HITACHI
Inspire the Next

JP1 Version 10

JP1/Cm2/Network Node Manager i　セットアップガイド

解説・操作書

3021-3-242-20

JP1 Version 10

前書き

■ 対象製品

適用 OS：Windows Server 2008，Windows Server 2012

P-2942-82A4 JP1/Cm2/Network Node Manager i 10-50

P-2942-83A4 JP1/Cm2/Network Node Manager i Advanced 10-50

適用 OS：Linux 6

P-8242-82A1 JP1/Cm2/Network Node Manager i 10-50

P-8242-83A1 JP1/Cm2/Network Node Manager i Advanced 10-50

適用 OS：HP-UX (IPF)

P-1J42-82A1 JP1/Cm2/Network Node Manager i 10-50

P-1J42-83A1 JP1/Cm2/Network Node Manager i Advanced 10-50

適用 OS：Solaris

P-9D42-82A1 JP1/Cm2/Network Node Manager i 10-50

P-9D42-83A1 JP1/Cm2/Network Node Manager i Advanced 10-50

■ 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

Acrobat は，Adobe Systems Incorporated（アドビシステムズ社）の商標です。

AMD は，Advanced Micro Devices,Inc.の商標です。

Cisco は，Cisco Systems, Inc. またはその関連会社の米国およびその他の国における登録商標または商標です。

Firefox は Mozilla Foundation の登録商標です。

HP-UX は，Hewlett-Packard Development Company, L.P.のオペレーティングシステムの名称です。(HP 9000 コンピュータ上の HP-UX リリース 10.20 以降および HP-UX リリース 11.00 以降（32 ビットおよび 64 ビット両方の環境）は，すべて Open Group UNIX 95 製品です。)

HP Serviceguard は，Hewlett-Packard Development Company, L.P.の商品名称です。

Internet Explorer は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Itanium は，アメリカ合衆国およびその他の国における Intel Corporation の商標です。

JBoss および Hibernate は，Red Hat, Inc.の登録商標です。

Kerberos は，マサチューセッツ工科大学（MIT：Massachusetts Institute of Technology）で開発されたネットワーク認証のプロトコルの名称です。

Linux は，Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

Microsoft は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Microsoft Office および Excel は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Netscape は，AOL Inc.の登録商標です。

Nokia は，ノキア・コーポレーションの登録商標です。

Oracle と Java は，Oracle Corporation 及びその子会社，関連会社の米国及びその他の国における登録商標です。

Oracle テクノロジの制限された権限に関する通知

DOD FAR 補足規定に従って供給されたプログラムは，「商用コンピュータソフトウェア」であり，関連文書を含むプログラムの使用，複製，および公開は，該当する Oracle 使用許諾契約に記載されている使用許諾の制限に従うものとします。そうでなければ，連邦調達規則に従って供給されたプログラムは，「制限されたコンピュータソフトウェア」であり，関連文書を含むプログラムの使用，複製，および公開は，FAR 52.227-19，『商用コンピュータソフトウェア - 制限された権限』（1987 年 6 月）に記載されている制限に従うものとします。Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

PostgreSQL は，PostgreSQL Global Development Group が提唱する，オープンソースのオブジェクトリレーショナルデータベース管理システムの名称です。

Red Hat は，米国およびその他の国で Red Hat,Inc. の登録商標もしくは商標です。

すべての SPARC 商標は，米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標がついた製品は，米国 Sun Microsystems, Inc. が開発したアーキテクチャに基づくものです。

Symantec は，Symantec Corporation の米国およびその他の国における商標または登録商標です。

UNIX は，The Open Group の米国ならびに他の国における登録商標です。

Veritas および Veritas Storage Foundation は，Symantec Corporation の米国およびその他の国における商標または登録商標です。

Windows は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

Windows Server は，米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。

その他記載の会社名，製品名は，それぞれの会社の商標もしくは登録商標です。

その他製品名などの固有名詞は各社の商品名，商標および登録商標です。

プログラムプロダクト「P-9D42-82A1，P-9D42-83A1」には，Oracle Corporation またはその子会社，関連会社が著作権を有している部分が含まれています。

プログラムプロダクト「P-9D42-82A1，P-9D42-83A1」には，UNIX System Laboratories,Inc.が著作権を有している部分が含まれています。

■ 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
Excel			Microsoft(R) Office Excel
Internet Explorer			Microsoft(R) Internet Explorer(R)
			Windows(R) Internet Explorer(R)
Windows	Windows 7		Microsoft(R) Windows(R) 7 Enterprise
			Microsoft(R) Windows(R) 7 Professional
			Microsoft(R) Windows(R) 7 Ultimate
	Windows Server 2008	Windows Server 2008	Microsoft(R) Windows Server(R) 2008 Enterprise
			Microsoft(R) Windows Server(R) 2008 Standard
		Windows Server 2008 R2	Microsoft(R) Windows Server(R) 2008 R2 Datacenter
			Microsoft(R) Windows Server(R) 2008 R2 Enterprise
			Microsoft(R) Windows Server(R) 2008 R2 Standard
	Windows Server 2012	Windows Server 2012	Microsoft(R) Windows Server(R) 2012 Datacenter
			Microsoft(R) Windows Server(R) 2012 Standard
		Windows Server 2012 R2	Microsoft(R) Windows Server(R) 2012 R2 Datacenter
			Microsoft(R) Windows Server(R) 2012 R2 Standard
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)
この製品には，Trantor Standard Systems Inc.が開発したソフトウェアが組み込まれています。
(http://www.trantor.ca)

■ 発行

2014 年 9 月　3021-3-242-20

■ 著作権

変更内容

変更内容（3021-3-242-20）　JP1/Cm2/Network Node Manager i 10-50，JP1/Cm2/Network Node Manager i Advanced 10-50

追加・変更内容	変更個所
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。	2.6.4
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。	3.3.2，3.3.2(1)，3.3.2(2)，3.3.2(3)，3.3.2(4)
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。	4.2.7
NNMi Northbound インタフェースの説明を追加した。	6.1.2，6.1.3，25.，表 C-2，付録 F
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。	8.2
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。	10.1，10.2.9，10.3
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。	12.1
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。	16.2，16.3，16.3.1，16.3.2，16.3.3
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。	16.4.1
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。	16.7.2(2)
HA クラスタのソフトウェアとして，Symantec Cluster Server（SCS）を追加した。	17.1，17.4.2(1)，17.4.4，17.6.2，17.7.2，17.7.3，17.8.1，17.8.3，17.9.2，17.9.3，付録 E.3，付録 F
HA 設定の注意事項を追加した。	17.3
WSFC の各リソースの設定内容の例を追加した。	17.4.3(1)
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。	19.20
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。	20.1
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。	20.2
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。	20.5

追加・変更内容	変更個所
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	21.，21.1，21.2

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/Cm2/Network Node Manager i および JP1/Cm2/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，およびバージョン 8 以前の JP1/Cm2/Network Node Manager（以降，NNM と省略します）から移行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

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。

このマニュアルでは，NNMi をインストール済みであること，コミュニティ文字列の設定，ネットワークノードの限られた範囲の検出設定，初期管理者アカウントの作成のような，初期設定作業に慣れていることを仮定しています。これら作業の詳細は，マニュアル「JP1/Cm2/Network Node Manager i　インストールガイド」を参照してください。

■ 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

使用する前に確認する項目について説明しています。

第 2 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 3 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 4 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 5 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 6 編　移行編

バージョン 10 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 7 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。

目次

第 7 編　NNMi との統合編

1 ハードウェアとソフトウェアの要件

この章では，NNMi の対応ハードウェアとソフトウェアについて説明します。

1.1　対応ハードウェアとソフトウェア

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。なお，このマニュアルはマニュアル「*JP1/Cm2/Network Node Manager i インストールガイド*」に従ってインストールが済んでいることを前提としています。

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。

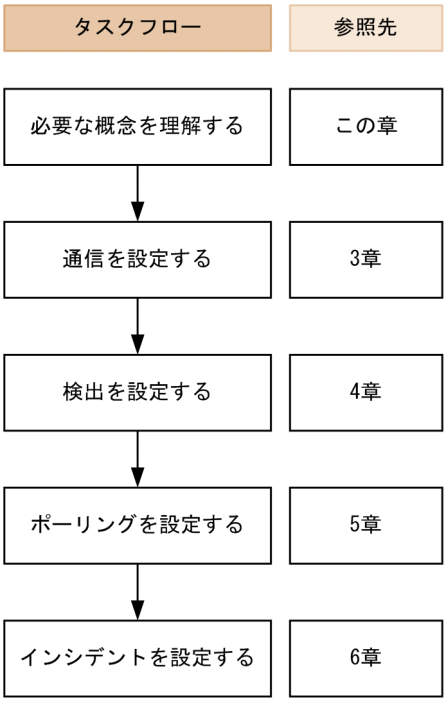
1.2 システム設定（UNIX）

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。

2 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 3 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。

2.1　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**－（任意で実行）設定を調整して，NNMi のパフォーマンスを向上します。

2.2 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。

2.3 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。

2.4 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。

2.5 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。

ポイント

- 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。

2.6 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からグローバルマネージャに渡されたノードのフィルタリング

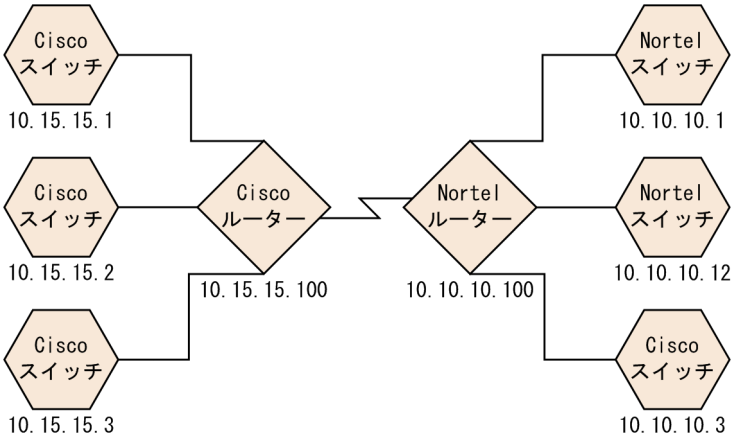
インタフェースグループは，次のどれか，またはすべての目的に使用できます。

- 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。

2.6.1 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。

2.6.2 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。

  > **参考**
  >
  > 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。

> **参考**
>
> 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，「4.1.1 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。

> **参考**
>
> 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。

2.6.3 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 **[ほとんどの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 **[ほとんどの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。

2.6.4 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，ifDescr，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。

インタフェースグループの定義に，集約リンク機能への依存関係が含まれていて，その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は，NNMi に制限事項があります。この場合，それらのインタフェースが常に除外されるとは限りません。

2.7　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. **インタフェースグループの設定－NNMi は，最初に一致したインタフェースグループ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**インタフェースグループの設定**定義です。

2. **ノードグループの設定－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，ノードグループ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**ノードグループの設定**定義です。

   参考

   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。**

2.8 設定をやり直す

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. **(任意で実行) 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. **(任意で実行) 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. **(任意で実行) この手順によってデータベースが削除されるため，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するとよい。**

   ```
   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. **NNMi の設定を開始します。次のどれかを行う。**

   - 「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。

注意事項

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。

3 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。

3.1 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。

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

3.1.1 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，アクセス資格認定，タイムアウトと再試行の値，ICMP と SNMP のプロトコル使用可能性，および SNMP アクセス設定を設定できます。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 ICMP 通信および SNMP 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。

3.1.2 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。

3.1.3 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，または HMAC-SHA-1 のどちらか選択した方の，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，暗号化を使用する場合は，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）。

3.1.4 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。

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。

3.1.5 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。

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。

3.1.6 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。

3.1.7 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。

3.2　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

3.2.1　デフォルトの通信設定を計画する

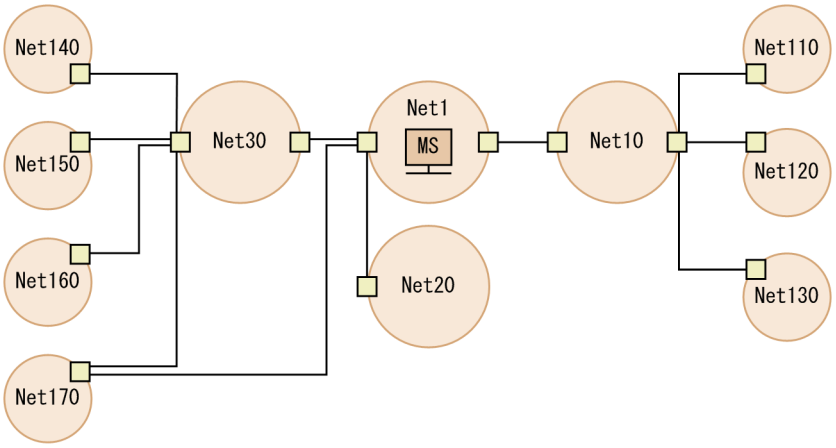
NNMi は，該当する領域や特定のノードで指定しなかった設定をデフォルト値を使用して完成させるため，大半のネットワークで妥当なものになるようデフォルトを設定します。

- 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？

3.2.2　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。

3.2.3　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。

3.2.4　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 の場合：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。
- UNIX の場合：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。

3.2.5　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。

参考

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。

3.2.6　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。

参考

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

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

3.3　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

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

参考

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，「2.2　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。

変更を反映するには，NNMi コンソールに戻るまでに，すべての［通信の設定］で［保存して閉じる］を実施してください。

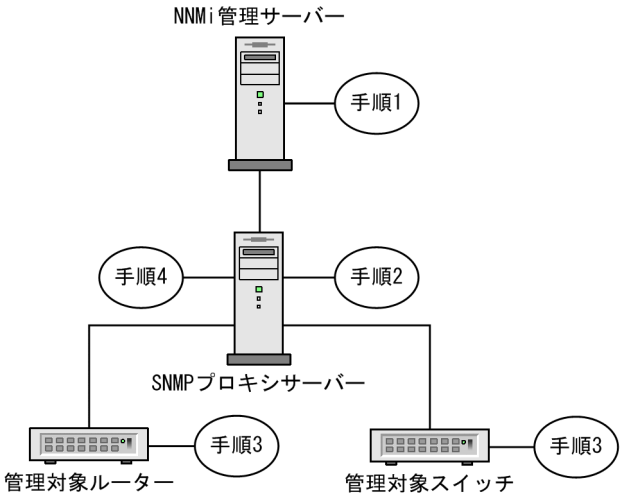
ポイント

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。

3.3.1　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。図 3-1 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。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。

図 3-1　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（.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）。

   プロキシサーバーを使用するように設定されている場合，NNMi は次の OID などを使用して SNMP 応答を処理します。

   - SecurityPackAgentAddressOid .1.3.6.1.4.1.99.12.45.1.1 (SNMP Research NetDiscover SECURITY-PACK-MIB)
   - SecurityPackNotificationAddressOid .1.3.6.1.4.1.99.12.45.2.1 (SNMP Research NetDiscover SECURITY-PACK-MIB)
   - ProxyOid .1.3.6.1.4.1.11.2.17.5.1.0 (HP)
   - 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。

3.3.2 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。

その名称からもわかるように，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，機械解析が容易であり，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。したがって，SNMP と比べると使用できるデバイスのベンダーは限られています。また，NETCONF コマンドは，一般にベンダー独自仕様の部分が多く，SNMP の多くの標準 MIB やベンダー独自の MIB ほど広く公開されていません。したがって，NNMi で NETCONF を活用できる範囲は限られています。しかし，特定のベンダーがデバイスに NETCONF を実装しており，NNMi が必要とする管理情報を報告する場合は，そのデバイス固有の NETCONF のサポートを NNMi に追加することもできます。

(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。

認証情報を設定すると，NNMi は次の検出時から，指定したデバイス（ノード）の新しい認証情報を使用するようになります。

各管理対象デバイスには，1 組の SSH ユーザーおよびパスワードしか設定できないので，そのデバイスに対する通常の SSH セッションと NETCONF セッションで同一の認証情報のセットが使用されます。NNMi の［通信設定］フォームを編集する方法の詳細については，NNMi のヘルプを参照してください。

3.4　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？
  「3.4.1　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？
  「3.4.2　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？
  「3.4.3　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？
  「3.4.4　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？
  「3.4.5　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。

3.4.1　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）。

3.4.2　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。

3.4.3　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］＞［通信の設定］を選択する。
3. ［通信の設定］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。

3.4.4　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］＞［通信の設定］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. ［通信の設定］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。

3.4.5　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。

3.5 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。

4 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。

NNM の使用経験があり NNMi で検出がどのように変わったのかを知りたい方は，「23.1　ネットワーク検出」を参照してこの両者の違いについての高度な説明をお読みください。

4.1　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。

注意事項

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。

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，「5.　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。

注意事項

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。

4.1.1 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。

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。

4.2　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

4.2.1　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。

注意事項

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。

4.2.2 自動検出ルールを計画する（ルールベース検出だけ）

(1) 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。

(2) 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）。

> **参考**
>
> 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。

(3) 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。

> **参考**
>
> - 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。

ポイント

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。

(4) 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 トラップからの検出ヒントは，WAN 内でデバイスを検出する場合に特に有効です。

(5) 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。

(6) 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。

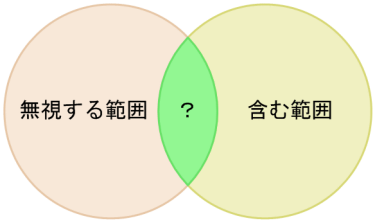
  －自動検出ルールに大きな IP アドレス範囲を追加したあとに，そのルール内の検出から幾つかの IP アドレスを除外できます。
- システムオブジェクト ID 範囲の指定は接頭部分であり，絶対値ではありません。例えば，範囲 1.3.6.1.4.1.11 は 1.3.6.1.4.1.11.*と同じです。

(7) 例

検出ルールの重複

図 4-1 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 4-1　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。

4.2.3　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

> **注意事項**
>
> 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。

> **ポイント**
>
> - 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。

4.2.4　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。

ポイント

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。

4.2.5　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。

ポイント

- 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。

ポイント

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。

4.2.6 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。

4.2.7　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，「3.1.6　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。

  除外 IP アドレス機能は，新規および既存のノードに対して有効です。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。

4.2.8　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。

4.3　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。

注意事項

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，［保存して閉じる］を実行する。

参考

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，「2.2　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。

4.3.1　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。

4.3.2　シードを設定する場合のヒント

- 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）。

4.4 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。

4.4.1 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。

4.4.2 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。
- 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，「3.4　通信の評価」を参照してください。

4.4.3　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。

4.4.4　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「3.4.1　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。

4.4.5　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，「4.3.1　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。

4.4.6 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。

4.4.7 デバイスを再検出する

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。

4.5　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，「19.19　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。

4.5.1　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。

5 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。

バージョン 8 以前の NNM をお使いの方で，NNMi で監視がどのように変更されたか知りたい場合は，相違点の高レベルの概要に関する「23.2　ステータス監視」を参照してください。

5.1　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「5.2　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。

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「2.6 ノードグループおよびインタフェースグループ」と「2.7　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。

5.1.1　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。**

5.2　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。

5.2.1　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

(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。ネットワークに複数の HP ProCurve 2810-48G 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，スイッチの 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スイッチがあることを確認します。検出の設定の詳細は，「4.　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？
   複数のスイッチを 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，スイッチ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，スイッチについて 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。スイッチの例として，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。

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。

- 「5.2.2　ステータスポーリングの監視対象を計画する」
- 「5.2.3　ノードグループとインタフェースグループを作成する」
- 「5.2.4　ポーリング間隔を計画する」
- 「5.2.5　収集するデータを計画する」

5.2.2　ステータスポーリングの監視対象を計画する

デフォルトで，NNMi State Poller は SNMP ポーリングを使って次を監視します。

- NNMi 検出対象デバイス上で既知の別のインタフェースに接続されたインタフェース。
- IP アドレスをホストするルーターインタフェース。

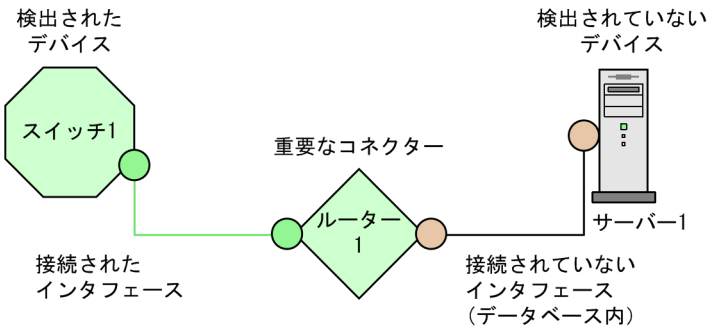
> **参考**
>
> 多くの場合，接続されたインタフェースへのポーリングだけでも，十分に正確な根本原因分析ができます。監視対象インタフェースのセットを拡張すると，ポーリングのパフォーマンスに影響が及ぶ可能性があります。

監視の拡張

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。

> **参考**
>
> 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。

(1) 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。

  参考

  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。

(2) 監視の停止

NNMi 管理モードを使って，デバイスまたはインタフェースを非管理対象またはサービス停止中に設定できます。非管理対象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必要はありません。サービス停止中は，1 つ以上のオブジェクトがオフラインになり，ダウン（停止の）インシデントが不要な一時的な状況と見なされます。

すべてのグループ設定を通して，管理モードを考えてください。グループ，ポーリング間隔，種類に無関係に，オブジェクトのステータスが非管理対象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。

ポイント

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。非管理対象に恒久的に設定する可能性のあるそれらオブジェクトに着目してください。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。

5.2.3 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。

ポイント

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 **[ノードグループの設定]** タブと **[インタフェースグループの設定]** 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？

> **ポイント**
>
> 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（「5.1 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。

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。

ポイント

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。

5.2.4　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。

ポイント

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。

参考

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。

5.2.5　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

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。

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。

ポイント

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。

5.3　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「*モニタリングの動作を設定する*」を参照してください。

> **参考**
>
> 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，「2.2　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。

5.3.1　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「5.2.3 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。

参考

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。

5.3.2　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。

- 障害ポー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。

ポイント

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。

5.3.3　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［ノードグループの設定］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。

- 障害ポー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。

ポイント

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。

5.3.4　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。

参考

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。

5.4　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。

5.4.1　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。

(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 ポーリングが有効になっています］と［障害のポーリング周期］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。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。

5.4.2 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。

(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 5-1 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
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
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。

情報	説明
	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。
状態アップデートキュー期間	• 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。

5.5 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）**［ノードグループの設定］** フォームまたは **［インタフェースグループの設定］** 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。

6 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。

バージョン 8 以前の NNM で作業した経験があり，イベント監視がどのように変更されたかを知りたい場合は，「23.3　イベント監視のカスタマイズ」を参照してください。

6.1 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，次を参照してください。

- NNMi ヘルプの「*イベントパイプラインについて*」
- NNMi ヘルプの「*NNMi の Causal Engine とインシデント*」

6.1.1 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 6-1　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
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，「6.1.7　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）。	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

6.1.2　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 6-2　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 • NNM 管理ステーションからのバージョン 8 以前の NNM イベント	• ネットワークデバイスからの SNMP トラップ • NNMi 管理イベント	• ネットワークデバイスからの SNMP トラップ • NNM 管理ステーションからのバージョン 8 以前の NNM イベント
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 **[リモート NNM 6.x/7.x のイベント設定]** フォームの **[グローバルマネージャへの転送]** タブ
注	－	－	グローバルマネージャ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)*」 • NNMi ヘルプの「*リモート 6.x/7.x イベント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。

6.1.3 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 SNMP トラップ転送メカニズムの設定方法の詳細については，NNMi ヘルプの「トラップ転送設定」を参照してください。
- 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 UNIX（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 UNIX 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 varbind*」，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`

  UNIX：`$NNM_SNMP_MIBS/Vendor/Hewlett-Packard/hp-nnmi.mib`

- 元のトラップ転送が設定された UNIX

  UNIX 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，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。

6.1.4 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ノードコンポーネント（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。

6.1.5 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 6-3 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。

名前	説明
	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 6-3 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 6-3 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

6.1.6 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。

6.1.7 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）。

6.1.8 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。

6.2 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

6.2.1 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。

6.2.2 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。

6.2.3 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。

6.3　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。

> **参考**
>
> 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，「2.2　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　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。

6.3.2　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
  ```

