイルス対策ソフトウェアのウイルスチェック除外設定をする
- Linux サーバーに NNMi をインストールしたあとに，言語環境を設定する手順の説明を変更した。
- NNMi と NNMi Advanced の一時試用ライセンスを切り替える場合についての説明を追加した。
- 「NNMi をアンインストールする（Windows の場合）」の手順で，削除が必要な一時ディレクトリおよび一時ファイルに，次のディレクトリおよびファイルを追加した。
  - `%TEMP%¥MicroFocusOvInstaller¥`

- %TEMP%¥NNM_X.X.X_MicroFocusOvInstaller.txt
- %TEMP%¥ovinstallparams.ini
- %TEMP%¥nnmilog¥
- <*drive*>:¥ProgramData¥apregid.com.hpe

- 「NNMi をアンインストールする（Linux の場合）」の手順で，削除が必要な一時ディレクトリおよび一時ファイルに，次のディレクトリおよびファイルを追加した。
  - /var/tmp/jp1nnmi
  - /tmp/MicroFocusOvInstaller
  - /tmp/NNM_X.X.X_MicroFocusOvInstaller.txt
  - /usr/share/apregid.com.hpe
- NNMi ヘルプの詳細情報についての説明を削除した。
- 「仮想環境における通信の設定」のタイトルを，「VMware ハイパーバイザーベースの仮想ネットワークの検出と監視」に変更した。
- ハイパーバイザー上の仮想ネットワークを検出して監視する場合に必要な設定の手順例，および注意事項を削除した。これに伴い，VMware のデフォルト証明書を置き換える手順について，VMware のドキュメントを参照するよう説明を変更した。
- 「ハイパーバイザーとの通信に HTTPS を使用するように NNMi を設定する」の説明から，NNMi 管理サーバーで TLSv1 暗号プロトコルを有効にする手順を削除した。
- 「通信の設定」で，次の説明を追加した。
  - Cisco ACI ネットワークの検出およびモニタリング
  - マルチホーム NNMi 管理サーバー
- インシデントの概念の詳細について，NNMi ヘルプの参照先を変更した。
- トラップサーバープロパティ com.hp.ov.nms.trapd.recvSocketBufSize のデフォルト値を変更した。
- CA 署名証明書を生成する手順で，次の説明を追加または変更した。
  - CSR ファイルの内容を確認するコマンドについてのメモを追加した。
  - トラストストアーの出力形式の内容を変更した。
- nnmkeytool.ovpl コマンドでトラストストアーにアクセスする実行例に，storepass オプションを追加した。
- ldap.properties ファイルの廃止に伴い，関連する記述を変更または削除した。
- NNMi IPv6 管理機能でサポートしていない機能の記述から次の項目を削除した。
  - IPv6 サブネット接続の検出
- アプリケーションフェイルオーバーを設定するための前提条件に，IPv4 アドレスに関する条件を追加した。

- アプリケーションフェイルオーバーの設定手順で，どちらかのノードが複数の IPv4 アドレスまたは NIC を持つ場合の注意事項を追加した。
- HA の設定完了後に，共有ディスクのマウントポイントを変更する場合の注意事項を追加した。
- 「NNMi の保守」から，次の説明を削除した。
  - 厳格に SNMPv3 インフォームを処理するように NNMi を構成する
- 「NNMi セキュリティ」から，次の説明を削除した。
  - 「NNMi が ovjboss バージョン番号を報告しないように設定する」
- 「バージョン 9・10・11 の NNMi からの移行」の説明を，12-00 へバージョンアップするための説明に変更した。

付録 I.7　11-50 の変更内容

(1)　資料番号（3021-3-A72-21）の変更内容

- NNMi 管理サーバーに NNMi をインストールする前のチェック項目として，インストールスクリプトの挙動とその対応に関する説明を追加した。
- Linux サーバーに NNMi をインストールするために必要なコマンドとして，unzip コマンドを追加した。
- 「通信の設定」で，次の説明を変更した。
  - SNMP プロキシを設定する
- NNMi 11-50 を新規インストールした環境で，TLSv1 暗号プロトコルを使用する手順を追加した。
- 「NNMi での証明書の使用」で，「PKCS #12 形式」および「JKS 形式」の証明書リポジトリを使用する場合の説明に変更した。それに伴い，次の説明を削除した。
  - アプリケーションフェイルオーバー機能で CA 証明書を使用する
  - 認証機関を使用するようにグローバルネットワーク管理機能を設定する
- PKCS #12 形式の証明書リポジトリのサポートに伴い，関連する説明を追加または変更した。
- 「NNMi での証明書の使用」の説明を，JKS 形式の証明書リポジトリを使用する場合の説明として変更した。また，次のトピックで，実行するコマンドを変更した。
  - 自己署名証明書の生成
  - CA 署名証明書の生成
  - アプリケーションフェイルオーバー機能で自己署名証明書を使用する
  - 新規証明書を使用するように高可用性クラスタを設定する
  - ディレクトリサービスへの SSL 接続を設定する
- LDAP 設定ファイルとして「nms-auth-config.xml ファイル」を追加した。それに伴い，関連する説明を追加，および既存の LDAP 設定ファイルである「ldap.properties ファイル」の説明を変更した。

- 「タスク 3：ディレクトリサービスからのユーザーアクセスを設定する」および「タスク 5：（「外部モード」の設定だけ）ディレクトリサービスからのグループの取得を設定する」から，次の説明を削除した。
  - Microsoft Active Directory の場合の簡単な方法
  - ほかのディレクトリサービスの場合の簡単な方法
- LDAP 設定ファイルの設定時，混合モードの場合は無視してよいメッセージについて，メモを追加した。
- defaultRole パラメーターの値を変更した場合に実行するコマンドを追加した。
- 「ユーザー識別」から，次の説明を削除した。
  - ディレクトリサービスからの NNMi ユーザーアクセスの設定（詳細な方法）
  - ディレクトリサービスでユーザーを識別する方法の判別（LDAP ブラウザを使用する方法）
  - ディレクトリサービスでユーザーを識別する方法の判別（Web ブラウザを使用する方法）
- LDAP 設定ファイルを「ldap.properties ファイル」から「nms-auth-config.xml」に切り替える手順を追加した。
- 「グローバルネットワーク管理」の「初期準備」で，証明書の設定についての説明を変更した。また，次の説明を追加した。
  - バージョン 11-50 にアップグレードされた NNMi 管理サーバー
- NNMi への HTTP アクセスについての説明を追加した。
- 異なるサーバーセット上の NNMi フェイルオーバー環境にリストアする場合の手順，およびそのリストアをする際に必要なバックアップについて，注記を追加した。
- 「高可用性クラスタに NNMi を設定する」で，次の説明を変更した。
  - 仮想 IP アドレスの変更
  - 物理ホスト名の変更
  - HA クラスタ内の NNMi の設定を解除する
  - パッシブなクラスタノードでの設定解除
  - アクティブなクラスタノードでの設定解除
- 「NNMi の保守」で，次の説明を変更した。
  - 通信設定の構成
  - リモートアクセスには暗号化を必須とするように NNMi を設定する
- Windows NNMi 管理サーバーでのファイルパスの指定について，説明を変更した。
- 「SNMP トラップの管理」で，次の説明を追加した。
  - SNMPv1 トラップまたは SNMPv2c トラップのブロック
- 「NNMi の保守」の「SNMP トラップインシデントの自動トリムを有効にする」で，説明中の値を変更した。

- 「物理センサーステータスの設定」の次のトピックで，プロパティファイルに挿入するテキストを変更した。
  - 物理コンポーネントへの物理センサーステータスの伝達
  - 物理コンポーネントに伝達しない物理センサーステータスの設定
  - 物理センサーステータス値の上書き
- 「NNMi セキュリティ」で，次の説明を変更した。
  - TLS プロトコルの設定
- 「バージョン 9・10・11 の NNMi からの移行」の説明を，11-50 へバージョンアップするための説明に変更した。また，次のトピックに，11-50 にアップグレードした後の手順を追加した。
  - グローバルネットワーク管理のアップグレード手順
  - アプリケーションフェイルオーバー構成の NNMi 11-50 へのアップグレード
- NNMi10-50 で，UCMDB と連携した状態で 11-00 にアップグレートする場合に必要な設定についての説明を削除した。
- 「第 8 編　NNMi との統合編」に，「30.　Restful API」を追加した。
- 環境変数（%jdkdir%および$jdkdir）の説明を追加した。

付録 I.8　11-10 の変更内容

(1)　資料番号（3021-3-A72-10）の変更内容

- 適用 OS に Windows Server 2016 を追加した。
- Linux サーバーに NNMi をインストールする場合に必要なライブラリファイルから，次のファイルを削除した。
  - /usr/lib/libstdc++.so.6
- Linux サーバーに NNMi をインストールしたあとに，言語環境を設定する手順の説明を変更した。
- ライセンス情報を追跡する際の注意事項を追加した。
- 「初期スタートアップの問題」に，次の説明を追加した。
  - NNMi のインストールまたはアップグレードが正常に終了した後，NNMi コンソールが開かない
- SNMP エージェントと Web エージェントが設定されている場合，NNMi は追加のプロトコルが使用できることの説明を追加した。
- 「SNMP アクセス制御」で，SNMPv3 のプライバシプロトコルについての説明を変更した。
- ハイパーバイザー NNMi を検出するには，管理アドレスではなくノード名が必要であることの説明を追加した。
- 仮想環境における通信の設定についての説明を追加した。

- 「通信設定を確認する」に，VMware 通信の場合の説明を追加した。
- 「検出の概念」に，NNMi で検出されたノードの数がライセンスされた容量限界に到達または超えた場合の説明を追加した。
- 「応答のないオブジェクトを削除する」に，仮想マシンノードを削除しない条件についての説明を追加した。
- 「ステータスポーリングの計画」で，次の説明を追加または変更した。
  - NNMi で監視できる項目
  - 監視の停止
  - 監視されないノードへのインタフェース
  - モニタリングの拡張
- ノードグループの説明に，デフォルトのノードグループに含まれる内容として「仮想マシン」を追加した。
- State Poller サービスで収集される情報に，「Web ポーリング」の情報を追加した。
- 「State Poller 稼働状態情報」の表に，「状態アップデータ例外」を追加した。
- 「トラップおよび NNMi インシデント転送でサポートされている方法」の表にある，「転送対象」の説明を変更した。
- 「NNMi コンソール」で，次の説明を追加または変更した。
  - 分析ペインのゲージの設定
  - マップラベルのスケールサイズと境界の設定
  - Loom 図および Wheel 図の自動折りたたみしきい値の設定
- 「NNMi での証明書の使用」で，次の説明を追加または変更した。
  - NNMi 証明書について
  - 既存の証明書と新規の自己署名証明書または CA 署名証明書との置き換え
- 「NNMi の保守」で，次の説明を追加または変更した。
  - MIB ブラウザパラメータの変更
  - レベル 2 オペレータがノードおよびインシデントを削除できるように構成する
  - レベル 2 オペレータがマップを編集できるように構成する
- 「NNMi セキュリティ」に，次の説明を追加した。
  - TLS プロトコルの設定
- 「バージョン 9・10・11 の NNMi からの移行」に，次の説明を追加した。
  - バージョン 11-00 の NNMi 管理サーバーをバージョンアップする
- 「HP-UX または Solaris オペレーティングシステムからの NNMi の移行」に，次の説明を追加した。
  - アプリケーションフェイルオーバー構成の HP-UX または Solaris から Linux への NNMi の変更