  UNIX

  ```
  $NNM_DATA/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。

  > **注意事項**
  >
  > 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。

6.3.3 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
- UNIX：$NNM_DATA/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。

6.3.4 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。

6.3.5　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 6-4　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.rmiPort	1,097
com.hp.ov.nms.trapd.trapInterface	すべてのインタフェース
com.hp.ov.nms.trapd.recvSocketBufSize	2,048 キロバイト
com.hp.ov.nms.trapd.pipeline.qSize	50,000 トラップ
com.hp.ov.nms.trapd.connectToWinSNMP	false
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.analysis.numSources	10 ソース
com.hp.ov.nms.trapd.analysis.windowSize	300 秒（5 分）
com.hp.nms.trapd.updateSourcesPeriod	30 秒
com.hp.nms.trapd.notifySourcesPeriod	300 秒
com.hp.ov.nms.trapd.hosted.object.trapstorm.enabled	false
com.hp.ov.nms.trapd.hosted.object.trapstorm.threshold	10 トラップ/秒
com.hp.ov.nms.trapd.database.fileSize	100 メガバイト

トラップサーバープロパティ	デフォルト値
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。

6.4　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。

6.4.1　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
- UNIX：/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。

6.4.2　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
- UNIX：/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，名前またはその他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。

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。

6.5　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。

  > **参考**
  >
  > 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。

6.6 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。

6.6.1 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
   - UNIX：$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. 任意で，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. 任意で，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. ファイルを保存してから，次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

6. 「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。

6.6.2 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。

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
- UNIX：$NNM_PROPS/nms-jboss.properties

2. 次の行を検索し，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. ファイルを保存してから，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
   - UNIX：$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. ファイルを保存してから，次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

7 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。

7.1 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。

7.1.1　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。

> **参考**
>
> 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。
8. [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する。
9. [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。
10. [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。
11. [追加] をクリックする。
12. [属性] フィールドで，[hostname] を選択する。
13. [演算子] フィールドで，[like] を選択する。
14. [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。
15. [追加] をクリックする。
16. [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。

> **参考**
>
> 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。

> **注意事項**
>
> **[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。「7.1.1(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。

7.1.2 ノードグループマップを設定する

(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. ［レイアウトの保存］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。

> **参考**
>
> 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［レイアウトの保存］を使用してください。［レイアウトの保存］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作成できたので，今度はマップを表示して内容を確認します。

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。

> **参考**
>
> ［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，[ほとんどの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。

> **参考**
>
> 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［トポロジマップワークスペース内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。

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/nnmbg/］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. ［world.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。

> **参考**
>
> コピーしたリンクを［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。

7.1.3 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。

7.2　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
   - UNIX：$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
```

   注意事項

   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
```

7.3 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「*デフォルトのマップ設定を定義する*」を参照してください。

7.4　アナリシス（分析）ペインに表示されるゲージの最大数を設定する

アナリシス（分析）ペインのゲージタブは，ステートポーラーとカスタムポーラーのリアルタイムデータを表示します。これらのゲージはノード，インタフェース，カスタムノード収集，またはカスタムポーリングインスタンスのデータと CPU，メモリなどのコンポーネントヘルスのデータを表示します。

アナリシス（分析）ペインに表示されるゲージの最大数の設定は，次の手順を実行します。

1. 次のファイルを編集する。

   Windows の場合：%NNM_PROPS%¥nms-ui.properties

   UNIX の場合：$NNM_PROPS/nms-ui.properties

2. 次の行を探す。

```
#!com.hp.nnm.ui.maxGaugePerAnalysisPanel = 24
```

   表示されるゲージの最大値を次のように指定します。

```
com.hp.nnm.ui.maxGaugePerAnalysisPanel = 12
```

> **注意事項**
>
> 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
```

> **参考**
>
> 表示するゲージの数を多くすると，アナリシス（分析）ペインを開くときのパフォーマンスが低下します。表示するゲージの数を少なくすると，ゲージのサイズが大きくなります。

7.5　アナリシス（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，次の手順を実行します。

1. 次のファイルを編集する。
   - Windows：%NNM_PROPS%¥nms-ui.properties
   - UNIX：$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
   ```

7.6　アナリシス（分析）ペインに表示されるゲージの更新間隔を設定する

アナリシス（分析）ペインに表示されるゲージの更新間隔を秒単位で指定するには，次の手順を実行します。

1. 次のファイルを編集する。

   Windows の場合：%NNM_PROPS%¥nms-ui.properties

   UNIX の場合：$NNM_PROPS/nms-ui.properties

2. 次の行を探す。

```
#!com.hp.nnm.ui.analysisGaugeRefreshSecs = 15
```

   次のように更新間隔を秒で指定します。

```
com.hp.nnm.ui.analysisGaugeRefreshSecs = 10
```

   注意事項

   - 行の先頭の「#!」を忘れずに削除してください。
   - 「0」を設定すると，ゲージは更新されません。また，更新間隔を短く（例えば 10 秒以下）にすると，SNMP エージェントが応答する MIB 値をキャッシュして，同じ値を返してくることがあります。

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
```

7.7 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。

7.8 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. 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 パラメータをメモします。
   （例）

```
viewInfoId=allIncidentsTableView
```

2. 次のファイルを編集する。
   - Windows：%NMS-PROPS%¥nms-ui.properties
   - UNIX：$NNM_PROPS/nms-ui.properties

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
```

注意事項

- VIEWKEYWORD は，ビューの URL 内の viewInfoId パラメータです。
- 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。

8 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.keystore 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.truststore 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.keystore ファイルとnnm.truststore 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.keystore およびnnm.truststore 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。

8.1　証明書を設定する

次の情報に従い，特別な要件に応じて証明書を設定します。

- CA 証明書を使用する場合は，「8.2　認証機関証明書を生成する」の指示に従ってください。
- グローバル，リージョナル，またはその両方の NNMi 管理サーバーでアプリケーションフェイルオーバー機能を使用するように設定した場合は，追加の設定手順があります。グローバルネットワーク管理設定を完了する前に，「8.3　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」の説明にある手順を実行して，それぞれの NNMi 管理サーバーの`nnm.keystore` ファイル，および `nnm.truststore` ファイルをマージします。
- 認証機関を使用する必要があり，グローバル，リージョナル，またはその両方の NNMi 管理サーバーでアプリケーションフェイルオーバー機能を使用するように設定した場合は，追加の設定手順があります。まず，「8.2　認証機関証明書を生成する」の説明にある手順を実行し，次に，グローバルネットワーク管理設定を完了する前に，「8.4　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」の説明にある手順を実行して，それぞれの NNMi 管理サーバーの`nnm.keystore` ファイル，および `nnm.truststore` ファイルをマージします。
- グローバル，リージョナル，またはその両方の NNMi 管理サーバーで高可用性（HA）を使用するように設定した場合は，グローバルネットワーク管理設定を完了する前に，「8.5　自己署名証明書または CA 証明書を使用するように高可用性クラスタを設定する」の説明にある手順を実行して，`nnm.keystore` および`nnm.truststore` ファイルで自己署名証明書を作成します。
- 各 HA またはアプリケーションフェイルオーバークラスタを正しく設定した後，アクティブなリージョナルノードからアクティブなグローバルノードに`nnm.truststore` ファイルをコピーして，トラストストアをマージすることで，グローバルネットワーク管理機能を有効にします。この操作は，アクティブなリージョナルノードごとに実行する必要があります。NNMi 管理サーバーが「8.2　認証機関証明書を生成する」の手順で生成した CA 証明書を使用する場合，グローバルトラストストアにマージする必要があるのはこれらの CA 証明書だけです。
- グローバルネットワーク管理設定で NNMi 管理サーバーを設定し，その後でリージョナル，グローバル，またはその両方をアプリケーションフェイルオーバークラスタに含めることにした場合は，「8.3 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」の指示に従ってください。そのセクションに示されているコマンドを使用して`nnm.keystore` および`nnm.truststore` ファイルを正しく設定し，変更された`nnm.truststore` ファイルをグローバル NNMi 管理サーバーにコピーし，そのファイルをグローバル NNMi 管理サーバーの`nnm.truststore` ファイルにマージする必要があります。
- グローバルネットワーク管理設定で NNMi 管理サーバーを設定し，その後でリージョナル，グローバル，またはその両方で HA を使用することにした場合は，「8.5　自己署名証明書または CA 証明書を使用するように高可用性クラスタを設定する」の指示に従ってください。
- 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，「8.8　ディレクトリサービスへの SSL 接続を設定する」の指示に従ってください。

8.2 認証機関証明書を生成する

CA（認証機関）を使用する場合は，次の手順で CA 証明書を生成します。

> **参考**
>
> NNMi で CA を使用する場合は，RSA アルゴリズムを使用して証明書に署名します。DSA アルゴリズムはサポートされていません。

1. nnm.keystore およびnnm.truststore ファイルが存在する NNMi 管理サーバーのディレクトリに移動する。
   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/shared/nnm/certificates
2. nnm.keystore ファイルのバックアップコピーを保存する。
3. システムからプライベートキーを生成する。このプライベートキーを生成するには，keytool コマンドを使用する。

   a　次のコマンドを実行します。

   Windows

   ```
   %NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe ¥
   -genkeypair -validity 36500 -keyalg rsa -keystore ¥
   nnm.keystore -storepass nnmkeypass ¥
   -keypass nnmkeypass -keysize 2048 -alias ¥
   myserver.mydomain
   ```

   UNIX

   ```
   $NnmInstallDir/nonOV/jdk/nnm/bin/keytool ¥
   -genkeypair -validity 36500 -keyalg rsa -keystore ¥
   nnm.keystore -storepass nnmkeypass ¥
   -keypass nnmkeypass -keysize 2048 -alias ¥
   myserver.mydomain
   ```

   （凡例）

   行の最後の¥は，行が続いていることを示します。

参考

- 別名（この例では*myserver.mydomain*）は，この新規作成キーを識別する名前です。別名は任意の文字列にできますが，*myserver.mydomain* 別名の変数として，ご使用のシステムの FQDN（完全修飾ドメイン名）を使用するようお勧めします。
- Linux オペレーティングシステムには，この手順で使用されるkeytool コマンドまたはコマンドオプションと互換性のないkeytool コマンドがあります。

b　必要な情報を入力します。

注意事項

姓名の入力を求められたら，システムの FQDN（完全修飾ドメイン名）を入力してください。

4. 次のコマンドを実行して CSR（証明書署名要求）ファイルを作成する。

Windows

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe ¥
-keystore nnm.keystore -certreq -storepass nnmkeypass ¥
-alias myserver.mydomain -file CERTREQFILE
```

UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -keystore ¥
nnm.keystore -certreq -storepass nnmkeypass -alias ¥
myserver.mydomain -file CERTREQFILE
```

（凡例）

行の最後の¥は，行が続いていることを示します。

参考

keytool コマンドの詳細については，http://www.oracle.com/technetwork/java/index.html で「鍵と証明書の管理ツール」を検索してください。

5. CA 署名機関に CSR を送信する。

次のどちらかが発行されます。

- myserver.crt という名前の署名付き証明書

  myserver.crt ファイルには，サーバー証明書（ファイルに含まれている最上位の証明書）と，1 つ以上の CA（認証機関）証明書の両方が含まれています。CA 証明書を新しいファイルであるmyca.crt ファイルにコピーします。サーバー証明書をnnm.keystore 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.truststore 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。

- myserver.crt とCA.crt という名前の 2 つのファイル

  CA.crt ファイルの内容をmyserver.crt ファイルの最後に追加します。サーバー証明書をnnm.keystore 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.truststore 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。

次に，CA 署名機関から受け取るファイルの例を示します。

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/AVQQKExNQU0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLEwdOZXR3b3Js
eGVSZXZvY2F0aW9uTGlzdD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlw
.....
.....
TZImiZPyLGQBGRYDaW50MRIwEAYKCZImiZPyLGQBGRYCc2cxEzARBgNVBAMTCmNb
pSo6o/76yShtT7Vrlfz+mXjWyEHaIy/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/VQQKExNQU0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLEwdOZXR3b3Js
eGVSZXZvY2F0aW9uTGlzdD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJ1dGlw
.....
.....
TZImiZPyLGQBGRYDaW50MRIwEAYKCZImiZPyLGQBGRYCc2cxEzARBgNVBAMTCmNbp
So6o/76yShtT7Vrlfz+mXjWyEHaIy/QLCpPebYhejHEg4dZgzWWT/lQt==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Sample2/Gh0dHA6Ly9jb3JwMWRjc2cyLnNnLmludC5wc2FnbG9iYWwuY29tL0Nlc
RaOCApwwggKYMB0GA1UdDgQWBBSqaWZzCRcpvJWOFPZ/Be9b+QSPyDAfBgNVHSMC
.....
.....
Wp5Lz1ZJAOu1VHbPVdQnXnlBkx7V65niLoaT90Eqd6laliVlJHj7GBriJ90uvVGu
BQagggEChoG9bGRhcDovLy9DTj1jb3JwMWRjc2cyL==
-----END CERTIFICATE-----
```

6. これらの証明書が記録されているファイルを NNMi 管理サーバーにコピーする。この例では，次の場所にファイルをコピーする。

   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/shared/nnm/certificates

前の手順で生成した証明書を使用して，自己署名証明書を置き換えます。

1. nnm.keystore およびnnm.truststore ファイルが存在する NNMi 管理サーバーのディレクトリに移動する。
   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/shared/nnm/certificates
2. 次のコマンドを実行して，サーバー証明書および CA 証明書を NNMi のnnm.keystore ファイルにインポートする。

   Windows

   ```
   %NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -importcert ¥
   -trustcacerts -keystore nnm.keystore -storepass nnmkeypass ¥
   -alias myserver.mydomain -file myserver.crt
   ```

   UNIX

   ```
   $NnmInstallDir/nonOV/jdk/nnm/bin/keytool -importcert ¥
   -trustcacerts -keystore nnm.keystore -storepass nnmkeypass ¥
   -alias myserver.mydomain -file myserver.crt
   ```

   （凡例）

   行の最後の¥は，行が続いていることを示します。

   > **参考**
   >
   > -storepass オプションを使用し，パスワードを入力する場合，キーストアプログラムはキーストアパスワードの入力を要求しません。-storepass オプションを使用しない場合は，キーストアパスワードの入力を求められたときにnnmkeypass と入力してください。

3. 証明書の信頼を確認するメッセージが表示されたら，y と入力する。

   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
   ```

4. 次のコマンドを実行して，CA 証明書を NNMi のnnm.truststore ファイルにインポートする。

Windows

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -import ¥
-alias myca -keystore nnm.truststore -file myca.crt
```

UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -import -alias myca ¥
-keystore nnm.truststore -file myca.crt
```

(凡例)

行の最後の¥は，行が続いていることを示します。

5. トラストストアのパスワードの入力を求められたら，ovpass と入力する。

6. トラストストアの内容を確認する。

Windows

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -list ¥
-keystore nnm.truststore
```

UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -list ¥
-keystore nnm.truststore
```

(凡例)

行の最後の¥は，行が続いていることを示します。

トラストストアのパスワードの入力を求められたら，ovpass と入力します。

トラストストアの出力例

トラストストアの出力形式は次のとおりです。トラストストアには複数の証明書を含めることができます。

```
Keystore type: jks
Keystore provider: SUN
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
```

7. myca.crt ファイルに次の 2 つの証明書エントリがあるとする。

```
-----BEGIN CERTIFICATE-----
IntermediateCert/lots of content
            :
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
RootCAcert/lots of content
            :
-----END CERTIFICATE-----
```

手順 4.から手順 6.までで，最初の証明書がnnm.truststore ファイルにインポートされました。ほかの証明書をインポートするには，一度に 1 つずつ，nnm.truststore ファイルにインポートする必要があります。

例えば，この例の 2 つ目以降の証明書をインポートするには，次のようにします。

a

2 つ目の証明書エントリをmyca.crt から新しいファイルrootCa.crt にコピーします。

トラストストアは複数の証明書を含むことができます。

手順 4.から手順 6.までで，最初の証明書がmyca.crt ファイルからインポートされます。myca.crt ファイルに複数の証明書があり，複数のBEGIN CERTIFICATE とEND CERTIFICATE のブロックによって示されている場合，それらの証明書もnnm.truststore ファイルにインポートする必要があります。

b

2 つ目の証明書を個別にnnm.truststore ファイルにインポートします。

- Windows

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -import -alias ¥
myrootca -keystore nnm.truststore -file rootCA.crt
```

- UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -import -alias ¥
myrootca -keystore nnm.truststore -file rootCA.crt
```

（凡例）

行の最後の¥は，行が続いていることを示します。

c

nnm.truststore ファイルにインポートする必要がある追加の証明書のそれぞれについて，手順 a から手順 b までを繰り返します。

8. 次のファイルを編集する。
   - Windows：%NNM_CONF%¥nnm¥props¥nms-local.properties
   - UNIX：$NNM_CONF/nnm/props/nms-local.properties
9. com.hp.ov.nms.ssl.KEY_ALIAS 変数を，*myserver*.*mydomain* で使用した値に更新する。

   忘れずに設定内容を保存してください。
10. 次のコマンドを実行して NNMi を再起動する。

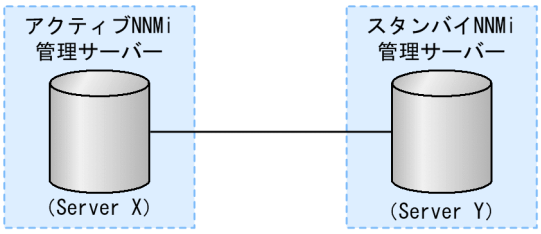
```
ovstop
ovstart
```

11. 構文 https://<fully_qualified_domain_name>:<port_number>/nnm/を使用して，NNMi コンソールへの HTTPS アクセスをテストする。

    ブラウザによって 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。

8.3　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

図 8-1　アプリケーションフェイルオーバーでの自己署名証明書の使用法



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.keystore ファイルおよびnnm.truststore ファイルの内容をマージして，それぞれ 1 つのnnm.keystore ファイルおよびnnm.truststore ファイルにする必要があります。次の手順を実行し，上の図に基づいて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ように設定します。

NNMi でアプリケーションフェイルオーバー機能とともに自己署名証明書を使用する場合，次の手順を完了しなければ，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. 手順 2.を完了する前に，Server Y で次のディレクトリに移動する。
   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/shared/nnm/certificates
2. nnm.keystore およびnnm.truststore ファイルを，Server Y から Server X の一時保存場所にコピーする。
   残りの手順では，これらのファイル保存場所は，*<keystore>*および*<truststore>*を指します。
3. Server X で次のコマンドを実行し，Server Y の証明書を Server X のnnm.keystore およびnnm.truststore ファイルにマージする。

   Windows および UNIX

   ```
   nnmcertmerge.ovpl -keystore <keystore> -truststore <truststore>
   ```
4. マージしたnnm.keystore およびnnm.truststore 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する。
   これらのファイル保存場所は，次のとおりです。
   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/shared/nnm/certificates
5. Server X と Server Y の両方で次のコマンドを実行する。
   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

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -list ¥
-keystore %NnmDataDir%¥shared¥nnm¥certificates¥nnm.keystore ¥
-storepass nnmkeypass
```

UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -list -keystore ¥
$NnmDataDir/shared/nnm/certificates/nnm.keystore -storepass ¥
nnmkeypass
```

（凡例）

行の最後の¥は，行が続いていることを示します。

6. Server X と Server Y の両方で次のコマンドを実行する。

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，手順 1.から手順 6.までをやり直します。

Windows

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -list ¥
-keystore %NnmDataDir%¥shared¥nnm¥certificates¥nnm.truststore ¥
-storepass ovpass
```

UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -list -keystore ¥
$NnmDataDir/shared/nnm/certificates/nnm.truststore -storepass ¥
ovpass
```

（凡例）

行の最後の¥は，行が続いていることを示します。

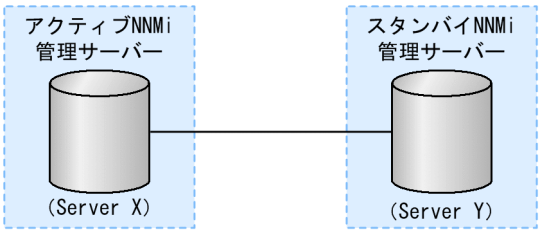
7. 「16.3　アプリケーションフェイルオーバー構成の NNMi を設定する」の手順 6.からアプリケーションフェイルオーバー機能の設定を続行する。

参考

手順 4.は手動で実行しましたが，アプリケーションフェイルオーバー機能を実行すると，NNMi は，マージされたキーストアとトラストストアの情報を Server X から Server Y へ自動的に複製します。

8.4　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

図 8-2　アプリケーションフェイルオーバーでの CA 証明書の使用法



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.keystore ファイルおよびnnm.truststore ファイルの内容をマージして，それぞれ 1 つのnnm.keystore ファイルおよびnnm.truststore ファイルにする必要があります。次の手順に従い，上記図に基づき CA 証明書を使用するアプリケーションフェイルオーバー機能を設定します。

NNMi でアプリケーションフェイルオーバー機能とともに CA 証明書を使用する場合，次の手順を完了しなければ，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。

Server Y については，「8.2　認証機関証明書を生成する」の手順に従います。

1. 手順 2.を完了する前に，Server Y で次のディレクトリに移動する。
   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/shared/nnm/certificates
2. nnm.keystore およびnnm.truststore ファイルを Server Y から Server X の一時ファイル保存場所にコピーする。
   残りの手順では，これらのファイル保存場所は〈*keystore*〉および〈*truststore*〉と呼びます。
3. Server X で次のコマンドを実行し，Server Y の証明書を Server X のnnm.keystore およびnnm.truststore ファイルにマージする。
   Windows および UNIX
   ```
   nnmcertmerge.ovpl -keystore 〈keystore〉 -truststore 〈truststore〉
   ```
4. マージしたnnm.keystore およびnnm.truststore ファイルを server X から server Y にコピーし，どちらのノードにもマージ済みファイルがあるようにする。
   これらのファイル保存場所は，次のとおりです。
   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/shared/nnm/certificates
5. Server X と Server Y の両方で次のコマンドを実行する。

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，手順 1.から手順 5.までをやり直します。

Windows

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -list ¥
-keystore %NnmDataDir%¥shared¥nnm¥certificates¥nnm.keystore ¥
-storepass nnmkeypass
```

UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -list -keystore ¥
$NnmDataDir/shared/nnm/certificates/nnm.keystore -storepass ¥
nnmkeypass
```

（凡例）

行の最後の¥は，行が続いていることを示します。

6. Server X と Server Y の両方で次のコマンドを実行する。

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，手順 1.から手順 6.までをやり直します。

Windows

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -list ¥
-keystore %NnmDataDir%¥shared¥nnm¥certificates¥nnm.truststore ¥
-storepass ovpass
```

UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -list -keystore ¥
$NnmDataDir/shared/nnm/certificates/nnm.truststore -storepass ¥
ovpass
```

（凡例）

行の最後の¥は，行が続いていることを示します。

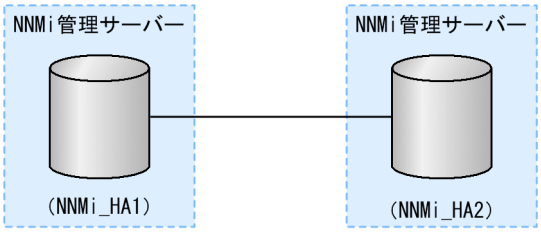
7.「16.3　アプリケーションフェイルオーバー構成の NNMi を設定する」の手順 6.からアプリケーションフェイルオーバー機能の設定を続行する。

参考

手順 4.は手動で実行しましたが，アプリケーションフェイルオーバー機能を実行すると，NNMi は，マージされたキーストアとトラストストアの情報を Server X から Server Y へ自動的に複製します。

8.5　自己署名証明書または CA 証明書を使用するように高可用性クラスタを設定する

図 8-3　HA での証明書の使用法



上記の図に基づき，自己署名証明書または CA 証明書を使用する高可用性クラスタを設定する手順について説明します。

8.5.1　自己署名証明書を使用するように高可用性クラスタを設定する

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。

8.5.2　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

新規の自己署名証明書または CA 証明書を作成し，newcert と呼ぶとします。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。

この手順は，NNMi に HA を設定する前または後に実行できます。HA の設定については，「17.4　HA を設定する」を参照してください。

1. 手順 2.を完了する前に，NNMi_HA1 で次のディレクトリに移動する。
   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/shared/nnm/certificates
2. NNMi_HA1 で，次のコマンドを実行して，newcert をnnm.keystore ファイルにインポートする。

   Windows

   ```
   %NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -import ¥
   -alias newcert_Alias -keystore nnm.keystore -file newcert
   ```

   UNIX

   ```
   $NnmInstallDir/nonOV/jdk/nnm/bin/keytool -import ¥
   -alias newcert_Alias -keystore nnm.keystore -file newcert
   ```

（凡例）

行の最後の¥は，行が続いていることを示します。

3. アクティブなクラスタノード（NNMi_HA1）とスタンバイノード（NNMi_HA2）の両方で次のファイルを編集する。
   - Windows：%NNM_DATA%¥conf¥nnm¥props¥nms-local.properties
   - UNIX：$NNM_DATA/conf/nnm/props/nms-local.properties
4. NNMi_HA1 と NNMi_HA2 の両方のnms-local.properties ファイルのcom.hp.ov.nms.ssl.KEY_ALIAS 変数を次のように更新する。

```
com.hp.ov.nms.ssl.KEY_ALIAS = newcert_Alias
```

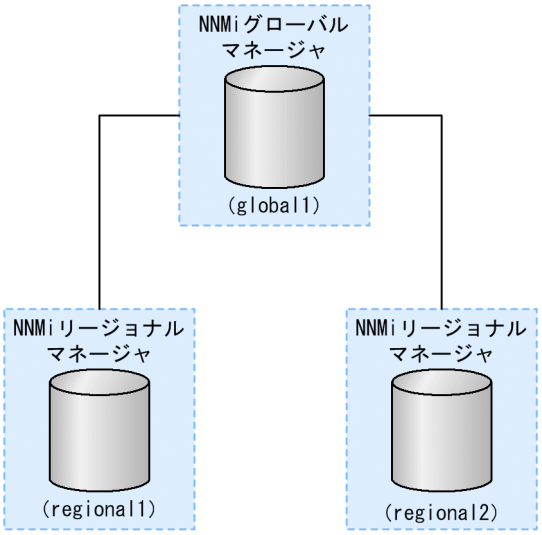
5. 変更を保存する。

8.6 自己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する

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別名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.keystore` ファイル，および `nnm.truststore` ファイルに追加します。

グローバルネットワーク管理設定で，次の図に示すモデルを実現するとします。

図 8-4　グローバルネットワーク管理



次の手順を実行し，上の図に基づいて自己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。

1. 手順 2.を完了する前に，regional1 および regional2 で次のディレクトリに移動する。
   - Windows：`%NNM_DATA%¥shared¥nnm¥certificates`
   - UNIX：`$NNM_DATA/shared/nnm/certificates`
2. `nnm.truststore` ファイルを，上の 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する。
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.truststore` ファイルにマージする。

   `nnmcertmerge.ovpl -truststore` *`regional1_nnm.truststore_location`*

   `nnmcertmerge.ovpl -truststore` *`regional2_nnm.truststore_location`*
4. global1 で，次のコマンドを次の順序で実行する。

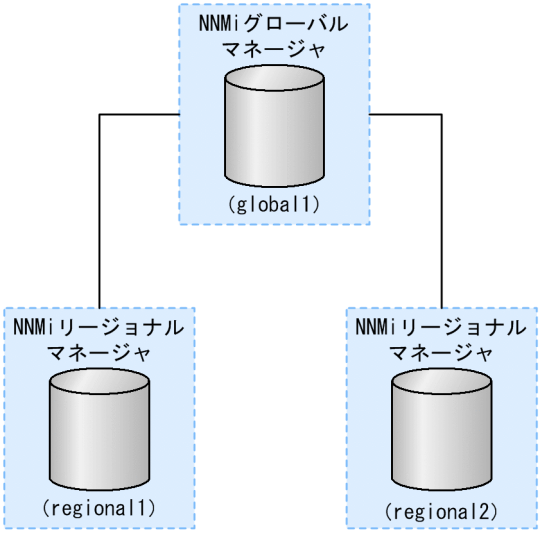
```
ovstop
ovstart
```

8.7　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

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別名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.keystore およびnnm.truststore ファイルに追加します。

グローバルネットワーク管理設定で，次の図に示すモデルを実現するとします。

図 8-5　グローバルネットワーク管理での証明書の使用法



1. regional1 および regional2 については，「8.2　認証機関証明書を生成する」の手順に従う。
2. regional1 および regional2 の次のディレクトリに移動してから，手順 3.を実行する。
   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/shared/nnm/certificates
3. nnm.truststore ファイルを，上の 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する。
4. 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.truststore ファイルにマージする。

   nnmcertmerge.ovpl -truststore *regional1_nnm.truststore_location*

   nnmcertmerge.ovpl -truststore *regional2_nnm.truststore_location*
5. global1 で，次のコマンドを次の順序で実行する。