  - グローバルマネージャーとリージョナルマネージャーの HP-UX または Solaris から Linux への NNMi の変更
  - 高可用性クラスタ（HA）構成の HP-UX または Solaris から Linux への NNMi の変更
- 新規インストール中に読み込む MIB から，次の MIB を削除した。
  - AX-BFD-MIB
  - AX-BOOTMANAGEMENT-MIB
  - AX-DEVICE-MIB
  - AX-FDB-MIB
  - AX-FLOW-MIB
  - AX-LOGIN-MIB
  - AX-MANAGEMENT-MIB
  - AX-NOTIFICATION
  - AX-OSPF-MIB
  - AX-OSPFV3-MIB
  - AX-QUEUE-MIB
  - AX-SMC-MIB
  - AX-SMCSERVICE-MIB
  - AX-SMI-MIB
  - AX-STATS-MIB
  - AX-SYSTEM-MIB
  - AX-TRACK-MIB
  - AX-VLAN-MIB
  - AX-VRF-MIB
  - AX1230S
  - AX1240S
  - AX2000R
  - AX2430S
  - AX2530S
  - AX3630S
  - AX4630S
  - AX5400S-TRAP
  - AX6300S

- AX7700R-TRAP
- AX7800R
- AX7800R-TRAP
- AX7800S
- AX7800S-TRAP
- AXS-6700S-TRAP
- AXS-AX1240S-TRAP
- AXS-AX1250S-TRAP
- AXS-AX2230S-TRAP
- AXS-AX3630S-TRAP
- AXS-AX3640S-TRAP
- AXS-AX3650S-TRAP
- AXS-AX3830S-TRAP
- AXS-AX4630S-TRAP
- AXS-AX6300S-TRAP
- AXS-AX6600S-TRAP
- Apresia-Series
- Apresia-SeriesLightFMGM
- BFD-TC-STD-MIB
- COMETAGT-AIX
- COMETAGT-LINUX
- COMETAGT-SOLARIS
- COMETAGT-TRU64
- RFC1253-MIB
- cmSmsAgt
- cometAgt
- cometAgtEx
- windowsNTAgt

- NNMi 環境変数の詳細リストから，次の環境変数を削除した。
  - %NNM_SUPPORT% （Windows）
  - $NNM_SUPPORT （Linux）

付録 I.9　11-00 の変更内容

(1)　資料番号（3021-3-A72）の変更内容

- 次の適用 OS を追加した。
  - Windows Server 2008 R2 (x64) SP2
  - CentOS 6.1 以降
  - CentOS 7.1 以降
  - Linux 6.1 (x64)以降
  - Linux 7.1 以降
  - Oracle Linux 6.1 以降
  - Oracle Linux 7.1 以降
  - SUSE Linux 12
- 次の適用 OS を削除した。
  - HP-UX (IPF)
  - Solaris
- 「第 1 編　準備編」に，「1.　インストール前チェックリスト」，「2.　NNMi のインストールとアンインストール」を追加した。
- 「第 2 編　入門編」に，「3.　NNMi 入門」を追加した。
- デバイスと通信するために NNMi で SNMPv3 を使用する際に注意する点についての説明を追加した。
- 「付録」に「付録 A　NNMi の man ページを表示できない場合（Linux)」と「付録 B　新規インストール中に読み込む MIB 一覧」を追加した。
- リンクアグリゲーションを使用した検出についての説明を追加した。
- SNMP トラップを受信したときに NNMi にデバイスをポーリングさせる方法についての説明を追加した。
- NNMi に NAT を実装する方法についての説明を，全面的に書き換えた。
- 静的 NAT のモニタリング，および動的 NAT のモニタリングの設定方法についての説明を追加した。
- ネットワークアドレス変換（NAT）環境で NNMi を配備する手順を追加した。
- 管理アドレスポーリングを有効にした場合の，ICMP 応答と SNMP 応答の組み合わせによって決定される，SNMP エージェントステータスの計算についての説明を追加した。
- リージョナルマネージャーからグローバルマネージャーへのカスタム属性の複製についての説明を追加した。
- IPv6 機能を再度アクティブにする手順を追加した。
- ノードグループ設定をコマンドラインツールを使って自動化する方法を追加した。

- ノードグループマップをコマンドラインツール使って設定する方法を追加した。
- 通信設定をコマンドラインツールで実施する方法を追加した。
- server.properties ファイルの設定を上書きする方法についての説明を追加した。
- SNMPv1 または SNMPv2c を使用して管理されているノードまたは検出されていないノードの SNMPv3 トラップを受信して保存するように NNMi を設定する手順，および Causal Engine がトラップの受け入れを停止する期間を設定する手順を追加した。
- プロキシ SNMP ゲートウェイによって送信されたトラップから元のトラップアドレスを判別する手順を追加した。
- SNMPv1 および SNMPv2c トラップアドレスの順序についての説明を追加した。
- フェイルオーバー時に SNMP トラップの損失を最小限に抑えるのに役立つスタンドアロン NmsTrapReceiver プロセスについての説明を追加した。
- NNMi を使用して任意のノードセットの停止をスケジュールする方法についての説明を追加した。
- ステータスを監視するために，物理センサーステータスとノードセンサーステータスを設定する手順を追加した。
- NNMi を設定して SSLv3 サイファーを有効化または無効化する手順を追加した。
- NNMi のデータ暗号化についての説明を追加した。

付録 I.10　10-50 の変更内容

(1)　資料番号（3021-3-242-20）の変更内容

- インタフェースグループが検出除外インタフェース構成で使用されている場合の説明を追加した。
- 通信の設定に NETCONF を使用したデバイスのサポートの説明を追加した。
- 除外 IP アドレス機能を使ったオブジェクトを検出しない方法の説明を変更した。
- NNMi Northbound インタフェースの説明を追加した。
- 認証機関証明書を生成するときの，システムからプライベートキーを生成するコマンドのパラメータを変更した。
- NNMi と LDAP によるディレクトリサービスの統合方法の説明を変更した。
- オブジェクトのアクセス制限による影響の，マップおよびパスビューの項目についての説明を変更した。
- アプリケーションフェイルオーバー機能の設定方法の説明を変更した。
- アプリケーションフェイルオーバーの NNMi データベースで，削除したスタンバイサーバーを再度同じクラスタに戻すときのコマンドを追加した。
- 通信障害後に再起動した際のアプリケーションフェイルオーバーの制御についての説明を追加した。
- HA クラスタのソフトウェアとして，Symantec Cluster Server（SCS）を追加した。

- HA 設定の注意事項を追加した。
- WSFC の各リソースの設定内容の例を追加した。
- 二次的な根本原因管理イベントに対するアクションを有効化する説明を追加した。
- 新しく作成した作成者を指定して，作成または変更する項目を変更した。
- NNMi 設定およびデータベースをシステム間で移動する場合の SSL 証明書をマージする説明を追加した。
- NNMi 管理サーバーのホスト名またはドメイン名を変える説明を変更した。
- 次の NNMi セキュリティの説明を追加した。
  - 組み込みデータベースツールのパスワードを入力する
  - NNMi が ovjboss バージョン番号を報告しないように設定する

(2) 資料番号（3021-3-343-20）の変更内容

- 次の適用 OS を追加した。
  - Microsoft(R) Windows Server(R) 2012 R2 Datacenter
  - Microsoft(R) Windows Server(R) 2012 R2 Standard
- Windows Server 2008 以降では，リモートデスクトップからコンソールセッションにログインできないため，関連する記述を削除した。
- NNMi 管理サーバーのインストール前チェックリストの説明を変更した。
- 「NNMi をインストールする（Windows の場合）」の説明に，次の項目を追加した。
  - インストール前チェックの実施およびインストール続行可否の確認
- 「NNMi をインストールする（UNIX の場合）」の説明に，次の項目を追加した。
  - インストール前チェックの実施およびインストール続行可否の確認
- NNMi のインストール後に必要な作業として，次の項目を追加した。
  - 言語環境を設定する（UNIX の場合だけ）
  - Java 最大ヒープサイズを確認する
- 「ディスクドライブのセキュリティ設定（Windows の場合）」の手順の説明を変更した。
- 新規インストール中に読み込む MIB に，次の MIB を追加した。
  - AX-BOOTMANAGEMENT-MIB
  - AX-DEVICE-MIB
  - AX-FLOW-MIB
  - AX-LOGIN-MIB
  - AX-NOTIFICATION
  - AX-OSPF-MIB

  - AX-OSPFV3-MIB
  - AX-QUEUE-MIB
  - AX-SMI-MIB
  - AX-STATS-MIB
  - AX-SYSTEM-MIB
  - AX-VRF-MIB
- インタフェースグループが検出除外インタフェース構成で使用されている場合の説明を追加した。
- 通信の設定に NETCONF を使用したデバイスのサポートの説明を追加した。
- 除外 IP アドレス機能を使ったオブジェクトを検出しない方法の説明を変更した。
- NNMi Northbound インタフェースの説明を追加した。
- 認証機関証明書を生成するときの，システムからプライベートキーを生成するコマンドのパラメーターを変更した。
- NNMi と LDAP によるディレクトリサービスの統合方法の説明を変更した。
- オブジェクトのアクセス制限による影響の，マップおよびパスビューの項目についての説明を変更した。
- アプリケーションフェイルオーバー機能の設定方法の説明を変更した。
- アプリケーションフェイルオーバーの NNMi データベースで，削除したスタンバイサーバーを再度同じクラスタに戻すときのコマンドを追加した。
- 通信障害後に再起動した際のアプリケーションフェイルオーバーの制御についての説明を追加した。
- HA クラスタのソフトウェアとして，Symantec Cluster Server（SCS）を追加した。
- HA 設定の注意事項を追加した。
- WSFC の各リソースの設定内容の例を追加した。
- 二次的な根本原因管理イベントに対するアクションを有効化する説明を追加した。
- 新しく作成した作成者を指定して，作成または変更する項目を変更した。
- NNMi 設定およびデータベースをシステム間で移動する場合の SSL 証明書をマージする説明を追加した。
- NNMi 管理サーバーのホスト名またはドメイン名を変える説明を変更した。
- 次の NNMi セキュリティの説明を追加した。
  - 組み込みデータベースツールのパスワードを入力する
  - NNMi が ovjboss バージョン番号を報告しないように設定する

付録 I.11　10-10 の変更内容

(1)　資料番号（3021-3-242-10）の変更内容

- 作成者属性の使用方法の説明を変更した。
- メニューおよびメニュー項目の設定の説明を変更した。
- 監視設定をモニタリングの設定に変更した。
- nnmconfigimport.ovpl コマンドを使用して大量の設定をインポートする場合の注意事項を追加した。
- 次の SNMP 通信の説明を追加した。
  - SNMPv3 通信使用時の暗号方式を変更する
  - 特定のデバイスの SNMP 通信を有効または無効に設定できる
  - SNMP プロキシエージェントを使用した場合の SNMP 通信手順
- テナントを使用した重複アドレスドメインを含んだネットワークの場合の検出についての説明を追加した。
- オブジェクトを検出しない設定に，除外対象 IP アドレスを指定する方法と，除外対象インタフェースグループを指定する方法を追加した。
- フィルタを定義して検出するインタフェース範囲を指定する方法の説明を追加した。
- シードの検出で問題が起こった場合の対処として，該当する IP アドレスをipnolookup.conf ファイルに含める方法の説明を追加した。
- 応答のないオブジェクトを削除する場合の説明を変更した。
- NNMi ステータスポーリングで次の項目を変更した。
  - プロキシサーバーではなく，スイッチに変更した
  - 監視するインタフェースグループとノードグループの設定方法の説明を変更した
- NNMi インシデントについて次の説明を追加または変更した。
  - インシデントの概念
  - トラップおよびインシデント転送
  - 受信済み SNMP トラップ
  - 解決済み管理イベントインシデントに追加される CIA
  - インシデントに対する NNMi の対応方法を計画する
  - トラップログの設定方法
  - インシデントログの設定方法
  - トラップサーバープロパティの設定方法
  - インシデント設定のバッチロード
  - インシデントの評価

- インシデントの調整
- NNMi コンソールについて次の説明を追加または変更した。
  - ノードグループを作成する
  - ノードグループマップを設定する
  - ノードグループを削除する
  - 分析ペインを無効にする
  - デバイスのアイコンをカスタマイズする
  - テーブルビューのリフレッシュレートをオーバーライドする
- NNMi での証明書の使用方法で次の手順を変更した。
  - 認証機関証明書を生成する
- ディレクトリサービスへの SSL 接続の設定の説明を変更した。
- 次の NNMi と LDAP によるディレクトリサービスの統合の説明を変更した。
  - NNMi ユーザーのアクセス情報と設定の方法
  - ディレクトリサービスへのアクセスを設定する
  - ディレクトリサービスのアクセス設定に NNMi のセキュリティモデルを設定する
  - ディレクトリサービス管理者が所有する情報
  - ディレクトリサービス統合のトラブルシューティング
- NNMi グローバルオペレータユーザーグループ（globalops）では，すべてのトポロジオブジェクトだけにアクセス権が与えられることを記載した。
- NAT 環境の設定方法を追加した。
- NNMi のセキュリティおよびマルチテナントの設定の説明を変更した。
- グローバルネットワーク管理の場合，NAT，PAT および NAPT のときの注意事項を追加した。
- 初期準備のファイアウォールの設定で，アクセス可能にしておく必要があるソケットのパラメータを変更した。
- グローバルネットワーク管理で NNMi ウィンドウおよび説明を変更した。
- グローバルネットワーク管理のトラブルシューティングのヒントで次の説明を変更した。
  - グローバルマネージャとリージョナルマネージャの検出情報の同期
- グローバルネットワーク管理環境での NNMi のバージョンアップ手順の説明を変更した。
- グローバルネットワーク管理とアドレス変換プロトコルの説明を追加した。
- NNMi IPv6 管理機能で次の説明を追加または変更した。
  - NNMi IPv6 管理機能の概要
  - NNMi IPv6 管理機能を使用するための必要条件

  - IPv6 管理機能を有効にする
  - IPv6 管理機能を無効にしたあとの IPv6 インベントリ
- アプリケーションフェイルオーバーを設定するための前提条件を追加した。
- 次のアプリケーションフェイルオーバーの設定方法を追加または変更した。
  - アプリケーションフェイルオーバー構成の NNMi を設定する
  - クラスタセットアップウィザードを使用したアプリケーションフェイルオーバーの設定方法
  - アプリケーションフェイルオーバー通信の設定方法
  - アプリケーションフェイルオーバーの動作
  - アプリケーションフェイルオーバーシナリオ
  - アプリケーションフェイルオーバーを無効にする
  - NNMi のバージョンアップ（修正版の適用を含む）
  - NNMi データベースパスワードの変更
- HA の設定で説明を追加または変更した。
  - HA 用 NNMi を設定するための前提条件の検証
  - NNMi HA 設定情報
  - プライマリクラスタノードでの NNMi の設定
  - セカンダリクラスタノードでの NNMi の設定
  - パッシブなクラスタノードでの設定解除
  - アクティブなクラスタノードでの設定解除
- NNMi データのバックアップの説明を変更した。
- NNMi の保守で次の説明を追加または変更した。
  - フォルダのアクセス権限の管理
  - アクションサーバーのキューサイズを変更する
  - インシデントアクションのログ
  - 文字セットエンコードの設定方法
  - レベル 2 オペレータがノードを削除できるように構成する
  - レベル 2 オペレータがマップを編集できるように構成する
  - レベル 2 オペレータがステータスのポーリングおよび設定のポーリングを実行できるように構成する
  - 監視対象外のノードについて SNMPv3 トラップを認証するように NNMi を構成する
  - プロキシ SNMP ゲートウェイによって送信されたトラップからオリジナルトラップアドレスを特定するように NNMi を構成する
  - リモートアクセス時に暗号化を必須とするように NNMi を設定する

  - 厳格に SNMPv3 インフォームを処理するように NNMi を構成する
  - 以前にサポートされていた varbind 順序を保持するように NNMi を構成する
  - データベースポートを変更する
- NNMi 管理サーバーのホスト名，ドメイン名，またはその両方を変更する場合に，新しい証明書で HTTPS 設定を更新する説明を変更した。
- バージョン 9・10-00 の NNMi からの移行で次の説明を追加した。
  - バージョン 10-00 の NNMi 管理サーバーのバージョンアップ
  - バージョン 9 の NNMi 管理サーバーのバージョンアップ
  - NNMi 10-00 以前からのグローバルマネージャとリージョナルマネージャのアップグレードの方法
  - アプリケーションフェイルオーバー構成の NNMi 10-10 へのアップグレードの方法
- バージョン 8 以前の NNM からの移行で次の説明を変更した。
  - SNMP を設定する
  - デバイスプロファイルをカスタマイズする
  - 検出のスケジュールを設定する
  - 自動検出ルールを設定する
  - ポーリング間隔を設定する
  - ポーリングプロトコルを選択する
  - デバイスからのトラップを表示する
- 環境変数のデフォルトの場所を変更した。
- NNMi が使用するポートの一覧を変更した。

(2) 資料番号（3021-3-343-10）の変更内容

- Windows Server 2012 に対応する説明を追加した。
- ノードを検出する場合，ネットワークアドレス変換（NAT）を使用したときの注意事項を追加した。
- クイックスタート設定ウィザードの画面と説明を変更した。
- NNMi をアンインストールする場合の説明を変更した。
- NNMi ヘルプウィンドウおよび［通信の設定］フォームを変更した。
- Windows の場合のディスクドライブのセキュリティ設定の説明を追加および変更した。
- Web ブラウザの有効化の設定項目および手順の説明を変更した。
- Linux への必要なライブラリのインストールの説明を追加および変更した。
- インストールの問題に次の内容を追加した。

- インストール時に，プレインストール手順（フェーズ II）に失敗し，/tmp/nnm_preinstall_phaseII.log ファイルで詳細を確認する必要があることを示すメッセージが表示される
- 新規インストール中に読み込む MIB 一覧の説明を追加および変更した。
- 作成者属性の使用方法の説明を変更した。
- メニューおよびメニュー項目の設定の説明を変更した。
- 監視設定をモニタリングの設定に変更した。
- nnmconfigimport.ovpl コマンドを使用して大量の設定をインポートする場合の注意事項を追加した。
- 次の SNMP 通信の説明を追加した。
  - SNMPv3 通信使用時の暗号方式を変更する
  - 特定のデバイスの SNMP 通信を有効または無効に設定できる
  - SNMP プロキシエージェントを使用した場合の SNMP 通信手順
- テナントを使用した重複アドレスドメインを含んだネットワークの場合の検出についての説明を追加した。
- オブジェクトを検出しない設定に，除外対象 IP アドレスを指定する方法と，除外対象インタフェースグループを指定する方法を追加した。
- フィルタを定義して検出するインタフェース範囲を指定する方法の説明を追加した。
- シードの検出で問題が起こった場合の対処として，該当する IP アドレスをipnolookup.conf ファイルに含める方法の説明を追加した。
- 応答のないオブジェクトを削除する場合の説明を変更した。
- NNMi ステータスポーリングで次の項目を変更した。
  - プロキシサーバーではなく，スイッチに変更した
  - 監視するインタフェースグループとノードグループの設定方法の説明を変更した
- NNMi インシデントについて次の説明を追加または変更した。
  - インシデントの概念
  - トラップおよびインシデント転送
  - 受信済み SNMP トラップ
  - 解決済み管理イベントインシデントに追加される CIA
  - インシデントに対する NNMi の対応方法を計画する
  - トラップログの設定方法
  - インシデントログの設定方法
  - トラップサーバープロパティの設定方法
  - インシデント設定のバッチロード

  - インシデントの評価
  - インシデントの調整
- NNMi コンソールについて次の説明を追加または変更した。
  - ノードグループを作成する
  - ノードグループマップを設定する
  - ノードグループを削除する
  - 分析ペインを無効にする
  - デバイスのアイコンをカスタマイズする
  - テーブルビューのリフレッシュレートをオーバーライドする
- NNMi での証明書の使用方法で次の手順を変更した。
  - 認証機関証明書を生成する
- ディレクトリサービスへの SSL 接続の設定の説明を変更した。
- 次の NNMi と LDAP によるディレクトリサービスの統合の説明を変更した。
  - NNMi ユーザーのアクセス情報と設定の方法
  - ディレクトリサービスへのアクセスを設定する
  - ディレクトリサービスのアクセス設定に NNMi のセキュリティモデルを設定する
  - ディレクトリサービス管理者が所有する情報
  - ディレクトリサービス統合のトラブルシューティング
- NNMi グローバルオペレータユーザーグループ（globalops）では，すべてのトポロジオブジェクトだけにアクセス権が与えられることを記載した。
- NAT 環境の設定方法を追加した。
- NNMi のセキュリティおよびマルチテナントの設定の説明を変更した。
- グローバルネットワーク管理の場合，NAT，PAT および NAPT のときの注意事項を追加した。
- 初期準備のファイアウォールの設定で，アクセス可能にしておく必要があるソケットのパラメーターを変更した。
- グローバルネットワーク管理で NNMi ウィンドウおよび説明を変更した。
- グローバルネットワーク管理のトラブルシューティングのヒントで次の説明を変更した。
  - グローバルマネージャーとリージョナルマネージャーの検出情報の同期
- グローバルネットワーク管理環境での NNMi のバージョンアップ手順の説明を変更した。
- グローバルネットワーク管理とアドレス変換プロトコルの説明を追加した。
- NNMi IPv6 管理機能で次の説明を追加または変更した。
  - NNMi IPv6 管理機能の概要

  - NNMi IPv6 管理機能を使用するための必要条件
  - IPv6 管理機能を有効にする
  - IPv6 管理機能を無効にしたあとの IPv6 インベントリ
- アプリケーションフェイルオーバーを設定するための前提条件を追加した。
- 次のアプリケーションフェイルオーバーの設定方法を追加または変更した。
  - アプリケーションフェイルオーバー構成の NNMi を設定する
  - クラスタセットアップウィザードを使用したアプリケーションフェイルオーバーの設定方法
  - アプリケーションフェイルオーバー通信の設定方法
  - アプリケーションフェイルオーバーの動作
  - アプリケーションフェイルオーバーシナリオ
  - アプリケーションフェイルオーバーを無効にする
  - NNMi のバージョンアップ（修正版の適用を含む）
  - NNMi データベースパスワードの変更
- HA の設定で説明を追加または変更した。
  - HA 用 NNMi を設定するための前提条件の検証
  - NNMi HA 設定情報
  - プライマリクラスタノードでの NNMi の設定
  - セカンダリクラスタノードでの NNMi の設定
  - パッシブなクラスタノードでの設定解除
  - アクティブなクラスタノードでの設定解除
- NNMi データのバックアップの説明を変更した。
- NNMi の保守で次の説明を追加または変更した。
  - フォルダのアクセス権限の管理
  - アクションサーバーのキューサイズを変更する
  - インシデントアクションのログ
  - 文字セットエンコードの設定方法
  - レベル 2 オペレータがノードを削除できるように構成する
  - レベル 2 オペレータがマップを編集できるように構成する
  - レベル 2 オペレータがステータスのポーリングおよび設定のポーリングを実行できるように構成する
  - 監視対象外のノードについて SNMPv3 トラップを認証するように NNMi を構成する
  - プロキシ SNMP ゲートウェイによって送信されたトラップからオリジナルトラップアドレスを特定するように NNMi を構成する