```
ovstop
ovstart
```

8.8 ディレクトリサービスへの SSL 接続を設定する

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 プロトコルを有効にするときは ldap.properties ファイルに java.naming.security.protocol=ssl パラメータを設定してください。

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証明書を取得する。
   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。

2. NNMi トラストストアが格納されているディレクトリに移動する。
   - Windows：%NNM_DATA%¥shared¥nnm¥certificates
   - UNIX：$NNM_DATA/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証明書を NNMi トラストストアにインポートする。
   a 次のコマンドを実行します。

   Windows

   ```
   %NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool -import ¥
   -alias nnmi_ldap -keystore nnm.truststore ¥
   -file <Directory_Server_Certificate.txt>
   ```

   UNIX

   ```
   $NnmInstallDir/nonOV/jdk/nnm/bin/keytool -import ¥
   -alias nnmi_ldap -keystore nnm.truststore ¥
   -file <Directory_Server_Certificate.txt>
   ```

   <Directory_Server_Certificate.txt>は，会社のトラストストア証明書です。

   （凡例）

   行の最後の¥は，行が続いていることを示します。

   b トラストストアのパスワードの入力を求められたら，ovpass と入力します。

   c 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
シリアル番号 : 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. トラストストアの内容を確認する。

Windows

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -list ¥
-keystore nnm.truststore
```

UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -list ¥
-keystore nnm.truststore
```

(凡例)

行の最後の¥は，行が続いていることを示します。

トラストストアのパスワードの入力を求められたら，ovpass と入力します。

トラストストアの出力例

トラストストアの出力形式は次のとおりです。

```
Keystore type: jks
Keystore provider: SUN
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
```

トラストストア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 を再起動する。

```
ovstop
ovstart
```

keytool コマンドの詳細については，http://www.oracle.com/technetwork/java/index.html で「鍵と証明書の管理ツール」を検索してください。

9 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，Internet Explorer と Mozilla Firefox のどちらでも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。

9.1 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。

9.2 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。

Internet Explorer のバージョンを確認するには，**［ヘルプ］** > **［バージョン情報］** をクリックします。バージョン情報にテキスト **［64 ビット版］** が含まれない場合，この Internet Explorer は 32 ビットです。

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 9-1　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
Internet Explorer 32 ビット	32 ビット	• 「9.2.1　Windows オペレーティングシステム提供の Telnet クライアント」 • 「9.2.2　サードパーティ Telnet クライアント（標準 Windows）」 • 「9.2.4　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 7	• 「9.2.2　サードパーティ Telnet クライアント（標準 Windows）」 • 「9.2.4　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 7 以外	• 「9.2.3　サードパーティ Telnet クライアント（Windows on Windows）」 • 「9.2.4　サードパーティ SSH クライアント（標準 Windows および Windows on Windows）」
Internet Explorer 64 ビット	64 ビット	• 「9.2.1　Windows オペレーティングシステム提供の Telnet クライアント」 • 「9.2.2　サードパーティ Telnet クライアント（標準 Windows）」 • 「9.2.4　サードパーティ SSH クライアント（標準 Windows および Windows on Windows）」

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	32 ビット	• 「9.2.1　Windows オペレーティングシステム提供の Telnet クライアント」 • 「9.2.2　サードパーティ Telnet クライアント（標準 Windows）」 • 「9.2.4　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 7	• 「9.2.2　サードパーティ Telnet クライアント（標準 Windows）」 • 「9.2.4　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 7 以外	• 「9.2.3　サードパーティ Telnet クライアント（Windows on Windows）」 • 「9.2.4　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，「9.4 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

9.2.1　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 Internet Explorer
- 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 Internet Explorer

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 7，Windows Vista，Windows Server 2008，または Windows Server 2012 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 7 または Windows Vista

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 2008 または Windows Server 2012

a ［サーバーマネージャー］の［機能の概要］で，［機能の追加］をクリックします。

b ［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. (Internet Explorer 専用) Telnet を使用する Internet Explorer を有効化する。

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_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Internet Explorer¥MAIN¥FeatureControl¥FEATURE_DISABLE_TELNET_PROTOCOL］キーに次の値を追加します。

名前	タイプ	データ
iexplore.exe	REG_DWORD	0

3. 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
```

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 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。

9.2.2 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 Internet Explorer
- 64 ビット Windows 7 オペレーティングシステム上の 32 ビット Internet Explorer
- 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 Internet Explorer

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. **(Internet Explorer 専用) Telnet を使用する Internet Explorer を有効化する。**

   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_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Internet Explorer¥MAIN¥FeatureControl¥FEATURE_DISABLE_TELNET_PROTOCOL] キーに次の値を追加します。

名前	タイプ	データ
iexplore.exe	REG_DWORD	0

3. **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。

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`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。

9.2.3 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 Internet Explorer（Windows 7 以外）
- 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. （Internet Explorer 専用）Telnet を使用する Internet Explorer を有効化する。

   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_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥Microsoft¥Internet Explorer¥MAIN¥FeatureControl¥FEATURE_DISABLE_TELNET_PROTOCOL］キーに次の値を追加します。

名前	タイプ	データ
iexplore.exe	REG_DWORD	0

3. 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。

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。

9.2.4 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 Internet Explorer
- 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 Internet Explorer

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 プロトコル
EditFlags	REG_DWORD	2
FriendlyTypeName	REG_SZ	SSH
URL プロトコル	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。

9.3 Linux 上の Firefox に Telnet または SSH を設定する

Linux オペレーティングシステムに Telnet または SSH プロトコルを定義してから，新規プロトコルを使用するように Firefox を設定します。

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。詳細については，http://kb.mozillazine.org/Register_protocol を参照してください。

9.3.1 Linux 上の Firefox に Telnet を設定する

Linux オペレーティングシステムで Telnet プロトコルを使用するように Firefox を設定するには，次の手順に従います。

1. Telnet プロトコルを定義する。

   a /usr/local/bin/nnmtelnet ファイルを次の内容で作成します。

```
#!/bin/bash
#
# Linux shell script called by Firefox in response to
# telnet:// URLs for the NNMi telnet menu.
#
address=`echo $1 | cut -d : -f 2 | sed 's;/;;g'`
port=`echo $1 | cut -d : -f 3`
exec /usr/bin/xterm -e telnet $address $port
```

   b 誰でも実行可能なスクリプト権限を設定します。

```
chmod 755 /usr/local/bin/nnmtelnet
```

2. Telnet 用の Firefox プリファレンスを設定する。

   a Firefox アドレスバーに，about:config と入力します。

   b プリファレンスリスト内を右クリックし，［新規］をクリックしてから，［ブール値］をクリックします。

   c プリファレンス名network.protocol-handler.expose.telnet を入力します。

   d プリファレンス値false を選択します。

3. 新規に定義されたプロトコルを使用するように Firefox を設定する。

   a Telnet リンクを参照します。

   リンクを含む簡易 HTML ファイルを作成，または NNMi コンソールで［アクション］＞［ノードアクセス］＞［Telnet... (クライアントから)］を使用できます。アドレスバーに直接リンクを入力しても，同じ結果にはなりません。

   b ［アプリケーションの起動］ウィンドウで，［選択］をクリックしてから，/usr/local/bin/nnmtelnet を選択します。

   c ［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。

9.3.2 Linux 上の Firefox に SSH を設定する

Linux オペレーティングシステムで SSH プロトコルを使用するように Firefox を設定するには，次の手順に従います。

1. SSH プロトコルを定義する。

a /usr/local/bin/nnmssh ファイルを次の内容で作成します。

```
#!/bin/bash
#
# Linux shell script called by Firefox in response to
# ssh:// URLs for the NNMi SSH menu.
#
address=`echo $1 | cut -d : -f 2 | sed 's;/;;g'`
port=`echo $1 | cut -d : -f 3`
exec /usr/bin/xterm -e ssh $address $port
```

b 誰でも実行可能なスクリプト権限を設定します。

```
chmod 755 /usr/local/bin/nnmssh
```

2. SSH 用の Firefox プリファレンスを設定する。

a Firefox アドレスバーに，about:config と入力します。

b プリファレンスリスト内を右クリックし，［新規］をクリックしてから，［ブール値］をクリックします。

c プリファレンス名network.protocol-handler.expose.ssh を入力します。

d プリファレンス値false を選択します。

3. 新規に定義されたプロトコルを使用するように Firefox を設定する。

a SSH リンクを参照します。

リンクを含む簡易 HTML ファイルを作成，または NNMi コンソールで定義した新規 SSH メニュー項目を使用できます。アドレスバーに直接リンクを入力しても，同じ結果にはなりません。

b ［アプリケーションの起動］ウィンドウで，［選択］をクリックしてから，/usr/local/bin/nnmssh を選択します。

c ［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。

9.4 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。

9.4.1 nnmtelnet.reg の例

このレジストリの内容例は，「9.2.1 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_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Internet Explorer¥MAIN¥FeatureControl
¥FEATURE_DISABLE_TELNET_PROTOCOL]
"iexplore.exe"=dword:00000000
[HKEY_CLASSES_ROOT¥telnet¥shell¥open¥command]
@="¥"C:¥¥Windows¥¥system32¥¥rundll32.exe¥"
¥"C:¥¥Windows¥¥system32¥¥url.dll¥",TelnetProtocolHandler %l"
```

9.4.2 nnmputtytelnet.reg の例

このレジストリの内容例は，「9.2.2 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_LOCAL_MACHINE¥SOFTWARE¥Microsoft¥Internet Explorer¥MAIN¥FeatureControl
¥FEATURE_DISABLE_TELNET_PROTOCOL]
"iexplore.exe"=dword:0c0000000
[HKEY_CLASSES_ROOT¥telnet¥shell¥open¥command]
@="¥"C:¥¥Program Files¥¥PuTTY¥¥putty.exe¥" %l"
```

9.4.3 nnmtelnet32on64.reg の例

このレジストリの内容例は，「9.2.3 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_LOCAL_MACHINE¥SOFTWARE¥Wow6432Node¥Microsoft¥Internet Explorer¥MAIN¥FeatureControl
¥FEATURE_DISABLE_TELNET_PROTOCOL]
```

```
"iexplore.exe"=dword:00000000
[HKEY_CLASSES_ROOT¥Wow6432Node¥telnet¥shell¥open¥command]
@="¥"C:¥¥Program Files¥¥PuTTY¥¥putty.exe¥" %l"
```

9.4.4 nnmssh.reg の例

このレジストリの内容例は，「9.2.4 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。

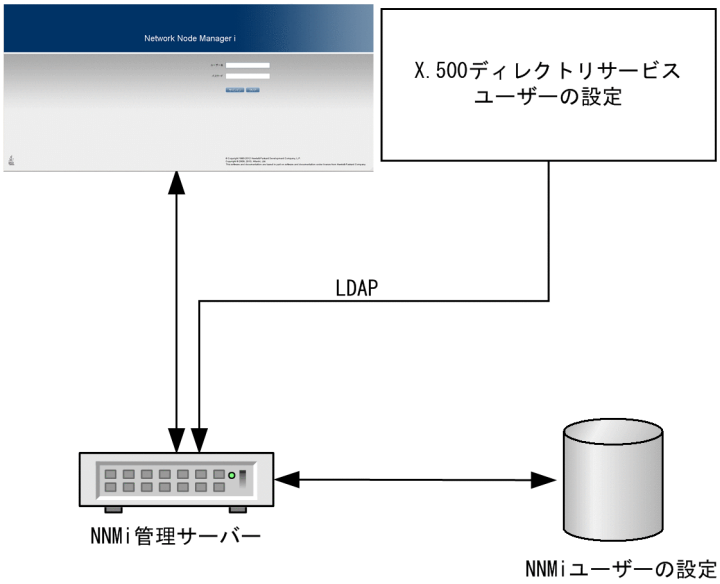
```
Windows Registry Editor Version 5.00
[HKEY_CLASSES_ROOT¥ssh]
@="URL:ssh Protocol"
"EditFlags"=dword:00000002
"FriendlyTypeName"="Secure Shell"
"URL Protocol"=""

[HKEY_CLASSES_ROOT¥ssh¥shell¥open¥command] @="¥"C:¥¥Windows¥¥System32¥¥WScript.exe¥" ¥"c:¥
¥Program Files¥¥PuTTY¥¥ssh.js¥" %l"
```

10 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。

10.1　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。

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，次の表に示します。

表 10-1　ユーザー認証戦略

オプション	ユーザー認証に使用する方法	NNMi でのユーザーアカウントの定義	NNMi でのユーザーグループの定義	グループのメンバーシップに使用する方法
1.内部	NNMi パスワード	あり	あり	NNMi のユーザーアカウントのマッピング
2.混合	LDAP パスワード	あり	あり	NNMi のユーザーアカウントのマッピング
3.外部	LDAP パスワード	なし	あり	LDAP

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。

10.1.1 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

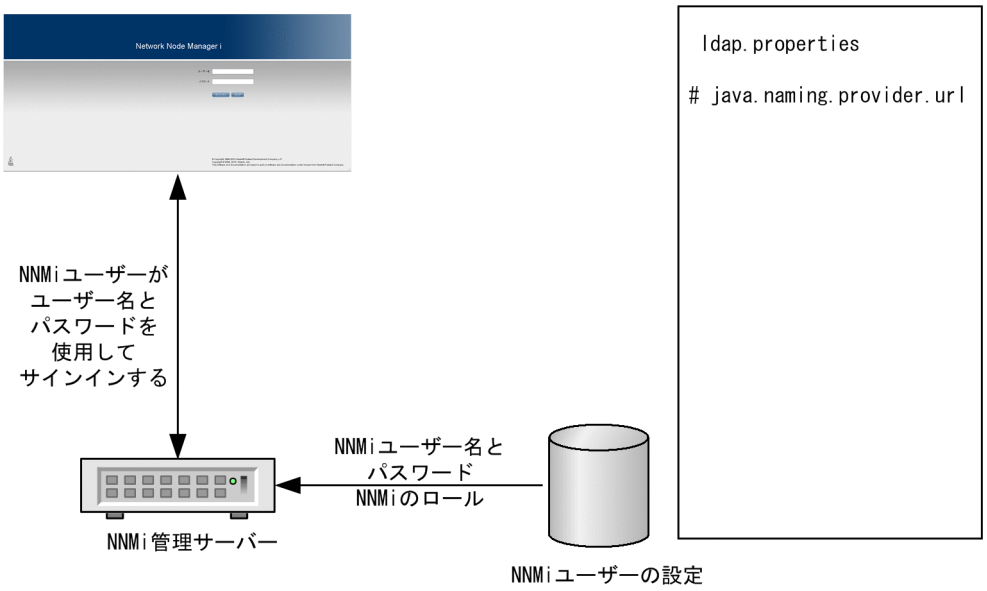
NNMi は，すべてのユーザーアクセス情報を取得するために NNMi データベースにアクセスします。それらの情報は，NNMi 管理者が NNMi コンソールで定義およびメンテナンスします。ユーザーアクセス情報は，NNMi にとってローカルの情報となります。NNMi はディレクトリサービスにアクセスせず，NNMi は次の図のコメント行に示されている`ldap.properties`ファイルを無視します。

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。

NNMi データベースですべてのユーザー情報を設定する方法の詳細については，NNMi 管理者用ヘルプの「*NNMi でアクセスを制御する*」を参照してください。この章を読む必要はありません。

図 10–1　内部モードの NNMi ユーザーサインインの情報フロー



10.1.2　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

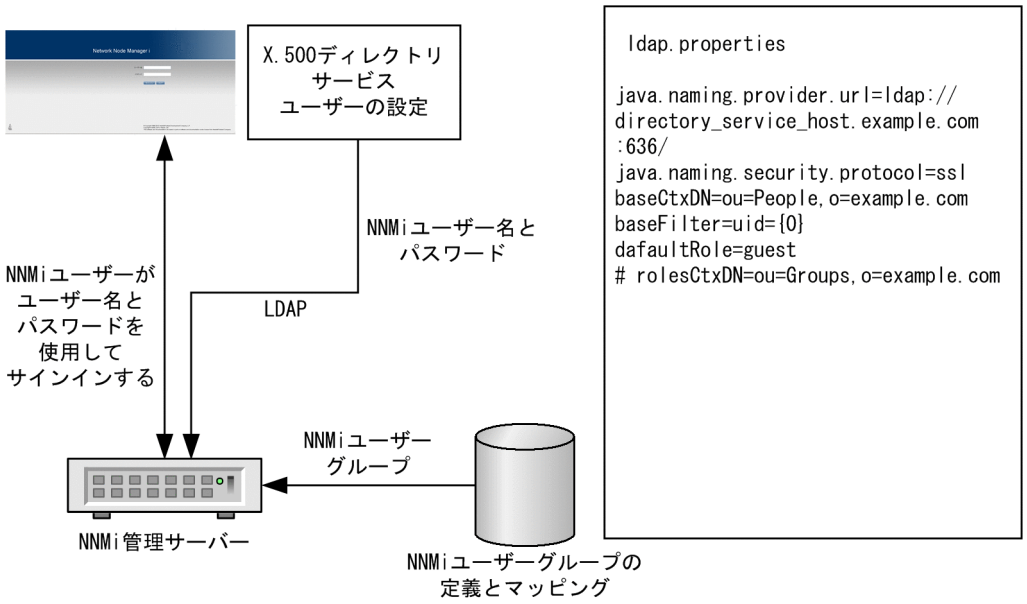
NNMi は，ユーザー名とパスワードを取得するためにディレクトリサービスにアクセスします。それらの情報は，NNMi の外部で定義され，ほかのアプリケーションでも使用できます。ユーザーから NNMi ユーザーグループへのマッピングは，NNMi コンソールでメンテナンスします。NNMi ユーザーアクセス情報の設定およびメンテナンスは，次で説明するように共同で行われます。

- 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 `ldap.properties` 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 10–2　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簡単には拡張できない。

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「*ディレクトリサービスおよび NNMi を使用してアクセスを制御する*」を参照してください。

10.1.3　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。

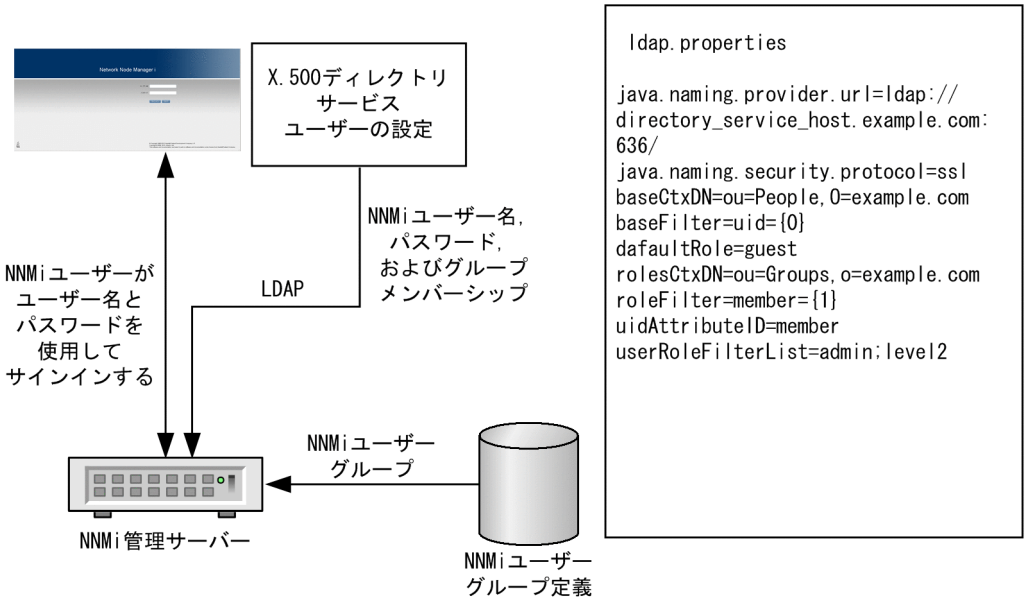
NNMi ユーザーアクセス情報の設定およびメンテナンスは，次で説明するように共同で行われます。

- 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 ldap.properties 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 10-3　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「*ディレクトリサービスを使用してアクセスを制御する*」を参照してください。

10.2　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，次のファイルで設定されています。

- Windows：%NNM_SHARED_CONF%¥ldap.properties
- UNIX：$NNM_SHARED_CONF/ldap.properties

このファイルの詳細については，「10.7　ldap.properties 設定ファイルリファレンス」を参照してください。「10.7.1　properties 設定ファイルの例」も参照してください。

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，「10.4　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。

- 10.2.1　タスク 1：現在の NNMi ユーザー情報をバックアップする
- 10.2.2　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
- 10.2.3　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
- 10.2.4　タスク 4：ユーザー名とパスワードの設定をテストする
- 10.2.9　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
- 10.2.10　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。

- 10.2.1　タスク 1：現在の NNMi ユーザー情報をバックアップする
- 10.2.2　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
- 10.2.3　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
- 10.2.4　タスク 4：ユーザー名とパスワードの設定をテストする
- 10.2.5　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

  参考

  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，「10.5　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。

- 10.2.6　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
- 10.2.7　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

- 10.2.8　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
- 10.2.9　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
- 10.2.10　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

10.2.1　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
```