  - リモートアクセス時に暗号化を必須とするように NNMi を設定する
  - 厳格に SNMPv3 インフォームを処理するように NNMi を構成する
  - 以前にサポートされていた varbind 順序を保持するように NNMi を構成する
  - データベースポートを変更する
- NNMi 管理サーバーのホスト名，ドメイン名，またはその両方を変更する場合に，新しい証明書で HTTPS 設定を更新する説明を変更した。
- バージョン 9・10-00 の NNMi からの移行で次の説明を追加した。
  - バージョン 10-00 の NNMi 管理サーバーのバージョンアップ
  - バージョン 9 の NNMi 管理サーバーのバージョンアップ
  - NNMi 10-00 以前からのグローバルマネージャーとリージョナルマネージャーのアップグレードの方法
  - アプリケーションフェイルオーバー構成の NNMi 10-10 へのアップグレードの方法
- バージョン 8 以前の NNM からの移行で次の説明を変更した。
  - SNMP を設定する
  - デバイスプロファイルをカスタマイズする
  - 検出のスケジュールを設定する
  - 自動検出ルールを設定する
  - ポーリング間隔を設定する
  - ポーリングプロトコルを選択する
  - デバイスからのトラップを表示する
- 環境変数のデフォルトの場所を変更した。
- NNMi が使用するポートの一覧を変更した。

付録 J　このマニュアルの参考情報

このマニュアルを読むに当たっての参考情報を示します。

付録 J.1　関連マニュアル

このマニュアルの関連マニュアルを次に示します。必要に応じてお読みください。

- JP1 Version 13 ネットワーク管理 基本ガイド（3021-3-L31）
- JP1 Version 13 JP1/Network Node Manager i Developer's Toolkit ガイド（3021-3-L33）

付録 J.2　このマニュアルでの表記

このマニュアルでは，日立製品およびそのほかの製品の名称を省略して表記しています。製品の正式名称と，このマニュアルでの表記を次の表に示します。

このマニュアルでの表記		正式名称
Firefox		Mozilla Firefox(R)
JP1/IM2		JP1/Integrated Management 2
JP1/IM3		JP1/Integrated Management 3
Linux	Linux 7.1	Red Hat Enterprise Linux(R) Server 7.1（64-bit x86_64）
	Linux 8.1	Red Hat Enterprise Linux(R) Server 8.1（64-bit x86_64）
	RHEL 8.1	
	Linux 9.1	Red Hat Enterprise Linux(R) Server 9.1（64-bit x86_64）
	Oracle Linux 7.1	Oracle Linux(R) Operating System 7.1
	Oracle Linux 8.1	Oracle Linux(R) Operating System 8.1
	Oracle Linux 9.1	Oracle Linux(R) Operating System 9.1
	SUSE Linux 12	SUSE Linux(R) Enterprise Server 12
NNMi	NNMi	JP1/Network Node Manager i
	NNMi Advanced	JP1/Network Node Manager i Advanced

付録 J.3　このマニュアルで使用する英略語

このマニュアルで使用する英略語を，次の表に示します。

このマニュアルでの表記	正式名称
ACL	Access Control List
APA	Active Problem Analyzer
ARP	Address Resolution Protocol
BIND	Berkeley Internet Name Domain
CA	Certification Authority
CIA	Custom Incident Attribute
DNS	Domain Name System
FQDN	Fully Qualified Domain Name
HSRP	Hot Standby Routing Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Security
ICMP	Internet Control Message Protocol
IPF	Itanium(R) Processor Family
ISP	Internet Services Provider
IT	Information Technology
MD5	Message Digest 5
MIB	Management Information Base
NFS	Network File System
NOC	Network Operations Center
REST	REpresentational State Transfer
SCS	Symantec Cluster Server※
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UUID	Universally Unique IDentifier
VCS	Veritas Cluster Server※
VLAN	Virtual LAN
VRRP	Virtual Router Redundancy Protocol
WAN	Wide Area Network

注※

JP1 10-50 以降では，Veritas Cluster Server または Symantec Cluster Server をサポートしています。JP1 10-10 以前では，Veritas Cluster Server をサポートしています。

付録 J.4　このマニュアルで使用する記号

このマニュアルで使用する記号を次に示します。

記 号	説 明
[ ]	メニュー項目やボタンを表します。
[ ] > [ ]	メニュー項目を連続して選択することを表します。
↵	[Enter] キーを押すことを表します。

付録 J.5　KB（キロバイト）などの単位表記について

1KB（キロバイト），1MB（メガバイト），1GB（ギガバイト），1TB（テラバイト）はそれぞれ 1,024 バイト，$1{,}024^2$ バイト，$1{,}024^3$ バイト，$1{,}024^4$ バイトです。

付録 K　用語解説

A

ARP キャッシュ

ARP（アドレス解決プロトコル）キャッシュは，データリンク層（OSI レイヤー 2）アドレスをネットワーク層（OSI レイヤー 3）アドレスにマップするオペレーティングシステムテーブルです。データリンク層アドレスは通常は MAC アドレスですが，ネットワーク層アドレスは通常は IP アドレスです。ルールベースの検出では，NNMi は，検出されたノードで ARP キャッシュエントリ（ならびにほかのテクニック）を使って，現在の検出ルールに照らしてチェックできる追加ノードを見つけます。

C

Causal Engine

因果関係ベースの方法を使って，根本原因解析（RCA）をネットワーク現象に適用する NNMi テクノロジ。Causal Engine RCA のきっかけとなるのは，ステータスポーリング，SNMP トラップ，特定のインシデントの結果として検出された変更など，特定の事象です。Causal Engine は RCA を使って，管理対象オブジェクトのステータスを調べ，これらオブジェクトに関する結論を明確化し，根本原因インシデントを生成します。

H

HA

「高可用性」を参照してください。

HA リソースグループ

Veritas Cluster Server，Symantec Cluster Server，Microsoft Cluster Service などの最新の高可用性環境では，アプリケーションは，アプリケーション自体，その共有ファイルシステム，仮想 IP アドレスのようなリソースの複合物として表されます。リソースは HA リソースグループで構成されます。これはクラスタ環境で実行中のアプリケーションを表します。

I

ICMP

「インターネット制御メッセージプロトコル」を参照してください。

J

JBoss アプリケーションサーバー

Java Platform, Enterprise Edition（Java EE），Enterprise Java Beans（EJB）と組み合わせて使用するアプリケーションサーバープログラムです。

L

L2

「レイヤー 2」を参照してください。

L3

「レイヤー 3」を参照してください。

M

MIB

「管理情報ベース」を参照してください。

N

NNMi

JP1/Network Node Manager i および JP1/Network Node Manager i Advanced の略称です。

ネットワーク管理の支援や統合のために設計されたソフトウェア商品です。ネットワークノードの継続検出，イベントの監視，ネットワーク障害管理といった機能を備えています。主に NNMi コンソールからアクセスします。

NNMi Northbound インタフェース

NNMi インシデントを SNMPv2c トラップとして Northbound アプリケーションに転送する NNMi の機能です。

NNMi 管理サーバー

NNMi ソフトウェアがインストールされ，NNMi プロセスやサービスが実行されるコンピュータシステムのことです。

NNMi コンソール

NNMi ユーザーインタフェース。オペレータや管理者は，NNMi コンソールを使用して NNMi ネットワーク管理タスクを実行できます。

NNM イベント

古い NNM 管理ステーションから NNMi に転送されたイベント用の NNMi 用語。NNMi には，転送されたイベントから NNMi が生成するインシデントを参照するためのインシデントビューがあります。

Northbound アプリケーション

SNMPv2c トラップを受信および処理できる任意のアプリケーションです。

Northbound 転送先

Northbound アプリケーションのトラップ受信コンポーネントへの接続を定義し，NNMi がその Northbound アプリケーションに送信するトラップのタイプを指定する NNMi Northbound インタフェースの設定の 1 つです。

O

OID

「オブジェクト識別子」を参照してください。

ovstart コマンド

NNMi の管理プロセスを起動するためのコマンドです。コマンドプロンプトで起動します。ovstart のリファレンスページを参照してください。

ovstatus コマンド

NNMi が管理するプロセスの現在のステータスを報告するコマンドです。NNMi コンソール（[ツール] ＞ [NNMi ステータス]）またはコマンドプロンプトで起動できます。ovstatus のリファレンスページを参照してください。

ovstop コマンド

NNMi の管理プロセスを停止するためのコマンドです。コマンドプロンプトで起動します。ovstop のリファレンスページを参照してください。

P

Ping スイープ

ICMP ECHO 要求を複数の IP アドレスに送信し，応答するノードにどのアドレスが割り当てられているか調べるネットワークプローブテクニック。ルールベースの検出で有効にすると，NNMi は，設定された IP アドレスの範囲で Ping スイープを使用して，そのほかのノードを検索できます。サービスの拒絶に Ping スイープを使用できるので，ICMP ECHO 要求をブロックするネットワーク管理者もいます。

PostgreSQL

トポロジ，インシデント，設定情報のような情報を保存するために NNMi がデフォルトで使用するオープンソースリレーショナルデータベース。

R

RCA

「根本原因解析」を参照してください。

S

SNMP

「簡易ネットワーク管理プロトコル（SNMP）」を参照してください。

SNMP トラップ

ポーリングを使ったネットワーク管理（SNMP マネージャーからの要求と SNMP エージェントからの応答）は，処理をできるだけ簡単にするための SNMP の設計原則です。しかし，このプロトコルは，SNMP エージェントから SNMP マネージャープロセス（この場合，NNMi）への要請されないメッセージの通信も提供します。要請されないエージェントメッセージは，「トラップ」として知られており，内部状態の変化または障害条件に応答して SNMP エージェントが生成します。NNMi は，受信した SNMP トラップ（[SNMP トラップ] インシデントの参照ビューに表示）からインシデントを生成します。

SNMP トラップストーム

要請されない大量の SNMP エージェントメッセージ。SNMP マネージャープロセス（この場合，NNMi）を圧迫する可能性があります。nnmtrapconfig.ovpl コマンドを使用して NNMi に SNMP トラップストームしきい値を指定できます。受信トラップレートが指定のしきい値レートを超えるとき，NNMi は，トラップレートが再対応レート未満に下がるまでトラップをブロックします。

sysObjectID

「システムオブジェクト ID」を参照してください。

あ

アカウント

「ユーザーアカウント」を参照してください。

アクティブなクラスタノード

アプリケーションフェイルオーバーまたは高可用性設定で NNMi プロセスを現在実行しているサーバーです。

アドレスのヒント

「検出のヒント」を参照してください。

アプリケーションフェイルオーバー

NNMi で，現在アクティブなサーバーが停止した場合に，NNMi のプロセスの制御をスタンバイサーバーに移行するオプション機能です。このオプション機能はユーザーが設定する必要があります。また，JBoss クラスタリングサポートを利用しています。

い

因果関係

あるイベント（原因）と別のイベント（影響）の間の関係を示します。イベント（影響）は最初のイベント（原因）の直接的な結果です。NNMi は，因果関係分析アルゴリズムを使用して，イベントのサイクルを分析し，ネットワーク問題を解決するソリューションを明らかにします。

インシデント

NNMi では，ネットワークに関連する事象の通知は，NNMi コンソールインシデントビューとフォームに表示されます。NNMi には，インシデント属性に基づいてユーザーがインシデントをフィルタできるようにする幾つもの［インシデントの管理］ビューと［インシデントの参照］ビューがあります。ほとんどのインシデントビューには，NNMi（管理イベントと呼ばれることもあります）が直接生成したインシデントが表示されます。NNMi には，SNMP トラップから生成されたインシデントおよび NNM イベントから生成されたインシデントを参照するビューもあります。

インターネット制御メッセージプロトコル

中核的なインターネットプロトコルスイート（TCP/IP）の 1 つです。ICMP ping は，状態ポーリング用の SNMP クエリーとともに NNMi で使用されます。

インタフェース

ネットワークで用いられる各仕様や規約を利用するための論理的な接続端。

インタフェースグループ

NNMi の主要なフィルタテクニックの 1 つ。ただし，グループごとに，グループまたはフィルタ視覚化に設定を適用する目的で，インタフェースはグループにまとめられます。インタフェースグループは，監視の設定，テーブルビューのフィルタ，マップビューのカスタマイズのどれか，またはすべてに使用できます。「ノードグループ」も参照してください。

え

エピソード

NNMi 根本原因解析で，特定の持続時間を指すのに使う用語。この持続時間は一次的な障害によって引き起こされ，その間，二次障害は抑制されるか，または一次的障害の下で相互に関連づけられます。

お

オブジェクト識別子

SNMP で，管理情報ベースデータオブジェクトを識別する数字のシーケンスです。OID は，小数点で分離された数字で構成されます。各数字は，MIB 階層のそのレベルにおける特定のデータオブジェクトを表します。OID は MIB オブジェクト名と同等の数字です。例えば，MIB オブジェクト名 `iso.org.dod.internet.mgmt.mib-2.bgp.bgpTraps.bgpEstablished`はその`OID 1.3.6.1.2.1.15.0.1` と同等です。

か

仮想 IP アドレス

特定のネットワークハードウェアに結び付かれていない IP アドレス。現在のフェイルオーバーまたはロードバランシングのニーズに基づいて，最も該当するサーバーに中断されないネットワークトラフィックを送信するため，高可用性設定で使われます。

仮想ホスト名

仮想 IP アドレスと関連づけられたホスト名。

簡易ネットワーク管理プロトコル（SNMP）

OSI モデルのアプリケーション層（レイヤー 7）で機能する簡易なプロトコルです。リモートユーザーは，このプロトコルによって，ネットワーク要素の管理情報を検査または変更できます。SNMP は，管理対照ノード上のエージェントプロセッサーとネットワーク管理情報を交換するために NNMi が使う主要なプロトコルです。NNMi は，SNMP の最も一般的なバージョンである SNMPv1，SNMPv2c，および SNMPv3 の 3 つをサポートしています。

管理サーバー

NNMi 管理サーバーは，NNMi がインストールされるコンピュータシステムです。NNMi のプロセスとサービスは，NNMi 管理サーバーで稼働します（以前の NNM リビジョンはこのシステムについて「NNM 管理ステーション」という用語を使用していました）。

管理情報ベース

SNMP で，管理対照ネットワークに関するデータの階層的に組織化された集合のことです。管理情報ベース内のデータオブジェクトは管理対照デバイスの特色を参照します。NNMi は，ネットワーク管

理情報を収集する場合，MIB データオブジェクト（「MIB オブジェクト」，「オブジェクト」，「MIB」と呼ばれることもあります）を使用して，管理対象ノードとの間で SNMP クエリーを出し，または SNMP トラップを受け取ります。

く

クイックスタート設定ウィザード

クイックスタート設定ウィザードは，NNMi のインストールが完了した直後に自動的に実行されます。クイックスタート設定ウィザードを使用して，SNMPv1 または SNMPv2c 環境の読み取りコミュニティ文字列を準備したり，検出されるノードの範囲に制限を設定したり，管理者アカウントを設定したりできます。

クラスタ

NNMi の関係では，高可用性テクノロジまたは jboss クラスタ化機能の使用によってリンクされるハードウェアおよびソフトウェアのグループ化のことです。これらは，一緒に機能して，コンポーネントに過剰負荷または障害が発生した場合，機能とデータの連続性を確保します。クラスタ内のコンピュータは一般に高速 LAN 経由でお互いに接続されます。クラスタは，通常，可用性またはパフォーマンス，もしくはその両方を向上させるために導入します。

クラスタメンバーまたはノード

NNMi の関係では，NNMi 高可用性またはアプリケーションフェイルオーバーをサポートするよう設定された，または設定される予定の高可用性または jboss クラスタ内のシステム。

グローバルネットワーク管理

地理的に分散している 1 つ以上のリージョナルマネージャーからのデータを統合する 1 つ以上のグローバルマネージャーを持つ，NNMi の分散型の配備です。

グローバルマネージャー

分散 NNMi リージョンマネージャーサーバーからのデータを統合する，グローバルネットワーク管理配備内の NNMi 管理サーバーです。グローバルマネージャーは，環境全体のトポロジおよびインシデントの統合ビューを提供します。グローバルマネージャーには，NNMi Advanced ライセンスが必要です。

け

結論

NNMi で，管理対象オブジェクト用に Causal Engine がステータスと根本原因インシデントを決定した方法を明らかにする Causal Engine が生成および使用するサポート詳細。

検出シード

「シード」を参照してください。

検出のヒント

SNMP ARP キャッシュクエリー，CDP，EDP，またはその他の検出プロトコルクエリー，または ping スィープを使用して NNMi が見つけた IP アドレスです。NNMi はさらに，検出ヒントとして見つかっ

た IP アドレスについてクエリーを実行し，結果をルールベース検出内の現在の検出ルールに照らしてチェックします。

検出プロセス

NNMi が，ネットワークノードを管理下におくために，これらの情報を収集するプロセス。初期検出は，まずデバイスインベントリの情報を収集し，次にネットワーク接続情報を収集するという 2 つのフェーズのプロセスで実行されます。

最初の検出のあとも検出プロセスは継続されます。つまり，リストに基づいた検出では，シードリスト内のデバイスは，設定が変更されると更新されます。ルールベースの検出では，新しいデバイスは現在の検出ルールに合致すると追加されます。検出プロセスは，NNMi コンソールまたはコマンドラインから，デバイスまたはデバイスセットについてオンデマンドで開始できます。

「スパイラル検出」,「ルールベースの検出」および「リストに基づいた検出」も参照してください。

検出ルール

ルールベース検出プロセスを制限するのに使用される，ある範囲のユーザー定義 IP アドレスかシステムオブジェクト ID（オブジェクト識別子），またはその両方です。検出ルールは，NNMi コンソールの［**自動検出ルール**］の［**検出の設定**］部分に設定します。「ルールベースの検出」も参照してください。

こ

高可用性

このマニュアルでは，設定の一部に障害があっても中断されないサービスを提供するハードウェアおよびソフトウェアの設定のことです。高可用性（HA）とは，コンポーネントに障害があった場合でもアプリケーションを実行し続けるよう冗長コンポーネントを備えた構成を意味します。NNMi は，市販されている幾つかの HA ソリューションの 1 つをサポートするように設定できます。「アプリケーションフェイルオーバー」も参照してください。

コミュニティ文字列

SNMP エージェントで SNMP クエリーを認証するために，SNMPv1 および SNMPv2C システムで使用されるパスワードのような仕組み。コミュニティ文字列は SNMP パケット内のクリアテキストに渡されるので，パケット傍受に対してもろくなります。SNMPv3 は，認証用の強力なセキュリティメカニズムを用意します。

コンソール

「NNMi コンソール」を参照してください。

コントローラ

NNMi アプリケーションフェイルオーバーでの，マスタークラスタの状態を持つクラスタメンバーを表す JGroups 用語。コントローラは，常にクラスタで最も古いメンバーです。

根本原因インシデント

Correlation Nature（相関関係の性質）属性が *Root Cause*（根本原因）に設定されている NNMi インシデント。NNMi は，関連問題の現象が処理されていない場合，根本原因解析（RCA）を使って現象をすぐ解決できる課題として根本原因インシデントを確定します。「根本原因解析」を参照してください。

根本原因解析

NNMi で，根本原因解析（RCA）とは，ネットワーク問題の原因を調べるために NNMi が使う問題解決方法のクラスのことです。根本原因とは，解決されることによって，関連づけられた問題の症状も解決するような問題のことです。NNMi は，次の 2 つの主要な方法で根本原因の識別を使います。根本原因が解決されるまで，すぐに実施できる問題についてユーザーに通知し，二次的問題の現象を報告しないようにします。根本原因を判別すると，管理対象オブジェクトのステータス変更または根本原因インシデント，もしくはその両方の生成が行われることがあります。

NNMi が RCA を使用する例として，管理対象ルーターで障害が発生し，NNMi 管理サーバーから見てルーターの反対側にある管理対象ノードがステータスポーリングクエリーに応答できなくなることが挙げられます。NNMi は RCA を使用し，ステータスポーリング障害が二次的問題の現象であるか調べます。ルーターが根本原因インシデントであることを報告し，根本原因ルーター障害が解決されるまでダウンストリームノードで発生している問題の現象を報告することは差し控えます。

し

シード

ネットワーク検出プロセスの開始点として機能することによって，NNMi のネットワーク検出を補助するネットワークノードのことです。例えば，管理環境内のコアルーターなどがシードになることができます。各シードは，IP アドレスやホスト名によって識別されます。ルールベース検出が設定されていない場合，NNMi の検出プロセスは指定シードのリストベース検出に制限されます。

シード検出

シード，またはシードファイルを基にしたプロセスで，シードとして指定したノードについてだけ検出し，レイヤー 2 の接続情報を返します。シード検出は，特定したクエリーとタスクのネットワークインベントリだけを保守します。自動検出と比べてください。「スパイラル検出」も参照してください。

シードによる検出

「リストに基づいた検出」を参照してください。

システムアカウント

NNMi のインストール時に使うために備わっている特別なアカウントです。NNMi システムアカウントは，インストール終了後は，コマンドラインのセキュリティや復旧目的だけに使用されます。「ユーザーアカウント」と読み比べてください。

システムオブジェクト ID

NNMi で，ネットワーク要素のモデルまたは種類を識別する SNMP オブジェクト識別子の専門化された用語。システムオブジェクト ID は，ネットワーク要素の MIB オブジェクトの一部です。このオブジェクトは，検出の間に個別のノードから NNMi がクエリーします。システムオブジェクト ID によって分類できるネットワーク要素の種類の例には，HP ProCurve スイッチファミリ，HP J8715A ProCurve Switch，HP IPF システム用の HP SNMP エージェントがあります。ほかのベンダーのネットワーク要素も同じようにシステムオブジェクト ID に従って分類できます。システムオブジェクト ID の重要な使用法は NNMi デバイスプロファイルの定義にあります。デバイスプロファイルは，ネットワーク要素の種類がわかると，推定できるネットワーク要素の特徴を指定します。

自動検出

「ルールベースの検出」を参照してください。

障害ポーリング

主要な NNMi 監視アクティビティ。このアクティビティでは，NNMi は，管理対象の各オブジェクトの状態を調べるために，管理対象インタフェース，IP アドレス，SNMP エージェントすべてに関し，ステータス MIB の SNMP 読み取り専用クエリーまたは ICMP ping，もしくはその両方を発行します。ユーザーは，NNMi コンソールの［設定］ワークスペースの［モニタリングの設定］で，さまざまなインタフェースグループ，ノードグループ，ノードすべてについて実行された障害ポーリングの種類をカスタマイズできます。障害ポーリングはステータスポーリングのサブセットです。

状態

NNMi では，一般的に，MIB II `ifAdminStatus`，MIB II `ifOperStatus`，パフォーマンス，または可用性に関連する自己報告された管理対象オブジェクト応答について**状態**という用語を使用します。「ステータス」と読み比べてください。