10.2.2　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，「8.8　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 トラストストアにインポートします。

10.2.3　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

このタスクは，「混合モード」および「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。このタスクには，次のセクションが含まれます。

- (1)　Microsoft Active Directory の場合の簡単な方法
- (2)　ほかのディレクトリサービスの場合の簡単な方法

設定の詳細な手順については，「10.4.4　ユーザー識別」を参照してください。

(1)　Microsoft Active Directory の場合の簡単な方法

1. NNMi に付属するldap.properties 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く。
2. ファイルの内容を次のテキストで上書きする。

```
java.naming.provider.url=ldap://<myldapserver>:389/

bindDN=<mydomain>¥¥<myusername>
bindCredential=<mypassword>
```

```
baseCtxDN=CN=Users,DC=<mycompanyname>,DC=<mysuffix>
baseFilter=CN={0}

defaultRole=guest

#rolesCtxDN=CN=Users,DC=<mycompanyname>,DC=<mysuffix>

roleFilter=member={1}
uidAttributeID=member
userRoleFilterList=admin;level2;level1
```

3. ディレクトリサービスにアクセスするときの URL を指定する。

手順 1.のテキストには次の行があります。

```
java.naming.provider.url=ldap://<myldapserver>:389/
```

<*myldapserver*>を，Active Directory サーバーの完全修飾ホスト名（例：myserver.example.com）で置き換えます。

複数のディレクトリサービス URL を指定するには，各 URL をスペース文字 1 つで区切ります。

4. 有効なディレクトリサービスユーザーの資格証明を指定する。

手順 1.のテキストには次の行があります。

```
bindDN=<mydomain>¥¥<myusername>
bindCredential=<mypassword>
```

次のように置き換えます。

- <*mydomain*>を Active Directory ドメインの NetBIOS 名で置き換えます。
- <*myusername*>および<*mypassword*>を Active Directory サーバーにアクセスするときに使用するユーザー名とパスワードで置き換えます。パスワードは平文で保存されるため，ディレクトリサービスへの読み取り専用アクセス権を付与してユーザー名を指定してください。

5. ディレクトリサーバードメインの中でユーザーレコードを保存する部分を指定する。

手順 1.のテキストには次の行があります。

```
baseCtxDN=CN=Users,DC=<mycompanyname>,
DC=<mysuffix>
```

<*mycompanyname*>，および<*mysuffix*>を Active Directory サーバーの完全修飾ホスト名のコンポーネントで置き換えます（例えばホスト名myserver.example.com の場合は，DC=example,DC=com と指定します）。

(2) ほかのディレクトリサービスの場合の簡単な方法

1. NNMi に付属するldap.properties 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く。

2. ディレクトリサービスにアクセスするときの URL を指定する。

手順 1.のテキストには次の行があります。

```
#java.naming.provider.url=ldap://<myldapserver>:389/
```

次を実行します。

- 行のコメントを解除します（#文字を削除します）。
- <*myldapserver*>をディレクトリサーバーの完全修飾ホスト名で置き換えます（例：myserver.example.com）。

  複数のディレクトリサービス URL を指定するには，各 URL をスペース文字 1 つで区切ります。

3. ディレクトリサーバードメインの中でユーザーレコードを保存する部分を指定する。

手順 1.のテキストには次の行があります。

```
baseCtxDN=ou=People,o=myco.com
```

ou=People,o=myco.com をユーザーレコードを保存するディレクトリサービスドメインの部分で置き換えます。

4. NNMi にサインインするユーザー名の形式を指定する。

手順 1.のテキストには次の行があります。

```
baseFilter=uid={0}
```

uid をディレクトリサービスドメインのユーザー名属性で置き換えます。

10.2.4 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.properties ファイルで，テスト用にdefaultRole=guest と設定する。

この値はいつでも変更できます。

2. ldap.properties ファイルを保存する。

3. 次のコマンドを実行して，NNMi にldap.properties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。

- 「10.2.3 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。
- 「10.4.4　ユーザー識別」の詳細な設定プロセスに従います。

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，「10.2.9 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。

10.2.5　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。このタスクには，次のセクションが含まれます。

- (1)　Microsoft Active Directory の場合の簡単な方法
- (2)　ほかのディレクトリサービスの場合の簡単な方法

設定の詳細な手順については，「10.4.5　ユーザーグループ識別」を参照してください。

(1)　Microsoft Active Directory の場合の簡単な方法

1. ldap.properties 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く。

2. 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する。

手順 1.のテキストには次の行があります。

```
#rolesCtxDN=CN=Users,DC=<mycompanyname>,
DC=<mysuffix>
```

次を実行します。

- 行のコメントを解除します（#文字を削除します）。
- *<mycompanyname>*，および*<mysuffix>*を Active Directory サーバーの完全修飾ホスト名のコンポーネントで置き換えます（例えばホスト名myserver.example.com の場合は，DC=example,DC=com と指定します）。

(2) ほかのディレクトリサービスの場合の簡単な方法

1. ldap.properties 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く。
2. 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する。

   手順 1.のテキストには次の行があります。

   ```
   #rolesCtxDN=ou=Groups,o=myco.com
   ```

   次を実行します。

   - 行のコメントを解除します（#文字を削除します）。
   - ou=Groups,o=myco.com を，ディレクトリサービスドメインのグループレコードを保存する部分で置き換えます。
3. ディレクトリサービスのグループ定義でグループメンバー名の形式を指定する。

   手順 1.のテキストには次の行があります。

   ```
   roleFilter=member={1}
   ```

   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。

10.2.6 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。

10.2.7　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. ldap.properties ファイルを保存する。

2. 次のコマンドを実行して，NNMi に ldap.properties 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。

- 「10.2.5 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，「10.2.6 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。
- 「10.4.5　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。

10.2.8　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. ldap.properties 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く。

2. インシデントを割り当てることができる NNMi ロールを NNMi オペレータが指定するように，userRoleFilterList パラメータ値を変更する。

   1 つ以上の定義済み NNMi ユーザーグループ名の一意の名前をセミコロンで区切ったリスト形式です。定義済みの NNMi ユーザーグループの一意の名前については，「10.4.5　ユーザーグループ識別」の「表 10-4　NNMi ユーザーグループ名のマッピング」を参照してください。

3. ldap.properties ファイルを保存する。

4. 次のコマンドを実行して，NNMi にldap.propertiesファイルを再読み込みさせる。

   ```
   nnmldap.ovpl -reload
   ```

5. 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。

6. 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。

7. 設定した各 NNMi ロールにインシデントを割り当てることができるまで，手順 1.から手順 6.の操作を繰り返す。

10.2.9 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.properties ファイルで，defaultRole パラメータの値を変更するか，またはコメントにする。

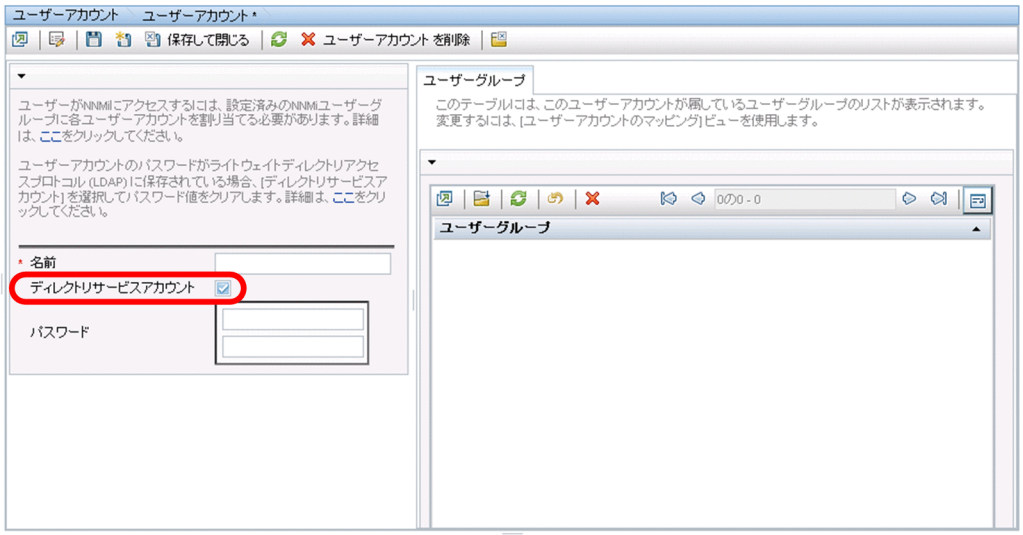
2. (「混合モード」の設定だけ) NNMi データベースにユーザーグループメンバーシップを保存するには，次の手順を実行して，NNMi データベースのユーザーアクセス情報をリセットする。

   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。

10.2.10　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。

10.3 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

ここでは，ldap.properties ファイルをバージョン 09-50 の NNMi から改訂して，ユーザーごとに複数の NNMi ユーザーグループをサポートする方法について説明します。このバージョンアップは，次の条件の両方で必要となります。

- ldap.properties ファイルによって，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が現在有効になっています。
  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する方法については，「10.1.3 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」を参照してください。
- NNMi をカスタムセキュリティグループで設定したか，設定することになっています。

バージョン 09-50 の NNMi では，NNMi ユーザーは，定義済みの NNMi ロールのうちどれかに割り当てられていました。各ユーザーは，NNMi トポロジのすべてのオブジェクトにアクセスできました。

バージョン 10 の NNMi では，定義済みの NNMi ユーザーグループで NNMi ロールが置き換わります。各 NNMi ユーザーは最低 1 つの定義済み NNMi ユーザーグループに属する必要があり，これによって NNMi ユーザーが NNMi コンソールで実行できる事項が定義されます。追加のユーザーグループが存在する場合は，次のように NNMi トポロジオブジェクトへのアクセスを制限します。

- カスタムユーザーグループが存在しない場合，すべての NNMi コンソールユーザーはすべてのトポロジオブジェクトにアクセスできます。
- 1 つ以上のカスタムユーザーグループが存在する場合，各ユーザーグループは NNMi トポロジのオブジェクトのサブセットにアクセスできます。

バージョン 09-50 の NNMi では，各ディレクトリサービスグループ定義に，NNMi ロールを指定するグループ属性を含める必要がありました。ldap.properties ファイルの次のパラメータで，このグループ属性を指定していました。

- roleAttributeID
- roleAttributeIsDN
- roleNameAttributeID

バージョン 10 の NNMi では，このパラメータが廃止されます。各ユーザーグループを NNMi コンソールで定義する必要があります。

ユーザーグループ定義には外部名を含めます。これが，ディレクトリサービスでのグループの識別名になります。

ディレクトリサービスのアクセス設定を変更して NNMi セキュリティモデルをサポートするには，次の手順を実行します。

1. NNMi データベースのユーザー情報をバックアップする。

```
nnmconfigexport.ovpl -c account -u <user> ¥
-p <password> -f NNMi_database_accounts.xml
```

2. ldap.properties 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く。

   ldap.properties ファイルの詳細については，「10.7　ldap.properties 設定ファイルリファレンス」を参照してください。

3. 次のパラメータが存在する場合は，コメントにするか削除する。

   - roleAttributeID
   - roleAttributeIsDN
   - roleNameAttributeID

4. ldap.properties ファイルを編集した場合は，次のコマンドを実行して NNMi に LDAP 設定を再読み込みさせる。

```
nnmldap.ovpl -reload
```

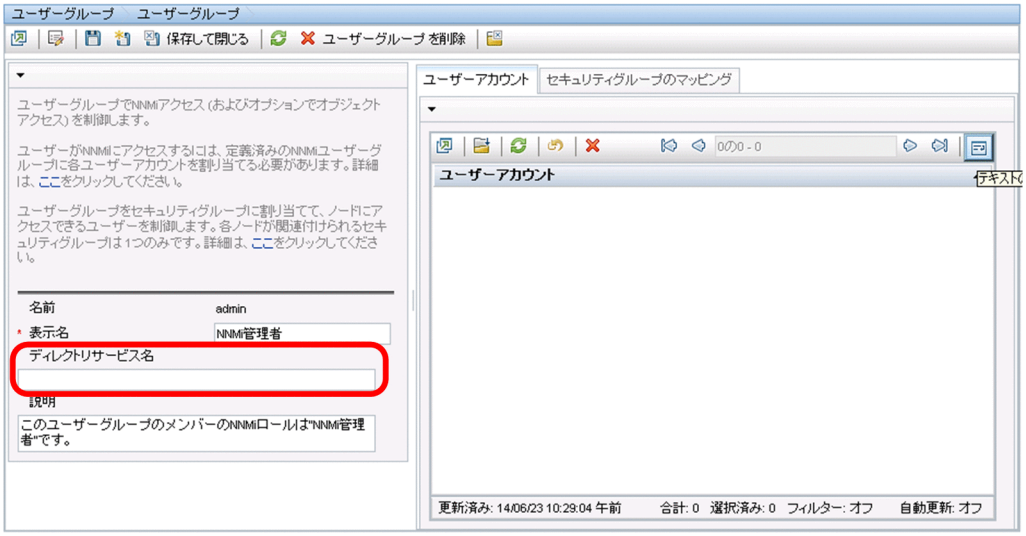
5. NNMi コンソールで，定義済みの NNMi ユーザーグループをディレクトリサービスのユーザーグループにマッピングする。

   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。

6. ディレクトリサービスで NNMi ユーザーの追加グループを識別します。必要に応じて新しいグループを定義する。

7. 手順 6.で追加された新しいグループごとに，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 ユーザーの新しい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。

このマッピングで，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。

8. （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「*セキュリティの設定*」を参照してください。

10.4　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。

10.4.1　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，「10.4.2　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，設定パラメータの有効性を検証できます。