状態ポーリング

NNMi の State Poller が実行する指令された監視。障害，パフォーマンス，コンポーネント稼働状態，管理対象オブジェクトの可用性データを取得するために ICMP ping と SNMP クエリーを使います。「障害ポーリング」も参照してください。

す

ステータス

NNMi では，全般的な稼働状態を示す管理対象オブジェクトの属性。ステータスは，管理対象オブジェクトの未解決結論から Causal Engine が計算します。「状態」と読み比べてください。

スパイラル検出

NNMi の管理するネットワークのインベントリ，包含，リレーションシップ，接続についての情報などのネットワークトポロジ情報を NNMi が常時更新する処理のことです。「検出プロセス」，「ルールベースの検出」および「リストに基づいた検出」も参照してください。

と

トポロジ（ネットワーク）

ネットワークのノードや接続などが，通信ネットワーク上でどのように配置されているのかを示す図のことです。

トラップ

「SNMP トラップ」を参照してください。

トラップ受信コンポーネント

SNMP トラップを受信する，Northbound アプリケーションの一部分です。

一部のアプリケーションには，SNMP トラップを受信して処理用に別のコンポーネントに転送する，個別にインストール可能なコンポーネントが含まれます。

そのようなコンポーネントがない Northbound アプリケーションの場合，「トラップ受信コンポーネント」は「Northbound アプリケーション」と同義語です。

の

ノード

ネットワーク関係で，ネットワークに接続されているコンピュータシステムやデバイス（プリンタ，ルーター，ブリッジなど）のことです。SNMP クエリーに応答できるノードは最も包括的な情報を NNMi に提供しますが，NNMi は非 SNMP ノードの制限された管理も実行できます。

ノードグループ

NNMi の主要なフィルタテクニックの 1 つ。ただし，グループごとに，グループまたはフィルタの視覚化に設定を適用する目的で，ノードはグループにまとめられます。ノードグループは，監視の設定，テーブルビューのフィルタ，マップビューのカスタマイズのどれか，またはすべてに使用できます。「インタフェースグループ」も参照してください。

は

パブリックキー証明書

ネットワークセキュリティおよび暗号化で使用されます。デジタル署名を組み込み，パブリックキーと識別情報を結合するファイルです。証明書は，パブリックキーが個人または組織に属することの確認に使われます。NNMi は SSL 証明書を使います。これにはクライアントとサーバーの通信の認証と暗号化のために，パブリックキーおよびプライベートキーが含まれています。

ほ

ポート

ネットワークハードウェアで，ネットワークデバイスの情報の受け渡しを行う場所です。

ボリュームグループ

コンピュータストレージ仮想化の用語。1 つの大規模ストレージエリアを形成するよう設定された 1 つまたは複数のディスクドライブ。NNMi がサポートする幾つかの高可用性製品は，共有ファイルシステムでボリュームグループを使用します。

み

未接続インタフェース

NNMi の観点からは，未接続インタフェースはほかのデバイスに接続されていないインタフェースのことです。デフォルトでは，NNMi が監視する未接続インタフェースは IP アドレスのあるものだけであり，［**ルーター**］ノードグループのノードに含まれます。

ゆ

ユーザーアカウント

NNMi では，ユーザーまたはユーザーグループが NNMi にアクセスする方法を提供します。NNMi ユーザーアカウントは NNMi コンソールにセットアップされ，事前定義されたユーザーロールを実装します。「システムアカウント」および「ユーザーロール」を参照してください。

ユーザーロール

NNMi 管理者は，ユーザーアクセス設定の一環として，NNMi の各ユーザーアカウントに定義済みのユーザーロールを割り当てます。ユーザーロールによって，NNMi コンソールにアクセス可能なユーザーアカウント，および各ユーザーアカウントで使用可能なワークスペースとアクションが決まります。NNMi には，プログラムによってあらかじめ定義され，変更することのできない次の階層型ユーザーロールがあります。

- 管理者
- Web サービスクライアント
- オペレータレベル 2
- オペレータレベル 1
- ゲスト

など

「ユーザーアカウント」も参照してください。

り

リージョナルマネージャー

デバイスの検出，ポーリングおよびトラップ受信を行い，情報をグローバルマネージャーに転送する，グローバルネットワーク管理配備内の NNMi 管理サーバーです。

リストに基づいた検出

シードのリストに基づいたプロセス。シードとして指定するノードだけに関する詳細ネットワーク情報を検出し，返します。リストに基づいた検出は，特定したクエリーとタスクのネットワークインベントリだけを保守します。ルールベース検出と比べてください。「検出プロセス」および「スパイラル検出」も参照してください。

領域

NNMi で，タイムアウト値やアクセスクレデンシャルのような通信設定を行うためにグループにまとめられたデバイス。

る

ルール

「検出ルール」を参照してください。

ルールベースの検出

自動検出と呼ばれることがよくあります。NNMi は，ルールベースの検出を使用し，ユーザー指定検出ルールに従って，NNMi がデータベースに追加する必要のあるノードを探し出します。NNMi は，

検出されたノードのデータ内で検出ヒントを探してから，指定の検出ルールに照らしてこれらの候補をチェックします。検出ルールは，NNMi コンソールの［**自動検出ルール**］の［**検出の設定**］部分に設定します。リストベース検出と比べてください。

れ

レイヤー 2

階層化通信モデルである Open Systems Interconnection（OSI）のデータリンク層です。データリンク層では，ネットワークの物理リンクを介してデータの伝送を行います。NNMi レイヤー 2 ビューは，デバイスの物理接続に関する情報を提供します。

レイヤー 3

階層化通信モデルである Open Systems Interconnection（OSI）のネットワーク層です。ネットワーク層は，ネットワーク上の隣接するノードのアドレスの取得，データ伝送経路の選択，サービス品質などに関与します。NNMi レイヤー 3 ビューは，ルーティングの観点から接続に関する情報を提供します。

ろ

ロール

「ユーザーロール」を参照してください。

論理ボリューム

個別のファイルシステムまたはデバイススワップ空間として使えるボリュームグループ内の任意のサイズの容量を指すコンピュータストレージ仮想化の用語。NNMi がサポートする幾つかの高可用性製品は共有ファイルシステムで論理ボリュームを使います。

索引

A

AddressNotResponding インシデント　605
Application_A.log ファイル　469
ARP キャッシュ　128, 928

C

Causal Engine　600, 928
Cisco
　スイッチ　80
　ルーター　80
Cisco ACI ネットワークの検出と監視　111
cluster.log ファイル　469
Cluster Manager　377
Cluster Member　377
com.hp.ov.nms.cluster.timeout.archive　389
ConnectionDown インシデント　605
CPU リソース　160

D

DHCP　28
DiskGroup_A.log ファイル　469
DNS 設定を確認する　34

H

HA　928
HA_nnmhaserver.log ファイル　469
haconfigure.log ファイル　469
HA クラスタ
　IP アドレスの変更　447
　NNMi　421
　アーキテクチャ　413
　概念　413
　起動の問題　463
　共有データ　444
　サポート対象の製品　413
　シナリオ　415
　スクリプト　468
　設定のトラブルシューティング　459
　ファイル　468
HA クラスタ内の NNMi をメンテナンスする　447
HA クラスタの設定解除　452
HA 情報
　NNMi　421
HA 設定　468
　共有ディスク　423
　スクリプト　468
　ファイル　468
　リファレンスページ　416
　ログファイル　469
HA 設定のメンテナンス　446
HA プライマリクラスタノード
　設定情報　423
HA 用のクラスタアーキテクチャ　413
HA リソースグループ　928
　起動できない　462
　設定　423
　説明　413
　停止　454
hostnolookup.conf ファイル　34

I

ICMP　928
　IPv4 アドレス　607
　アドレス監視　151
　トラフィックの無効化　96
ICMP ping　90
InterfaceDown インシデント　605
ipnolookup.conf ファイル　34
IPv4 アドレス　602
IP アドレス
　HA 用に変更　447
　管理サーバー　56
　検出シードの入力　51
　範囲　132

IP アドレス範囲　51

J

JavaScript の有効化　66
JBoss アプリケーションサーバー　928
JBoss ポートの競合　62
JP1/IM3 のインテリジェント統合管理基盤との連携　583

L

L2　928
L3　928
LDAP 設定ファイル　262
Linux への必要なライブラリのインストール（Linux の場合）　39

M

man ページ　416
MIB　928
MIB II 変数　151
MINCAUSE アルゴリズム　600
Mount_A.log ファイル　469

N

NAT　267
NAT 環境の重複 IP アドレスの管理　266
NAT タイプ　269
NAT の利点　268
NETCONF とは何か　105
NETCONF プロトコルの運用　106
NETCONF を使用するデバイスのサポート　105
nms-auth-config.xml ファイル　262
nms-cluster.properties ファイル　381
NmsApa サービス
　Causal Engine　600
　オブジェクトステータスの設定　601
　設定変更　635
　デバイスが発生したトラップ　635
　ネットワーク接続　635
nmsdbmgr サービス
　起動の問題　464
　ディスクフェイルオーバー　466
nnm.envvars.bat コマンド　597
nnm.envvars.sh コマンド　597
nnmbackup.ovpl　473
NnmClusterFailover インシデント　395
NnmClusterStartup インシデント　395
nnmcluster コマンド　381
nnmcommconf.ovpl コマンド　116
nnmconfigexport.ovpl
　設定の XML への出力　543
nnmconfigimport.ovpl コマンド　543
nnmdatareplicator.conf ファイル　468
nnmdatareplicator.ovpl
　スクリプト　468
nnmhaclusterinfo.ovpl スクリプト　468
nnmhaconfigure.ovpl
　スクリプト　468
nnmhadisk.ovpl
　nmsdbmgr のトラブルシューティング　464
nnmhadisk.ovpl コマンド
　nmsdbmgr のトラブルシューティング　464
nnmhadisk.ovpl スクリプト　468
nnmhamonitor.ovpl スクリプト　468
nnmhamscs.vbs スクリプト　468
nnmharg.ovpl スクリプト　468
nnmhargconfigure.ovpl
　コマンド　462
　スクリプト　468
nnmhargconfigure.ovpl コマンド　462
nnmhastart.ovpl スクリプト　468
nnmhastartrg.ovpl　462
nnmhastartrg.ovpl スクリプト　468
nnmhastop.ovpl スクリプト　468
nnmhastoprg.ovpl　468
nnmhaunconfigure.ovpl スクリプト　468
NNMi　928
　データベースの移動　542