10.4.2　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，UNIX 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
```

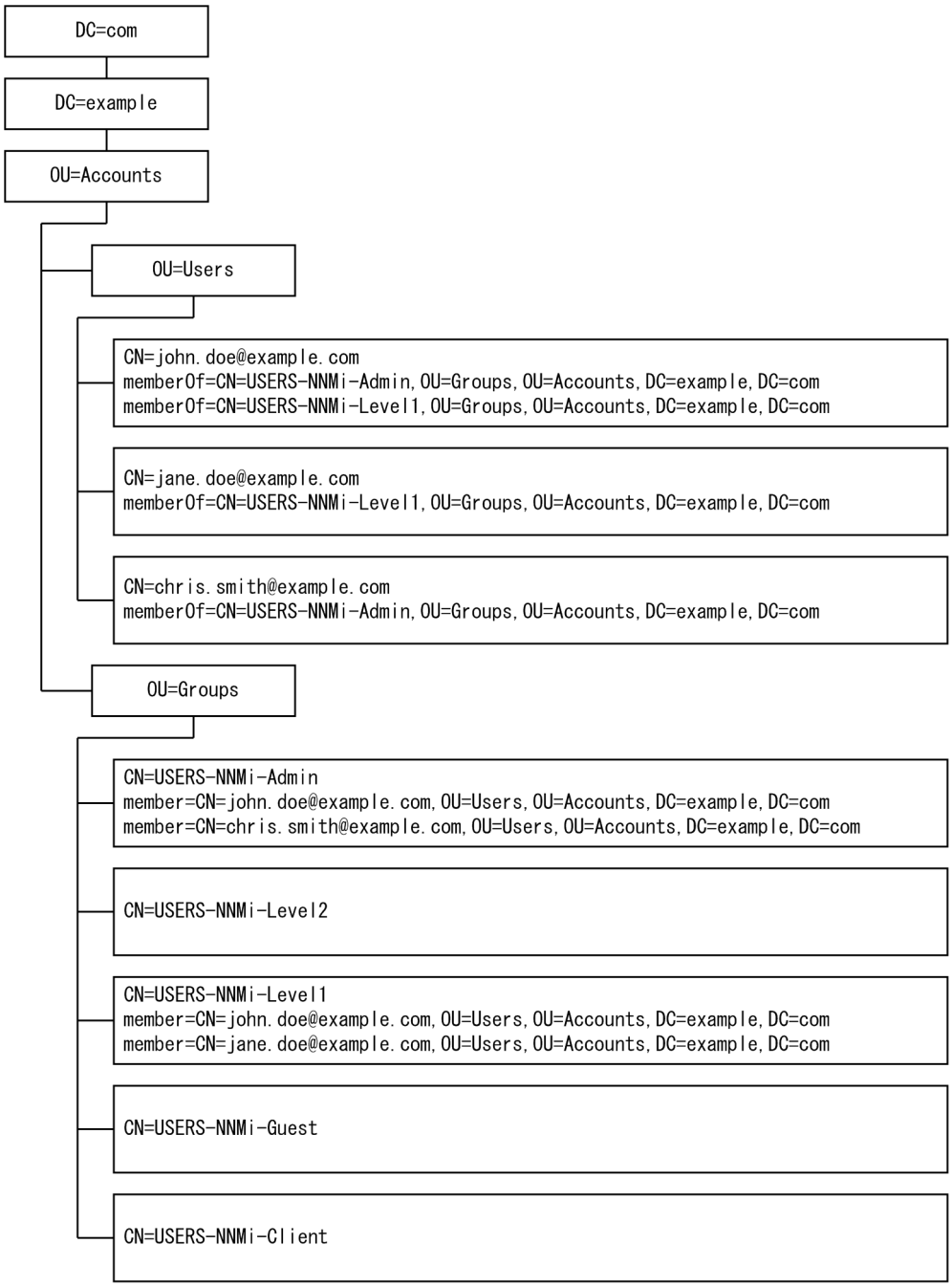
- ディレクトリサービスユーザー ID を保存するグループ属性：member

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 10-4　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：

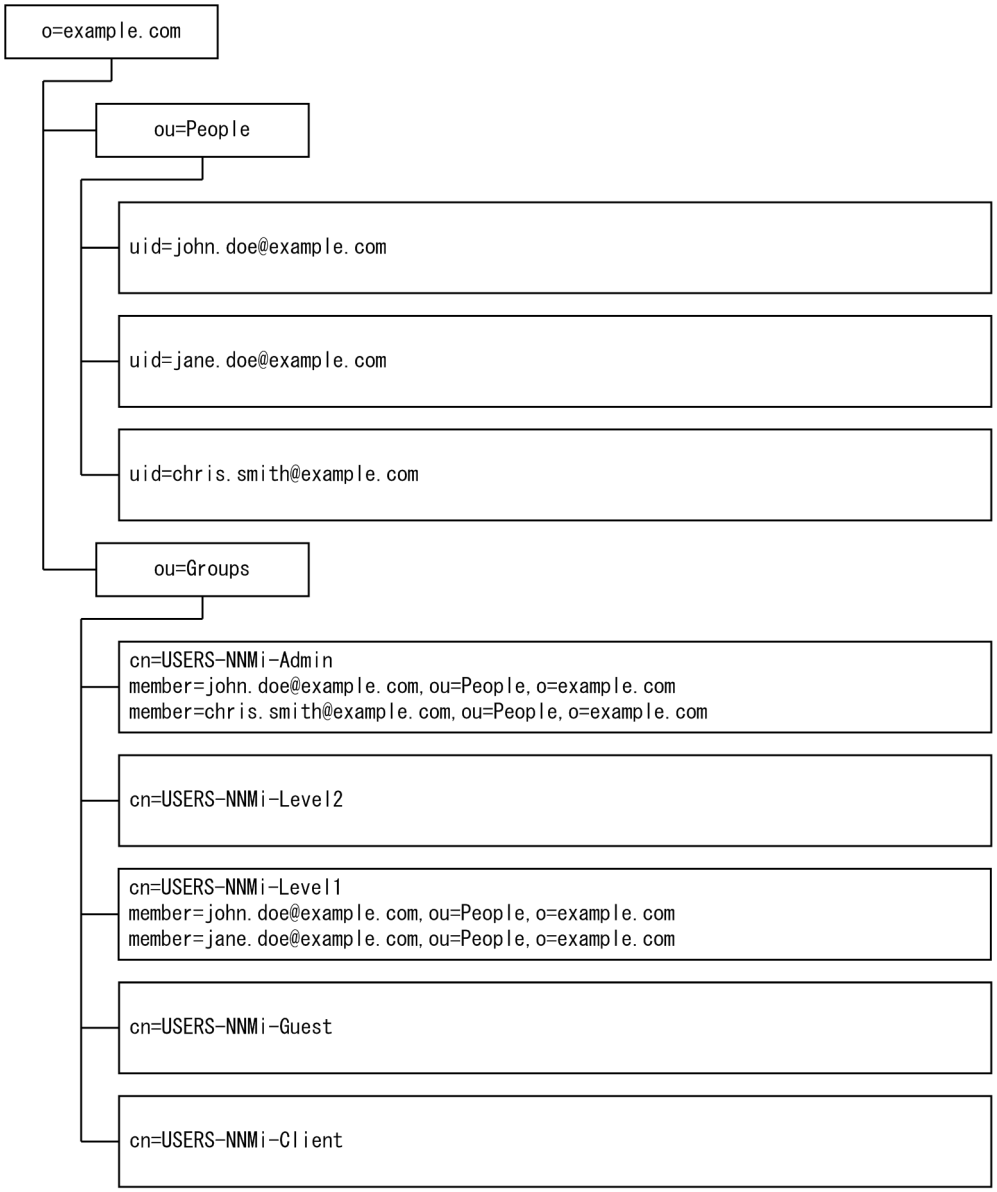
```
cn=USERS-NNMi-Admin,ou=Groups,o=example.com
```

- 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 10-5　ほかのディレクトリサービスのドメインの例



10.4.3　ディレクトリサービス管理者が所有する情報

表 10-2 および表 10-3 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 10-2 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 10-2 と表 10-3 の情報を収集します。

表 10-2　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証明書のコピーを取得し，「8.8 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 10-3　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-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-Admin, ou=Groups,o=example.com • cn=USERS-NNMi-Level2, 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

情報	Active Directory 例	そのほかのディレクトリサービス例
	• CN=USERS-NNMi-Guest, OU=Groups,OU=Accounts, DC=example,DC=com	

10.4.4 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.properties ファイルでのユーザー識別名は，baseFilter 値とbaseCtxDN 値を連結した値です。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.properties ファイルのdefaultRole パラメータの値
  - 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.properties ファイルのdefaultRole パラメータの値
  - 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

baseFilter がCN={0}に，baseCtxDN がOU=Users,OU=Accounts,DC=example,DC=com に設定されている場合，ユーザーが NNMi にjohn.doe としてサインインすると，次の文字列がディレクトリサービスに渡されます。

```
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

baseFilter がuid={0}@example.com に，baseCtxDN がou=People,o=example.com に設定されている場合，ユーザーが NNMi にjohn.doe としてサインインすると，次の文字列がディレクトリサービスに渡されます。

```
uid=john.doe@example.com,ou=People,o=example.com
```

(1) 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）

「10.2 ディレクトリサービスへのアクセスを設定する」の「10.2.3 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」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. ディレクトリサービス管理者から，「10.4.3 ディレクトリサービス管理者が所有する情報」の「表 10-2 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」に示す情報を取得する。

2. 適切な手順を完了し，ディレクトリサービスでのユーザー名の形式を確認する。
   - Active Directory およびその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：「(2) 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）」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：「(3) 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）」を参照してください。

3. 任意のテキストエディタでldap.properties ファイルを開く。

   ldap.properties ファイルの詳細については，「10.7 ldap.properties 設定ファイルリファレンス」を参照してください。

4. java.naming.provider.url パラメータを，LDAP によってディレクトリサービスにアクセスする場合の URL に設定する。
   - LDAP ブラウザを使用する方法：LDAP ブラウザ設定からこの情報を入手します。
   - Web ブラウザを使用する方法：「(3) 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）」から<*ディレクトリサービスホスト*>と<*ポート*>の値を含めます。

   複数のディレクトリサービス URL を指定するには，各 URL をスペース文字 1 つで区切ります。

5. ディレクトリサービスへのセキュア通信を設定した場合は，次の行のコメントを解除（または追加）する。

   ```
   java.naming.security.protocol=ssl
   ```

6. (Active Directory だけ) bindDN およびbindCredential パラメータを次のように設定する。

- <*mydomain*>を Active Directory ドメインの名前で置き換えます。
- <*myusername*>および<*mypassword*>を Active Directory サーバーにアクセスするときに使用するユーザー名とパスワードで置き換えます。パスワードは平文で保存されるため，ディレクトリサービスへの読み取り専用アクセス権を付与してユーザー名を指定してください。

7. baseCtxDN パラメータを，複数のユーザーで同じになっている，識別ユーザー名のエレメントに設定する。

8. NNMi のサインインで入力するときのユーザー名が，ディレクトリサービスでユーザー名が保存されるときの方法と相関するように，baseFilter パラメータを設定する。

   この値は，ユーザーごとに変更される識別ユーザー名のエレメントです。実際のユーザー名を式{0}で置き換えます。

9. 「10.2　ディレクトリサービスへのアクセスを設定する」の「10.2.4 タスク 4：ユーザー名とパスワードの設定をテストする」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）

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. ユーザーのグループを識別し，そのグループに関連づけられているユーザーの識別名の形式を調べる。

(3)　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）

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>/<user_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。
   - <*user_search_string*>は，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です。

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。

   - 要求が時間切れになったり，ディレクトリサービスに到達できなかったことを示すメッセージが表示される場合は，<*directory_service_host*>と<*port*>の値を確認してから，手順 1.を繰り返してください。
   - 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，<*user_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。

- 該当するユーザーレコードが表示された場合，そのアクセス情報は正しいことになります。<*user_search_string*>の値は，識別ユーザー名です。

10.4.5　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.properties` ファイルの`defaultRole` および `userRoleFilterList` パラメータも，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 10-4　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`
オペレータレベル 2	`level2`
オペレータレベル 1	`level1`
ゲスト	`guest`
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（`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）

「10.2　ディレクトリサービスへのアクセスを設定する」の「10.2.5　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. ディレクトリサービス管理者から，「10.4.3　ディレクトリサービス管理者が所有する情報」の「表 10-3　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」に示す情報を取得する。

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. 任意のテキストエディタで ldap.properties ファイルを開く。

   ldap.properties ファイルの詳細については，「10.7　ldap.properties 設定ファイルリファレンス」を参照してください。

4. rolesCtxDN パラメータを，複数のグループで同じになっている，識別グループ名のエレメントに設定する。

5. 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，roleFilter パラメータを設定する。実際のユーザー名を次の式のどちらかで置き換える。
   - 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（例えば，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（例えば，uid=john.doe@example.com,ou=People，o=example.com）。

6. uidAttributeID パラメータを，ユーザー ID を保存するグループ属性の名前に設定する。

7. 「10.2　ディレクトリサービスへのアクセスを設定する」の「10.2.7　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」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。

10.5 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。

10.6　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.properties 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.properties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，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。

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（UNIX）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，「8.8　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.properties ファイルのsearchTimeLimit の値を増やします。

```
javax.naming.TimeLimitExceededException : [LDAP: error code 3 - Timelimit Exceeded]
```

10.7 ldap.properties 設定ファイルリファレンス

ldap.properties ファイルには，ディレクトリサービスと通信し，それに対する LDAP クエリーを作成する場合の設定が保存されています。このファイルは次の場所にあります。

- Windows：%NNM_SHARED_CONF%¥ldap.properties
- UNIX：$NNM_SHARED_CONF/ldap.properties

ldap.properties ファイルでは，次の規則が適用されます。

- 行をコメントにするには，その行の先頭をコメント記号（#）にします。
- 特殊記号については，次のルールが適用されます。
  - 円記号（¥），コンマ（,），セミコロン（;），プラス記号（+），大なり記号（<），小なり記号（>）を指定する場合は，円記号でエスケープします。（例: ¥¥ や ¥+）
  - 半角スペースを先頭または最終文字として指定する場合は，半角スペースを円記号（¥）でエスケープします。
  - シャープ記号（#）を先頭文字として指定する場合は，円記号（¥）でエスケープします。

> **参考**
>
> ldap.properties ファイルを編集したら，次のコマンドを実行して NNMi に LDAP 設定を再読み込みさせます。
>
> ```
> nnmldap.ovpl -reload
> ```

次の表に，ldap.properties ファイルのパラメータの説明を示します。

表 10-5 ldap.properties ファイルのパラメータ

パラメータ	説明
java.naming.provider.url	ディレクトリサービスにアクセスするときの URL を指定します。 URL は，プロトコル（ldap）のあとにディレクトリサービスの完全修飾ホスト名が続き，任意でさらにポート番号が続く形式で指定します。 例： `java.naming.provider.url=ldap://ldap.example.com:389/` ポート番号を省略すると，次のデフォルト値が適用されます。 • 非 SSL 接続の場合，デフォルト値は 389 です。 • SSL 接続の場合，デフォルト値は 636 です。 複数のディレクトリサービスの URL を指定すると，NNMi は可能な限り最初のディレクトリサービスを使用します。そのディレクトリサービスにアクセスできない場合，NNMi はリスト内の次のディレクトリサービスにクエリーを実行し，以降同様に対処します。各 URL は 1 つのスペース文字で区切ります。

パラメータ	説明
	例： `java.naming.provider.url=ldap://ldap1.example.com/ ldap://ldap2.example.com/` このパラメータを設定すると，NNMi とディレクトリサービス間の LDAP 通信が有効になります。LDAP 通信を無効にするには，このパラメータをコメントにしてからファイルを保存します。これによって NNMi は，ldap.properties ファイルの設定を無視します。
java.naming.security.protocol	接続プロトコル指定します。 • LDAP over SSL を使用するようにディレクトリサーバーが設定されている場合は，このパラメータをssl に設定します。 例： `java.naming.security.protocol=ssl` • ディレクトリサービスで SSL が不要な場合は，このパラメータをコメントにしたままにします。 詳細については，「8.8　ディレクトリサービスへの SSL 接続を設定する」を参照してください。
bindDN	匿名アクセスを許可しない（Active Directory などの）ディレクトリサービスの場合は，そのディレクトリサービスにアクセスするユーザー名を指定します。このユーザー名のパスワードはldap.properties ファイルに平文で保存されるため，ディレクトリサービスへの読み取り専用アクセス権を持つユーザー名を選択してください。 例： `bindDN=region1¥¥john.doe@example.com`
bindCredential	bindDN が設定されている場合は，そのbindDN によって識別されるユーザー名のパスワードを指定します。 例： `bindCredential=PasswordForJohnDoe`
baseCtxDN	ディレクトリサーバードメインの中でユーザーレコードを保存する部分を識別します。形式は，ディレクトリサービスの属性名と値のコンマ区切りリストです。 例： • `baseCtxDN=CN=Users,DC=ldapserver,DC=example,DC=com` • `baseCtxDN=ou=People,o=example.com` 詳細については，「10.4.4　ユーザー識別」を参照してください。
baseFilter	NNMi にサインインするユーザー名の形式を指定します。形式は，ディレクトリサービスのユーザー名属性の名前と，入力したユーザーサインイン名をディレクトリサービス内の名前の形式に関連づける文字列で構成されます。ユーザー名文字列には，式{0}（サインインで入力されたユーザー名を示す）と，ユーザー名のディレクトリサービス形式を照合するために必要なほかの文字が含まれます。 • NNMi のサインインで入力されたユーザー名がディレクトリサービスに保存されているユーザー名と同じ場合，値は置換表現になります。 例： － `baseFilter=CN={0}`

パラメータ	説明
	— baseFilter=uid={0} • NNMi のサインインで入力したユーザー名がディレクトリサービスに保存されているユーザー名のサブセットになっている場合は，値に追加の文字を含めます。 例： — baseFilter=CN={0}@example.com — baseFilter=uid={0}@example.com 詳細については，「10.4.4　ユーザー識別」を参照してください。
defaultRole	任意で指定。LDAP に従って NNMi にサインインするディレクトリサービスユーザーすべてに適用されるデフォルトロールを指定します。このパラメータの値は，（NNMi データベースまたはディレクトリサービスでの）ユーザーグループマッピングの保存場所に関係なく適用されます。 定義済みの NNMi ユーザーグループにユーザーが直接設定されている場合，NNMi は，デフォルトロールおよび割り当て済みユーザーグループの権限のスーパーセットをユーザーに付与します。 有効な値は，admin，level2，level1，またはguest です。 この名前は，定義済み NNMi ユーザーグループ名の一意の名前です。定義済み NNMi ユーザーグループ名の一意の名前については，「10.4.5　ユーザーグループ識別」の「表 10-4」を参照してください。 （例） defaultRole=guest コメントにするか，または省略すると，NNMi はデフォルト値を使用しません。
rolesCtxDN	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コンマ区切りリストです。 例： • rolesCtxDN=CN=Users,DC=ldapserver,DC=example,DC=com • rolesCtxDN=ou=Groups,o=example.com ほかのディレクトリサービス（Active Directory 以外）では，検索速度を高めるため，NNMi ユーザーグループを含むディレクトリサービスグループを 1 つ以上指定できます。グループ名にパターンがある場合は，ワイルドカードを指定できます。例えば，ディレクトリサービスにUSERS-NNMi-administrators やUSERS-NNMi-level1Operators などの名前のグループが含まれる場合は，次のような検索コンテキストを使用できます。 rolesCtxDN=cn=USERS-NNMi-*,ou=Groups,o=example.com このパラメータを設定すると，LDAP を介した NNMi ユーザーグループ割り当てのディレクトリサービスのクエリーが有効になります。LDAP を介した NNMi ユーザーグループ割り当てのディレクトリサービスのクエリーを無効にするには，このパラメータをコメントにしてからファイルを保存します。NNMi は，ldap.properties ファイルにある残りのユーザーグループ関連の値を無視します。 詳細については，「10.4.5　ユーザーグループ識別」を参照してください。
roleFilter	ディレクトリサービスのグループ定義でグループメンバー名の形式を指定します。形式は，ユーザー ID のディレクトリサービスグループ属性の名前と，入力したユーザーサインイン名をディレクトリサービス内のユーザー ID の形式に関連づける文字列で

パラメータ	説明
	構成されます。ユーザー名文字列には，次の式の 1 つと，グループメンバー名のディレクトリサービス形式を照合するために必要なほかの文字が含まれています。 • 式{0}は，サインインで入力されたユーザー名を示します（例えば，john.doe）。サインインで入力される（短い）ユーザー名で照合するロールフィルタ 例： roleFilter=member={0} • 式{1}は，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します（例えば，CN=john.doe@example.com,OU=Users,OU=Accounts,DC=example,DC=com または uid=john.doe@example.com,ou=People,o=example.com）。 （完全に）認証されたユーザー名で照合するロールフィルタ 例： roleFilter=member={1} 詳細については，「10.4.5　ユーザーグループ識別」を参照してください。
uidAttributeID	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を指定します。 例： uidAttributeID=member 詳細については，「10.4.5　ユーザーグループ識別」を参照してください。
userRoleFilterList	任意で指定。NNMi コンソールで関連ユーザーにインシデントを割り当てることができる NNMi ユーザーグループを制限します。このリストのユーザーグループは，LDAP によって認証されるディレクトリサービスユーザー名だけに適用されます。このパラメータでは，NNMi ユーザーグループが NNMi コンソールで割り当てられて，NNMi データベースに保存されるときに使用できない機能が提供されます。1 つ以上の定義済み NNMi ユーザーグループ名の一意の名前をセミコロンで区切ったリストという形式です。定義済みの NNMi ユーザーグループの一意の名前については，「10.4.5　ユーザーグループ識別」の「表 10-4」を参照してください。 （例） userRoleFilterList=admin;level2;level1
searchTimeLimit	任意で指定。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=10000

注　初期の ldap.properties ファイルには，この表のリストにあるパラメータの一部が含まれていない場合があります。必要なパラメータを追加してください。

10.7.1　properties 設定ファイルの例

Active Directory の場合の ldap.properties ファイルの例

Active Directory の場合の ldap.properties ファイルの例を次に示します。

```
java.naming.provider.url=ldap://MYldapserver.example.com:389/
bindDN=MYdomain¥¥MYusername
bindCredential=MYpassword
baseCtxDN=CN=Users,DC=MYldapserver,DC=EXAMPLE,DC=com
baseFilter=CN={0}
defaultRole=guest
rolesCtxDN=CN=Users,DC=MYldapserver,DC=EXAMPLE,DC=com
roleFilter=member={1}
uidAttributeID=member
userRoleFilterList=admin;level2;level1
```

ほかのディレクトリサービスの場合の ldap.properties ファイルの例

ほかのディレクトリサービスの場合のldap.properties ファイルの例を次に示します。

```
java.naming.provider.url=ldap://MYldapserver.example.com:389/
baseCtxDN=ou=People,o=EXAMPLE.com
baseFilter=uid={0}
defaultRole=guest
rolesCtxDN=ou=Groups,o=EXAMPLE.com
roleFilter=member={1}
uidAttributeID=member
userRoleFilterList=admin;level2;level1
```

11 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。

11.1 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。

11.2 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

11.3 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。

11.4　NNMi に NAT を実装する方法

NNMi では，テナントを使用して NAT 環境を管理します。テナントは，論理グループの概念で，ノードグループ，マッピング，およびセキュリティサポートが提供されます。インターネットプロバイダのネットワーク内の顧客がテナントの例として挙げられます。インターネットプロバイダは，ネットワーク内で動的 NAT を使用して内部 IP アドレスを再利用していることがあります。このような場合，NNMi ではリージョナルマネージャを使用してネットワーク内の各顧客を管理し，適切なネットワークセキュリティを確保します。つまり，1 つのテナント（顧客）はリージョナルネットワーク内の別のテナント（顧客）と通信できなくなります。テナントの詳細については，「12.　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。

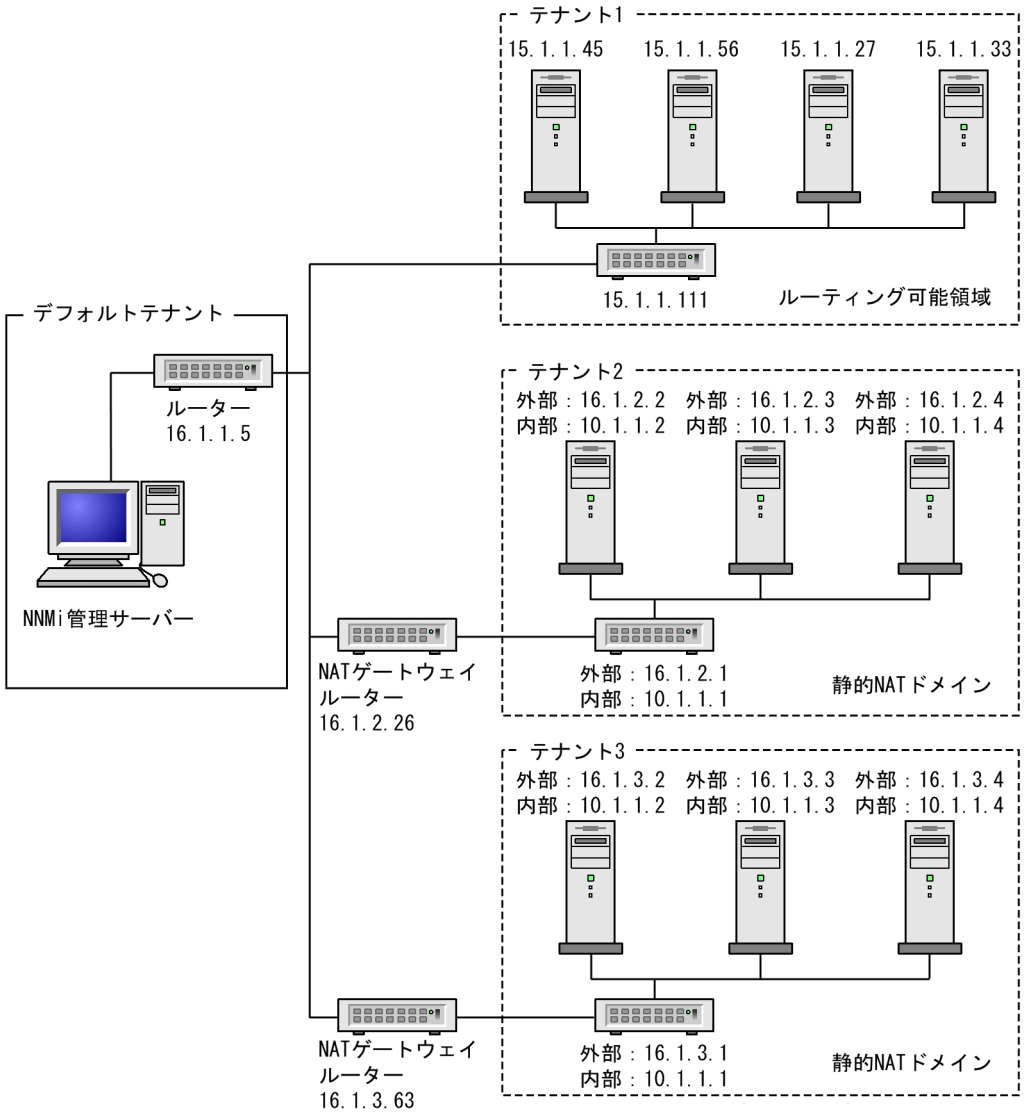
ネットワーク管理環境に重複アドレスドメインが含まれている場合，最低でも一意のテナントとして各ドメインを設定する必要があります。使用しているプロトコルによって，NNMi の実装方法や要件は異なる場合があります。例えば，動的 NAT または動的 PAT を使用している場合，追加のハードウェアおよびライセンスが必要になります。使用している NAT プロトコルのタイプに基づいて，後続の適切な項を参照してください。

11.5　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，「12.　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 11–1　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。

11.5.1 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。

11.5.2 静的 NAT での通信

(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

NNMi では，NAT 環境に存在するノードも含めてすべてのノードの ICMP 管理アドレスポーリングが自動的に有効になります。NNMi は NAT 環境で次のように動作します。

- ［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 11-1　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 11-2　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

11.5.3　検出と静的 NAT

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 NAT 領域内に存在することがあります。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）が必要になります。詳細については，NNMi ヘルプを参照してください。

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。

11.5.4　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

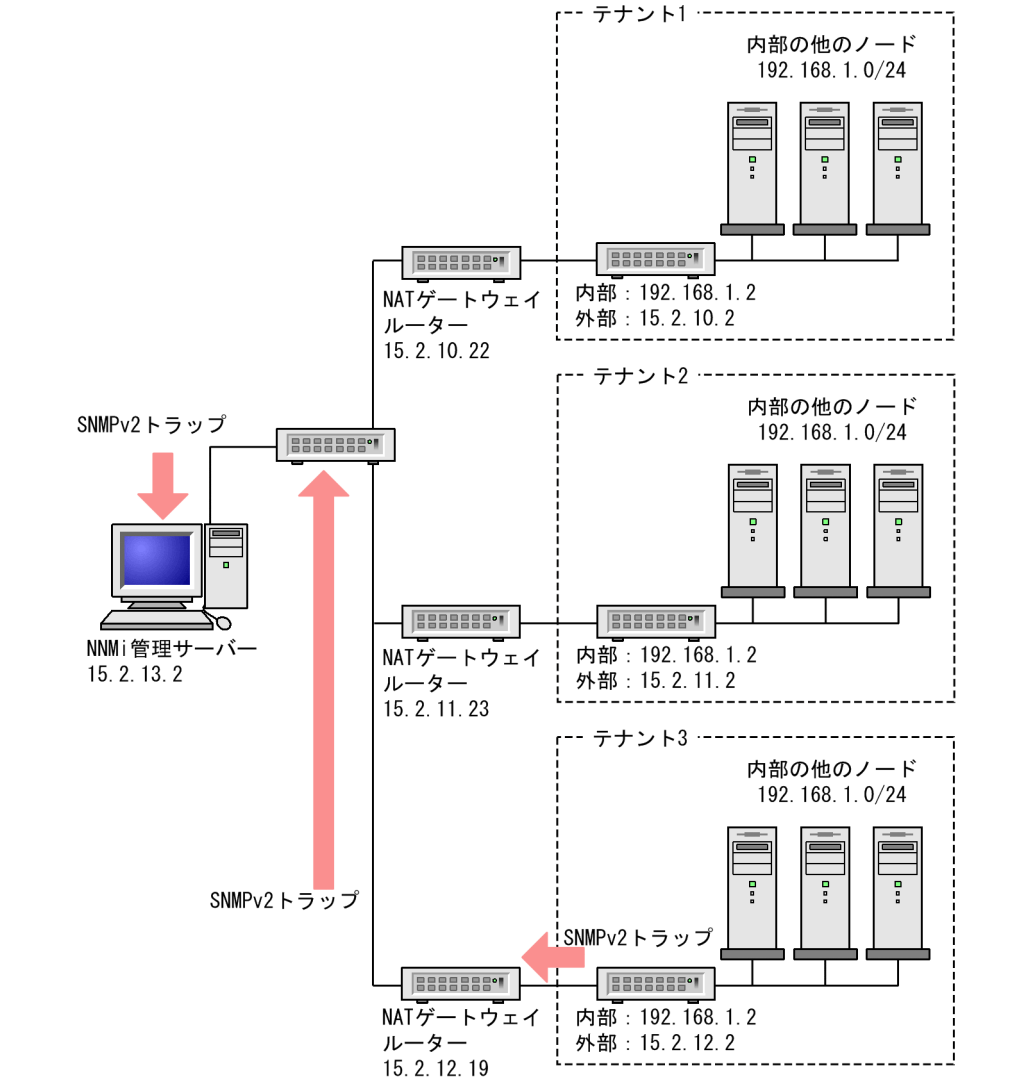
バージョンおよびその他の情報
ソースアドレス
デスティネーションアドレス
PDUタイプ: 7
要求識別子
エラーステータス
エラーインデックス
PDU変数のバインド

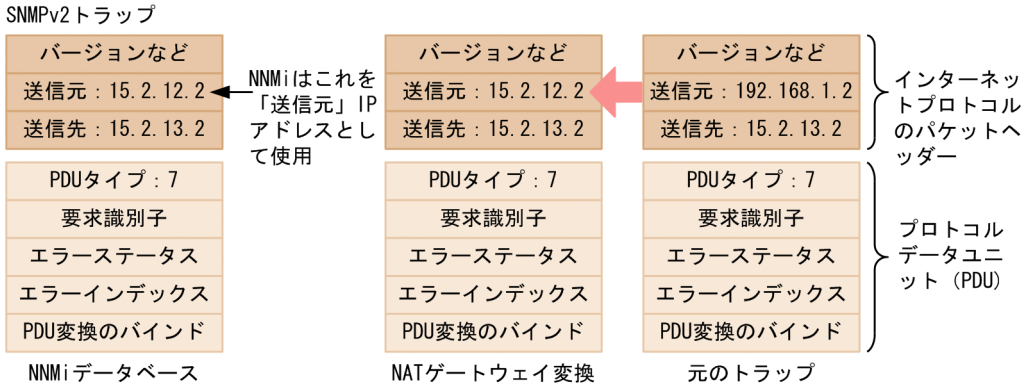
SNMPv2c トラップの PDU には，エージェントアドレスフィールドがありません。そのため，トラップの唯一のソースフィールドが IP パケットヘッダー内に存在します。ソースフィールドは，NAT ルーターによって適切に変換されます。

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 11-2　SNMPv2c の例





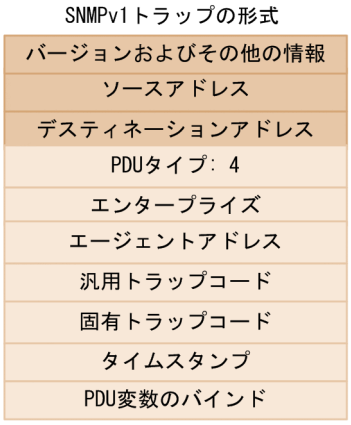
（凡例）

外部：外部アドレス

内部：内部アドレス

(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。



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します。
   - Windows：%NNM_PROPS%¥nms-jboss.properties
   - UNIX：$NNM_PROPS/nms-jboss.properties
2. 次の行を探します。

```
#!com.hp.nnm.trapd.useUdpHeaderIpAddress=false
```