NNMi Northbound アプリケーションの接続パラメーター　576
NNMi Northbound インタフェース 563, 564, 928
NNMi Northbound インタフェースで使用される MIB 情報　580
NNMi Northbound インタフェースで使用される SNMP トラップ情報　581
NNMi Northbound インタフェース転送先のステータス情報　580
NNMi Northbound インタフェース統合の内容 577
NNMi Northbound インタフェースの概要　564
NNMi Northbound インタフェースの使用法　566
NNMi Northbound インタフェースのトラブルシューティング　573
NNMi Northbound インタフェースの変更　571
NNMi Northbound インタフェースの無効化　572
NNMi Northbound インタフェースの有効化　565
NNMi 管理サーバー　28, 928
　ドメイン名を変更する　545
　ホスト名を変更する　545
NNMi 管理サーバーをバージョンアップする　552
NNMi 管理サーバーを変更する　544
NNMi クイックスタート設定ウィザード　36
NNMi コンソール　928
　URL　66
　Web ブラウザの有効化　38
　アクセス　66
　サインイン　66
　トランザクションベースの更新　78
NNMi スパイラル検出　70
NNMi 設定移動の準備　541
NNMi 設定およびデータベースのリセット　87
NNMi 設定およびデータベースを移動する　542
NNMi データ暗号化　549
NNMi データベースパスワードの変更　406
NNMi と LDAP によるディレクトリサービスの統合　232
NNMi とディレクトリサービスの統合　232
NNMi との統合
　ディレクトリサービス　232
NNMi に NAT を実装する方法　270
NNMi に NETCONF デバイスの認証情報を設定する　107
NNMi のアンインストール
　(Linux の場合)　58
　(Windows の場合)　57
NNMi のインストール
　ディスク容量が足りない場合　60
NNMi の起動と停止および再起動　399
NNMi の設定の移動　542
NNMi の設定の変更　403
NNMi のバージョンアップ（修正版の適用を含む）399
NNMi のバックアップとリストア　401
NNMi のポート一覧　637
NNMi のライセンスを取得する　55
NNMi へアクセスする　66
NNMi ヘルプへアクセスする　68
NNMi ユーザーアクセス情報　233
NNMi ユーザーグループ　235
NNMi をアンインストールする　57
NNMi をインストールする　41
　(Linux の場合)　45
　(Windows の場合)　41
nnmloadseeds.ovpl コマンド　132
nnmofficialfqdn.ovpl スクリプト　66
nnmrestore.ovpl スクリプト　476
nnmsnmpwalk.ovpl コマンド　62
NNM イベント　928
NodeDown インシデント　605
NodeOrConnectionDown インシデント　605
Nortel
　スイッチ　80
　ルーター　80
Northbound アプリケーション　564, 928
Northbound 転送先　564, 928
nslookup の応答時間の改善　34
nslookup 要求を避ける　34
nsswitch.conf ファイル　34

O

OID　928
ov.conf
　HA 設定　468
ov.conf ファイル　464, 468
ovstart コマンド　62, 394, 928
ovstatus コマンド　928
ovstop コマンド　62, 394, 928

P

ping
　コマンド　606
　要求　151
Ping スイープ　126, 928
Postgres　377
PostgreSQL　928

R

RCA　928
recovery.conf ファイル　389
root 権限　58

S

SNMP　90, 928
　エージェントステータス　602
　監視　151
　コンポーネント稼働状態　151
　設定の調整　118
　対応バージョン　69
　通信　100
　通信の問題　135
　ノードの設定　115
　バージョンの優先　93
　プロトコル　602
　要求　118
SNMPv1 トラップ　279
SNMPv2c トラップ　276
SNMPv3 資格情報　76
SNMPv3 トラップと通知　95
SNMP トラップ　928
SNMP トラップが NNMi に送信する内容を決定する　151
SNMP トラップストーム　928
SNMP プロキシを設定する　103
State Poller
　概念　141
　稼働状態情報　158
　監視できる項目　144
　計画作成　141
　症状　600
　設定　141
　設定の評価　157
　調整　160
　通信設定　116
Symantec Cluster Server
　HA リソースグループ　413
sysObjectID　928

T

TLS プロトコルの設定　548

V

Veritas Cluster Server
　HA リソースグループ　413
　nnmharg.ovpl スクリプト　468
VMware ハイパーバイザーベースの仮想ネットワークの検出と監視　107
Volume_A.log ファイル　469

W

Web ブラウザ
　NNMi コンソールへのアクセス　66
Web ブラウザの有効化　38
Windows Server Failover Cluster
　HA リソースグループ　413
　nnmhamscs.vbs スクリプト　468

X

XML ファイル　87, 543

あ

アーキテクチャ　413
アカウント　928
アクティブ　377
　プロトコル　100
アクティブなクラスタノード　414, 928
アドレスのヒント　928
アプリケーションフェイルオーバー　377, 928
　NNMi 管理サーバーの要件　378
　NNMi の設定　381
　インシデント　395
　クラスタマネージャー〔モード〕　389
　シナリオ　392
　セットアップ　378
　ネットワークレイテンシ/帯域に関する考慮　407
　無効にする　397
アプリケーションフェイルオーバーと NNMi データベース　407
アプリケーションフェイルオーバーの使用　389
アプリケーションフェイルオーバーの動作　389
アプリケーションフェイルオーバーと NNMi Northbound インタフェース　575
暗号化およびユーザーアカウントパスワード　549

い

一時試用ライセンス　55, 56
一時試用ライセンスキー　55
移動
　NNMi 管理サーバー　540
　NNMi 設定　543
　インタフェース　635
イベント転送フィルター　569
イベント領域　473
因果関係　928
インシデント　928
　アプリケーションフェイルオーバー　395
インシデント削除通知　569
インシデント相関処理通知　568
インシデント転送　566
インシデントの概念　162
インシデントの計画　170
インシデントの設定　171
インシデントの調整　177
インシデントの評価　176
インシデントの例　605
インシデントライフサイクル状態変化通知　567
インストール
　インストール前チェックリスト　26
インストールおよび初期スタートアップのトラブルシューティング　60
インストールの問題　60
インストール前チェックリスト　26
　NNMi 管理サーバー　28
　NNMi クイックスタート設定ウィザード　36
インターネット制御メッセージプロトコル　928
インタフェース　928
　HA 設定の仮想ホストネットワーク　423
　移動　635
　管理　612
　グループ　154
　ステータス　602
　設定　86
　操作　610
　モデル　78
インタフェースグループ　928
［インタフェースグループの設定］フォーム　160
［インタフェースグループ］フォーム　147
［インタフェースグループ］ワークスペース　147
インタラクティブモード　389

う

ウイルスチェック除外設定　48

え

エージェント　606
エージェントクエリー
　応答性　606
　無反応　606

エピソード　602, 928
エンドツーエンドの診断　600

お

応答性
　ICMP への IPv4Address　608
オブジェクト〔ステータス設定〕　601
オブジェクト識別子　928
オブジェクトのグループ定義　147
オフラインバックアップ　473
オペレータレベル 1　66
オペレータレベル 2　66
オンラインバックアップ　473

か

解析〔管理対象ノード〕　602
階層〔ノードグループ〕　82
概念
　Causal Engine　600
　HA　413
　ステータスポーリング　141
　設定　74
　通信　91
確認
　SNMP アクセス　115
　SNMP 用に設定されたノード　115
　インタフェースグループ　157
　管理 IP アドレス　116
　順序番号　155
　通信設定　116
　ノードグループ　157
仮想 IP アドレス　928
仮想ホスト〔HA 設定〕
　ネットマスク　423
　ネットワークインタフェース　423
仮想ホストの名前　423
仮想ホスト名　928
カテゴリ
　ステータス　602
稼働
　管理　612
　シャドウの除去　619
　操作　610
　ノード　615
　分散ルーター　616
稼働状態情報　158
簡易ネットワーク管理プロトコル（SNMP）　928
環境変数　596
　MANPATH〔Linux〕　587
　アプリケーションフェイルオーバー　378
　概要　596
　管理　597
監視　146
　拡張　146
　ノード〔ネットワーク〕　160
監視の拡張　146
完全修飾ドメイン名　66
完全修飾ドメイン名の判断　66
管理
　設定変更　635
管理アドレスの優先　95
管理サーバー　928
　DHCP　28
　IP アドレス　56
管理者
　権限　57
管理者権限　57
管理者ロール　66
管理情報ベース　928
管理対象デバイスての NETCONF の有効化と設定　106
管理対象ノードの解析　602
管理対象ノードの数の確認　55

き

起動
　HA メンテナンス後の NNMi　449
　HA リソースグループ　462

起動の問題
nmsdbmgr 464
NNMi 463
基本的な検出方法を選択する 123
キャッシュ〔ARP〕 128
共有 HA データ 444
共有ディスク
データ 444
データファイルのコピー 421
共有ディスクのディレクトリ 444
共有ディスクフォーマット 423
共有ファイルシステムのタイプ〔HA 設定〕 423

く

クイックスタート設定ウィザード 928
URL 51
クイックスタート設定ウィザードを使用する 51
組み込みデータベースツールのパスワードを入力する 547
クラスタ 928
クラスタメンバーまたはノード 928
グループ
インタフェース〔フィルタリング〕 84
事前設定 148
設定 154
ディスク 423
フィルタリング 84
ボリューム 423
目的 80
グローバルネットワーク管理 928
グローバルネットワーク管理と静的 NAT 281
グローバルネットワーク管理と動的 NAT および動的 PAT 285
グローバルマネージャー 928

け

計画作成
ステータスポーリング 141
通信 98
ポーリング間隔 150
継続検出 70
ゲストロール 66
結論 928
権限サーバー 34
検出 69
検出シード 70
検出設定チェックリスト 69
検出モード 70
再スタート 87
自動検出ルール 70
進行状況を確認する 72
スイッチ 137
スパイラル 70, 119
ノードの削除 135
パフォーマンス 118
評価 135
ルーター 137
検出シード 51, 928
検出設定チェックリスト 69
検出と静的 NAT 275
検出と動的 NAT および動的 PAT 284
検出の進行状況 72
検出の調整 139
検出のデメリット 124
検出のヒント 928
検出プロセス 928
検出ルール 928

こ

高可用性 928
高可用性クラスタ 412
恒久ライセンス 55
恒久ライセンスキー 55
恒久ライセンスキーのインストールを準備する 55
恒久ライセンスキーを取得してインストールする 56
更新中
ノード 634

コマンド
nnm.envvars.bat 597
nnm.envvars.sh 597
nnmbackup.ovpl 473
nnmcluster 381
nnmcommconf.ovpl 116
nnmconfigimport.ovpl 543
nnmdatareplicator.conf 445
nnmdatareplicator.ovpl 445
nnmhargconfigure.ovpl 462
nnmloadseeds.ovpl 132
nnmsnmpwalk.ovpl 62
ovstart 62, 394
ovstop 62, 394
ping 606
コマンドラインのセキュリティ 48
コマンドラインモード 389
コミュニティ文字列 69, 928
コンソール 928
コントローラ 928
コンポーネント稼働状態監視 143
根本原因 601
結論を生み出す 600
根本原因インシデント 928
根本原因解析 928

さ

サーバー
NNMi の移動 541
サーバーからスイッチへのリンクアグリゲーション（S2SLA）の検出について 133
サービス〔NmsApa〕
ステータス 601
設定変更 635
デバイスが発生したトラップ 635
ネットワーク接続 635
ノードを更新 634
サービスレベル契約条項 150
再試行
値 98
調整 118
再スタート
HA メンテナンス後の NNMi 449
検出 87
サインイン 66
SNMPv1 トラップまたは SNMPv2c トラップのブロック 499
削減
デフォルトコミュニティ文字列 118
認証失敗 118
削除
検出されたノード 135
作成者属性 77
作成中
オブジェクトグループ定義 147
再使用可能なノードグループ 149
シャドウ 618
サブネットと静的 NAT 281
サブネットと動的 NAT および動的 PAT 285

し

シード 928
ルールベース検出 124
シード検出 928
シードによる検出 928
システム
共有ファイルタイプ〔HA 設定〕 423
リソース 160
システムアカウント 66, 928
システムアカウントのパスワードの設定 39
システムオブジェクト ID 928
システムオブジェクト ID 範囲
自動検出 132
評価 137
事前設定
インタフェースグループ 148
ノードグループ 149

失敗
ネットワークシナリオ 605
自動検出 928
設定 70
自動検出ルール 51, 70
規則 51
自動検出ルールの順序 125
シナリオ
HA クラスタ 415
ネットワーク失敗 605
シャドウ
作成中 618
除去 619
順序〔評価〕 141
順序属性
自動検出ルール 125
ベストプラクティス 77
順序番号〔確認〕 155
障害ポーリング 928
状態 928
状態とステータスの NNMi 計算 288
状態ポーリング 928
証明書
自己署名 66
認証機関 66
初期スタートアップの問題 61
新規インストール中に読み込む MIB 588
シンボリックリンク 60

す

スィープ 126
スイッチ
階層 82
検出 137
デフォルト 137
ノードグループの定義 80
スクリプト
HA 設定 468
nnmbackup.ovpl 472
nnmbackupembdb.ovpl 472
nnmhaclusterinfo.ovpl 468
nnmofficialfqdn.ovpl 66
nnmresetembdb.ovpl 472
nnmrestore.ovpl 472
nnmrestoreembdb.ovpl 472
データをリストアする 476
スタンドアロンの NNMi 管理サーバーの IP アドレスを変更する 544
スタンバイ 377
ステータス 602, 928
SNMP エージェント 602
インタフェース 602
オブジェクト 601
ノード 602
ノードグループ 602
ステータスポーリング
調整 160
ステータスポーリングの開始 158
ステータスポーリングの調整 160
ステータスポーリングを高度化 141
スパイラル検出 70, 119, 928

せ

正式な完全修飾ドメイン名の取得または設定 37
静的 NAT 269
静的 NAT での通信 273
静的 NAT の考慮事項 271
セカンダリ DNS サービス 34
セカンダリクラスタノード 414
接続
操作〔稼働〕 614
操作〔停止〕 613
ルーター〔稼働〕 621
ルーター〔停止〕 621
設定
DNS 34
HA のトラブルシューティング 461
HA を設定する 421

man ページ　416
NNMi 移動の準備　541
NNMi を移動する　543
オブジェクトステータス　601
概念　74
クイックスタート設定ウィザード　36
コミュニティ文字列　69
情報〔NNMi〕　423
スクリプト〔HA クラスタ〕　468
ステータスポーリング　141
ステータスポーリングの評価　157
通信の設定　103
トランザクションベースの更新　78
ネットワーク検出　69
ノード　99
ポーリングの例　143
リストベース検出　128
領域　98
ルールベース検出　128
ログファイル　469
設定ファイルの複製　445
設定領域　473
［設定］ワークスペース
ステータスポーリングの設定　154
ステータスポーリングの評価　157
前提条件
ハードウェア　27
全領域　473

そ

属性
作成者　77
順序　79
ソフトウェア　27
ソフトウェア使用許諾契約書　56

た

帯域に関する考慮　407
対応 Web ブラウザ　28
対応バージョン
SNMP　69
タイムアウト　92
値　100
調整　118
タスク
ネットワーク検出の設定　69

ち

チェックリスト　143
調整
ステータスポーリング　160
通信　118
重複する IP アドレスマッピング　290

つ

通信
概念　91
計画作成　98
設定　103
設定の評価　115
設定領域　98
調整　118
通信設定の構成　486

て

定義　605
定義済みのユーザーロール　66
停止
NNMi〔HA リソースグループ〕　454
NNMi〔HA フェイルオーバーを行わせないため〕　448
管理〔インタフェース〕　611
シャドウの作成〔ノード〕　618
接続　613, 621
操作〔インタフェース〕　609
分散ルーター〔ノード〕　615
ルーター〔ノード〕　621

ディスク
グループ〔HA 設定〕 423
ディレクトリ〔共有ディスク〕 444
データファイルのコピー〔共有ディスク〕 421
フェイルオーバー 466
ディスクグループ 423
ディレクトリサービス内のユーザー名とパスワード 235
データ
共有ディスク 444
収集〔State Poller〕 151
収集の確認 158
データの収集
確認〔ステータスのポーリング〕 158
データベース
トポロジ 141
リセット 87
データベースをバックアップおよびリストアする 481
データをリストアする
スクリプト 476
デーモンモード 389
テスト
通信設定 103
デバイス
発生したトラップ 635
フィルタ 82
デバイスの検出 90
デバイスの通信設定を確認する 116
デバイスを検出から除外 125
デフォルト
検出 120
コミュニティ文字列 118
スイッチ 137
設定 86
ルーター 137
ルールベース検出 132
デフォルト値〔Linux〕
環境変数 597
デフォルト値〔Windows〕
環境変数 597
デメリット
リストベース検出 123
ルールベース検出 124

と

到達可能なノード 620
到達できないノード 620
動的 NAT 269
動的 NAT および動的 PAT の考慮事項 282
動的 NAT および動的 PAT のハードウェアとソフトウェアの要件 284
動的ポートアドレス変換（動的 PAT） 269
トポロジ 70
データベース 141
トポロジ（ネットワーク） 928
ドメインネームシステム 34
トラップ 928
発生中 635
トラップ受信コンポーネント 564, 928
トラップと静的 NAT 276
トラフィック
無効化 96
トラブルシューティング
HA 設定 461
NNMi 固有の HA 463
NNMi コンソールが開かない 63
NNMi コンソールを起動できない 63
インストール 60
検出 62
初期スタートアップ 61

に

認証失敗の削減 118
認証プロファイル 101

ね

ネットマスク〔HA 設定の仮想ホスト〕 423

ネットワーク
基幹　120
失敗のシナリオ　605
接続　135
接続の確認　135
ノード　160
負荷　160
ネットワークアドレス変換（NAT）環境での NNMi の配備　285
ネットワークインタフェース〔HA 設定の仮想ホスト〕　423
ネットワーク監視の設定を確認する　157
ネットワーク検出　70
ネットワーク失敗のシナリオ　605
ネットワーク接続の確認　135
ネットワーク設定の変更　634
ネットワークトポロジ　70
ネットワーク待ち時間　92
ネットワークレイテンシに関する考慮　407

の

ノード　928
監視　160
グループ　154
更新済み　634
更新中　634
削除する　135
シャドウの除去　619
ステータス　602
設定　86, 99
到達可能　620
到達不可能　620
分散ルーター　616
ルーター　621
ノードグループ　928
インタフェースグループ　84
階層　82
確認　157
事前設定　149
ステータス　602
設定　154
定義　80
デバイスフィルター　82
非 SNMP デバイス　146
ノードグループのステータス　84
ノードグループの設定　484
［ノードグループの設定］フォーム　160
ノードグループのメンバーシップ　81
［ノードグループ］フォーム　147
ノードグループマップ設定の構成　485
［ノードグループ］ワークスペース　147
ノードに ping を送る　62

は

バージョン
SNMP 優先　93
バージョンアップ前のデータをバックアップする　480
ハードウェア　27
ハードウェアおよびソフトウェア　27
パスワード
サインイン　66
パソコン〔検出の概念〕　120
バックアップ
イベント領域　473
オフライン　473
オンライン　473
設定領域　473
全領域　473
トポロジ領域　473
バックアップとリストア
方針　479
バックアップの方針　479
パッシブなクラスタノード　414
発生中
トラップ　635
パフォーマンス〔ステータスポーリング〕　158
パブリックキー証明書　928
範囲〔IP アドレス〕　137

ひ

非 SNMP デバイスノードグループ　146
評価
　ステータスポーリング設定　157
　通信設定　115
評価の順序　141
ヒント
　IP アドレス範囲　137
　システム ID 範囲　137
ヒント〔設定のヒント〕
　シード検出　132
　自動検出　132

ふ

ファイアウォール
　ネットワークアクセスの無効化　96
ファイル
　HA クラスタ　468
　HA 設定　468
　HA 用のレプリケーション　445
　hostnolookup.conf　34
　ipnolookup.conf　34
　nms-auth-config.xml　262
　nms-cluster.properties　381
　nnmdatareplicator.conf　468
　nsswitch.conf　34
　ov.conf　464
　XML　543
　クラスタノードで更新されない　462
　システムタイプ　423
　レプリケーション　445
ファイルシステムのタイプ〔HA 設定〕　423
フィルタ
　デバイス　82
フィルタリング
　インタフェースグループ　84
　ノードグループ　80
フェイルオーバー〔ディスク〕　466
フォーム
　インタフェースグループ　147
　［インタフェースグループの設定］　160
　ノードグループ　147
　［ノードグループの設定］　160
　モニタリングの設定　141, 160
複数　101
複数の DNS サーバーが使用される場合　34
不合格
　認証の削減　118
不十分なディスク容量　60
プライベート IP アドレスの範囲　290
プライマリクラスタノード　414
プリンタ〔検出の概念〕　120
フローモデル〔タスク〕　75
プロトコル
　SNMP　602
　アクティブ　100
　通信　91
　ポーリング　96
プロファイル〔デバイス〕
　概念　82

へ

ページ　416
ベストプラクティス
　NNMi 設定移動の準備　541
　オブジェクトグループ定義　147
　既存の設定を保存する　76
　再使用可能なノードグループ　149
　作成者属性　77
　順序属性　79
　順序番号の確認　155
　短いポーリング間隔　150
変更
　管理　635
　ネットワーク　635
変数〔MIB II〕　151

ほ

ポイント〔マウント〕　423
包含〔ノードグループ〕　82
方法
　リストベース検出　123
　ルールベースの検出　124
ポート　928
　JBoss ポートの競合　62
ポート一覧　637
ポーリング
　開始　158
　間隔の計画作成　150
　設定の例　143
　チェックリスト　143
　調整〔ステータス〕　160
　パフォーマンスの評価　158
　プロトコル　96
ホスト〔HA 設定用の仮想〕
　NNMi　423
ホスト名〔HA 用に変更する場合〕　447
ホスト名の変更
　NNMi　447
保存
　既存　76
ボリュームグループ　414, 423, 928

ま

マウントポイント　423
待ち時間〔ネットワーク〕　92
末端ノード〔検出の概念〕　120
マルチホーム NNMi 管理サーバー　113

み

未接続インタフェース　928

む

無効化
　SNMP　100
　トラフィック　96
無反応
　ICMP への IPv4Address　607

め

メモリリソース　160
メリット
　リストベース検出　123
　ルールベース検出　124
メンテナンスモード　446

も

モデル
　ユーザーインタフェース　78
モニタリング
　設定　86
［モニタリングの設定］フォーム　147, 155
　ステータスポーリングの調整　160
　説明　141
　ポーリングの種類と間隔の設定　147
モニタリングの設定フォーム　141
問題〔HA の起動〕
　nmsdbmgr　464
　NNMi　463

ゆ

有効な IP アドレス範囲　51
ユーザー　66
　サインイン　66
ユーザーアカウント　928
ユーザーインタフェースモデル　78
ユーザーロール　928
　システムアカウント　66
優先
　SNMP のバージョン　93

よ

要求〔SNMP/ICMP 要求〕　118
用語集
　HA　414

ら

ライセンス　55
　限度　124
ライセンスの種類の確認　55
ライセンスの追加　135
ラウンドロビン DNS　34

り

リージョナルマネージャー　928
リージョナルマネージャーからグローバルマネージャーへのカスタム属性の複製　352
リストア
　スクリプト　476
　ファイルシステムだけ　480
リストアの方針　479
リストに基づいた検出　928
リストベース検出
　概要　123
リセット
　設定　87
リソース〔システム〕　160
リソースグループ　423
リモート Northbound アプリケーション　575
領域　928
　通信設定領域　98
リリースノート　27
リンクアグリゲーションの検出　133

る

ルーター
　階層　82
　監視　146
　検出　137
　デフォルト　137
　ノードグループの定義　80
ルール　928
ルール〔自動検出〕
　順序　125
ルールベースの検出　928
　概要　124

れ

例
　アプリケーションフェイルオーバー　392
　ノードグループの設定　154
　ポーリング設定　143
レイヤ 2　928
レイヤ 3　928

ろ

ローカル Northbound アプリケーション　575
ロール　66, 928
ログファイル〔HA クラスタ〕
　設定　469
論理ボリューム　414, 928

わ

ワークスペース
　インタフェースグループ　147
　ステータスポーリングの設定　154
　ノードグループ　147

株式会社 日立製作所
〒 100-8280　東京都千代田区丸の内一丁目 6 番 6 号