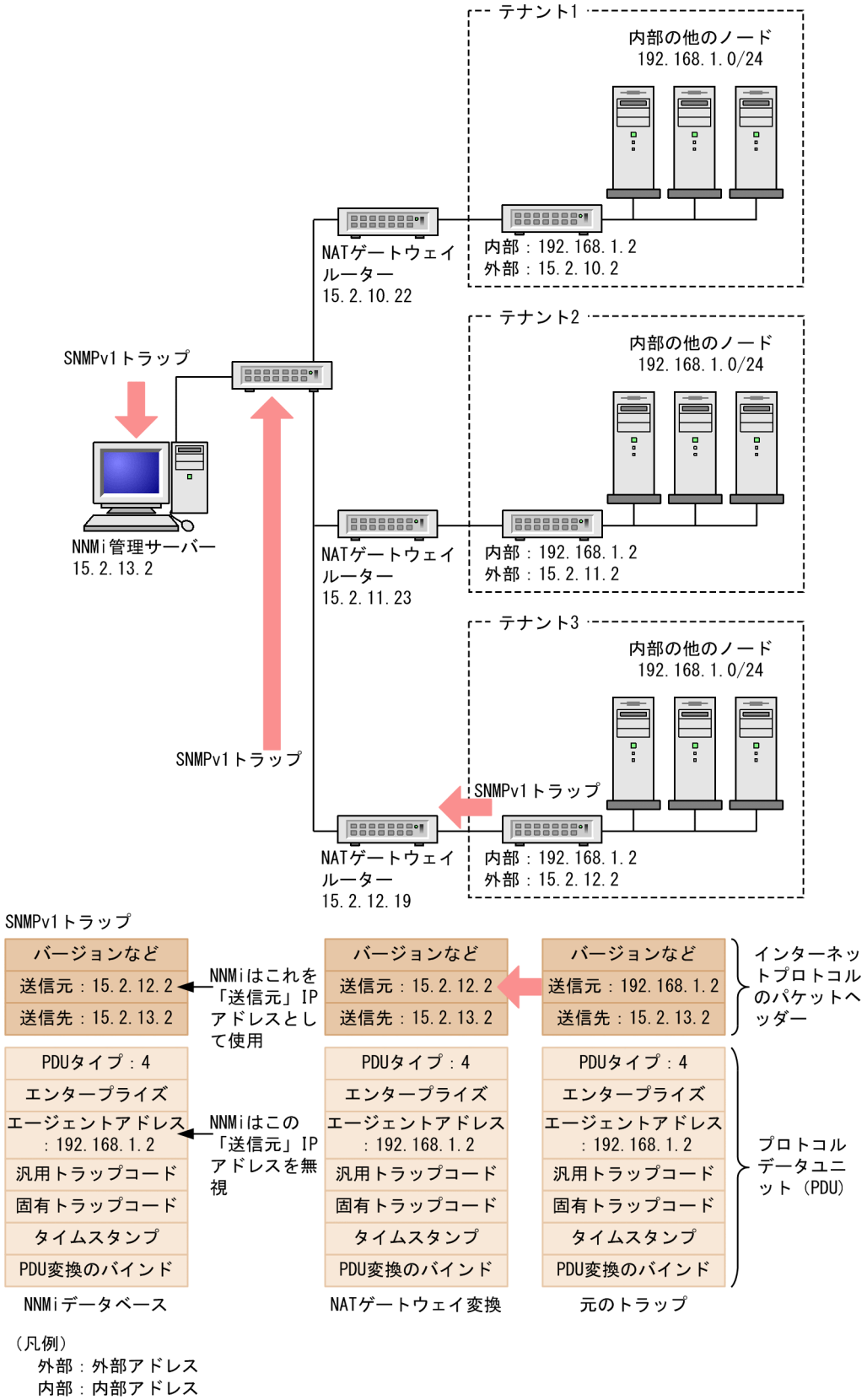
3. 次のように値を true に変更して#!文字を削除します。

```
com.hp.nnm.trapd.useUdpHeaderIpAddress=true
```

4. ファイルを保存して NNMi を再起動します。

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 11-3　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。

11.5.5 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。

11.5.6 グローバルネットワーク管理と静的 NAT

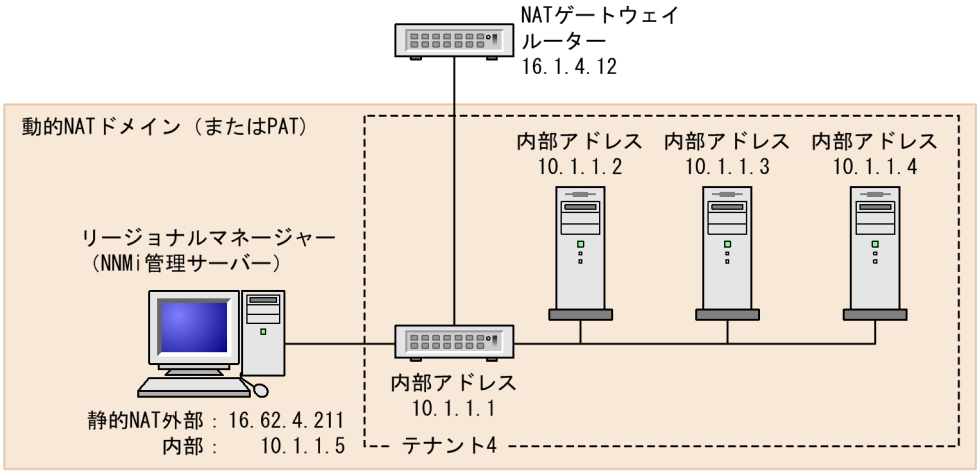
リージョナルマネージャごとに，少なくとも 1 つの静的またはルーティング可能（非変換）アドレスがある必要があります。これによって，NNMi 管理サーバーが相互に通信ができ，通信を隠ぺいしてセキュリティを確保できます。グローバルネットワーク管理の詳細については，「13. グローバルネットワーク管理」を参照してください。

11.6　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。

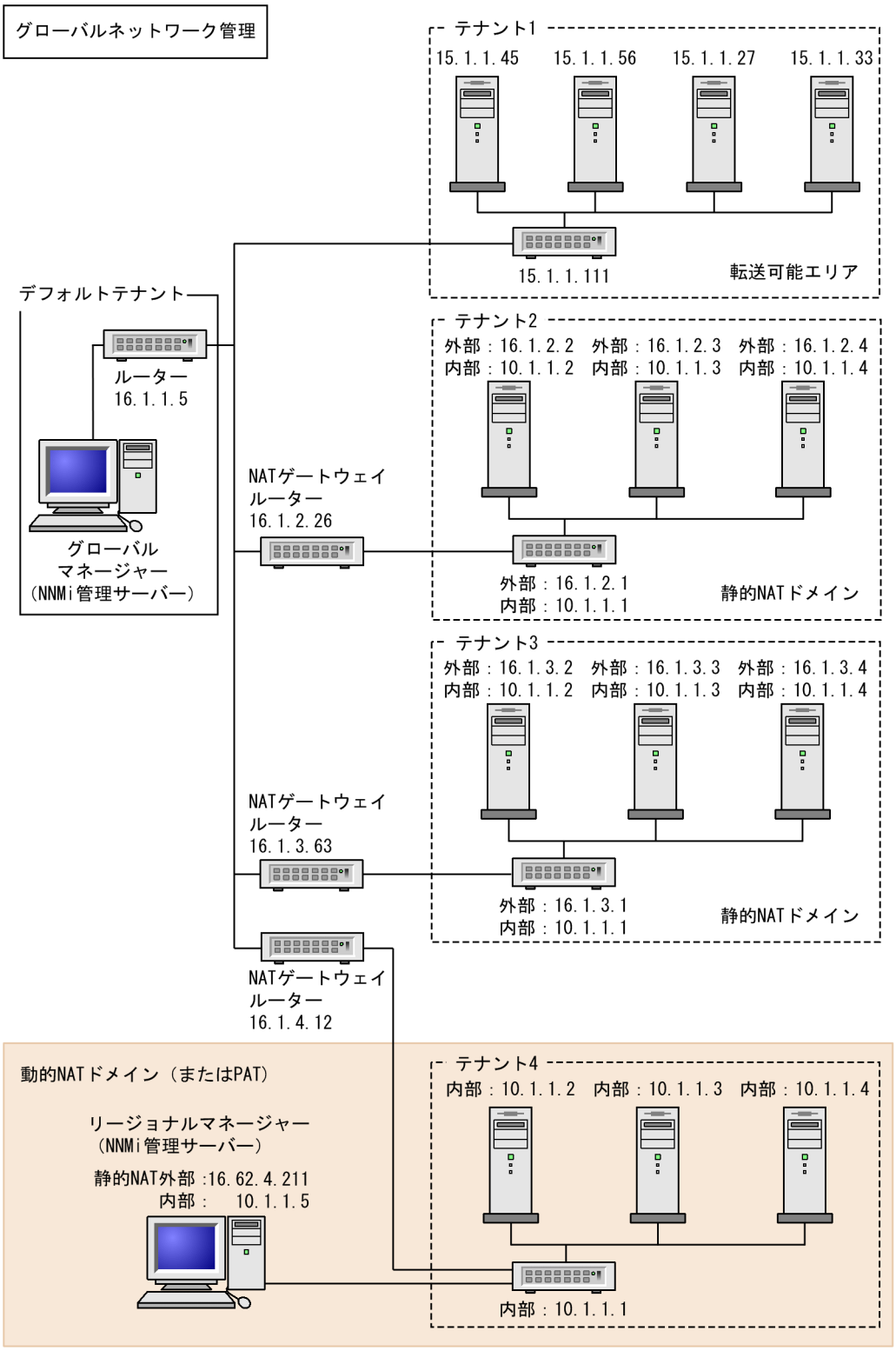
リージョナルマネージャが NAT ファイアウォールの背後にある場合，その外部（パブリック）アドレスは静的アドレスである必要があります。

図 11-4　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 11-5　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，「13. 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。

11.6.1 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が必要です。

11.6.2 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` コマンドまたはグラフィカルユーザーインタフェースを使用），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。

11.6.3 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。

11.6.4　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，「13.　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。

11.7　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「12.　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。「11.6　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。

11.7.1　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）

12 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。

12.1 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。
- 非デフォルトの OAD（Overlapping Address Domain）テナントに所属するノードを含むパスビューは，サポート対象外です。

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。

12.2 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する。
- 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 オペレータに提供する。
- 導入時にノードをステージングする。NNMi 管理者にはすべてのノードが表示されるが，NNMi オペレータには導入したノードだけが表示される。
- 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する。
- 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する。

12.2.1 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 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

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。

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。

すべてのノードコンポーネントは，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。

12.2.2　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 12-1 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 12-2 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 12–1　ユーザーアクセス要件に対応するトポロジの例

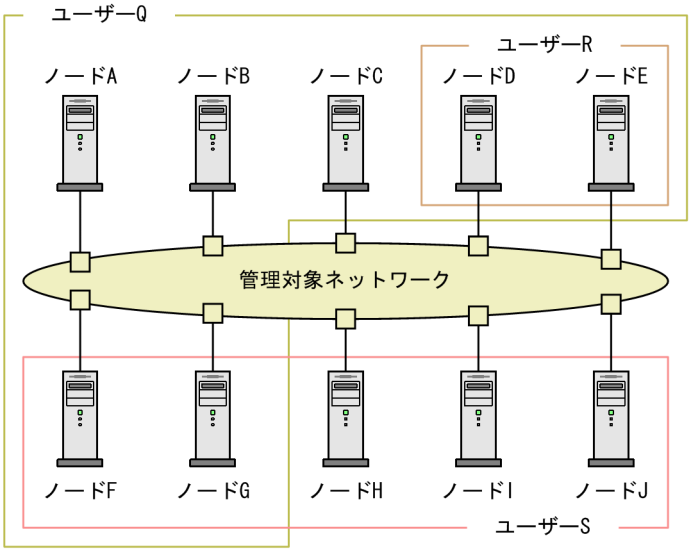


表 12–1　セキュリティグループマッピングの例

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

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
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 12–2　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 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
	UG1 ゲスト	A，B，C	
	UG2 管理者	D，E	
	UG3 レベル 2	F，G	

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
	UG4 レベル 1	H，I，J	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。

12.3 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。

12.3.1 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。

すべてのノードコンポーネントは，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。

12.3.2 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 12-3 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 12-4 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 12–2 複数のテナントのトポロジの例

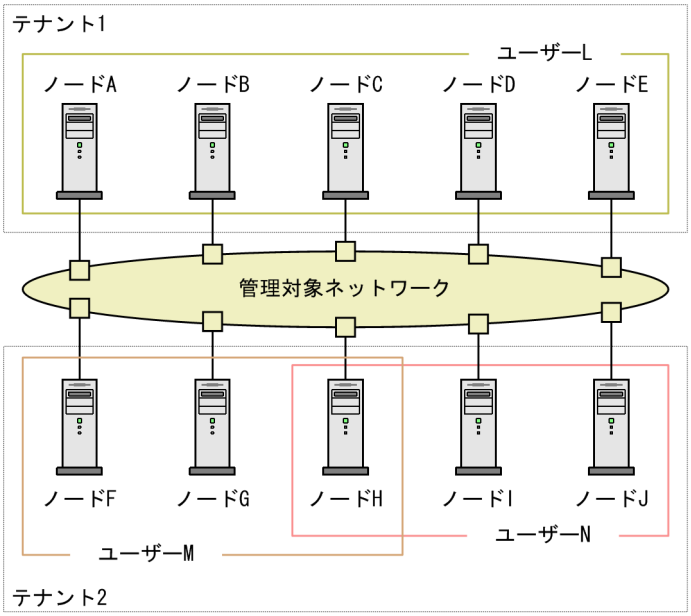


表 12–3 複数のテナントのセキュリティグループマッピングの例

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

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 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 12–4　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	

12.4 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。

ポイント

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，「10.　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。

12.4.1　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。

> **参考**
>
> ［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。

ポイント

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
secgroup1 -role level1
nnmsecurity.ovpl -assignUserGroupToSecurityGroup -userGroup local2 -securityGroup
secgroup2 -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
```

12.4.2　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，「12.4.3　セキュリティグループを設定する」の手順 4. を参照してください。

12.4.3　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。

> **参考**
>
> 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. 「12.4.4　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。

12.4.4 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。

> **参考**
>
> 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ピングを確認します。
- `-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。

12.4.5　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 12-5　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。

12.5 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とリージョナルマネージャで同じです。

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とリージョナルマネージャ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とリージョナルマネージャ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。

ポイント

- グローバルマネージャ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をセットアップする場合は，リージョナルマネージャ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に一意のテナントを設定し，グローバルマネージャ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からグローバルマネージャ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

12.5.1 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。

ポイント

グローバルマネージャ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とグローバルマネージャ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によって 1 つのグローバルマネージャが更新されている場合，そのグローバルマネージャ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上で，リージョナルマネージャ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。

12.5.2 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にどのように影響を及ぼすかを示しています。

表 12-6 リージョナルマネージャでの設定変更がグローバルマネージャに及ぼす影響

アクション	影響
リージョナルマネージャで，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にない場合は作成されます。
リージョナルマネージャで，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。

アクション	影響
リージョナルマネージャで，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。

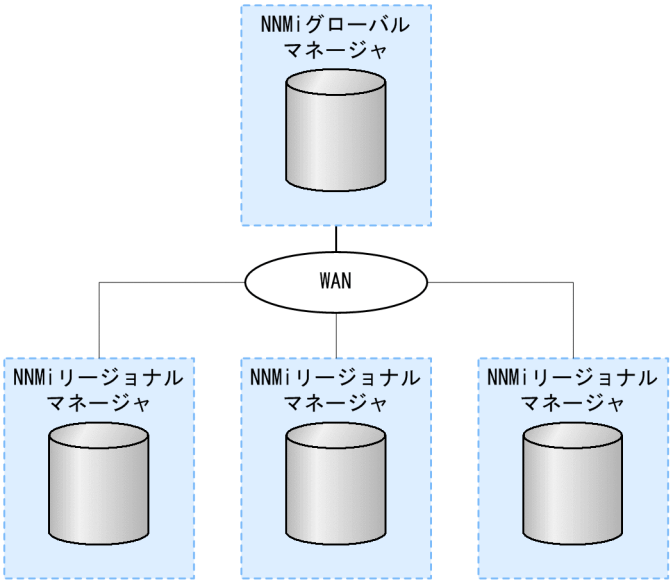
13 グローバルネットワーク管理

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。

13.1　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。

13.2 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として指定し，複数のリージョナルマネージャ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が必要です。詳細については，「11. 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。

13.3 グローバルネットワーク管理の適用を検討する

13.3.1 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。

13.3.2 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からグローバルマネージャ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に転送するようにできます。

13.3.3 ライセンスを考慮する

グローバルマネージャとして使用する 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に転送フィルタを設定して，グローバルマネージャ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用に NNMi ライセンスを増やします。グローバルマネージャには，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をすべてのリージョナルマネージャと同期させて，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に送信します。

グローバルマネージャ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で`nnmnoderediscover.ovpl -all` スクリプトを実行します。

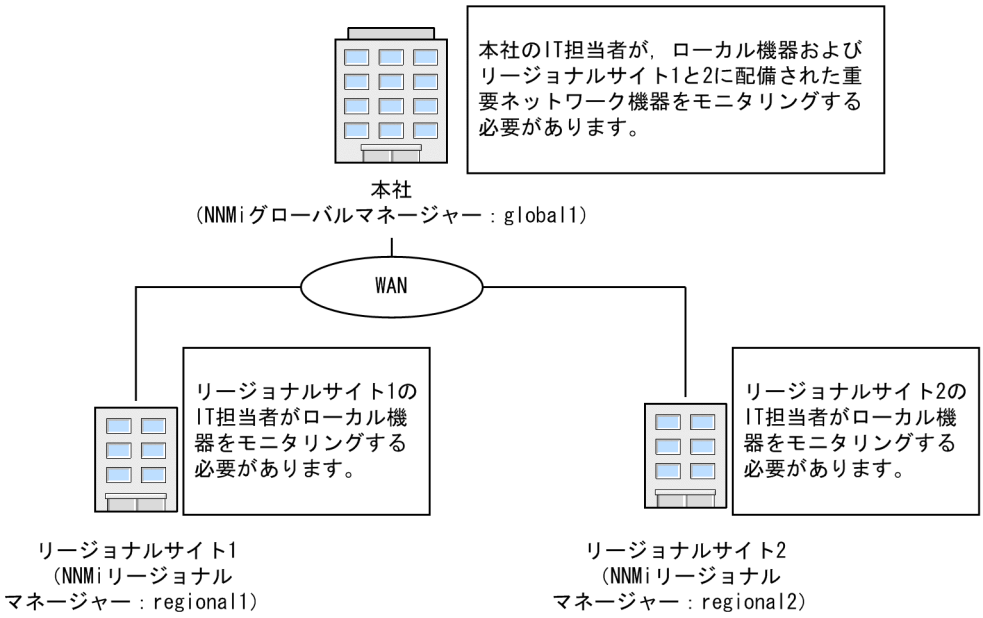
> **参考**
>
> 2 番目の方法では，ネットワーク上のトラフィックが増加し，NNMi マネージャのセット全体から多くの NNMi リソースが消費されることにもなります。このオプションは，最初の NNMi 検出ほどリソースの多くを消費しませんが，最初の検出を実行することに似ています。最適な方法では，ある程度の時間をおくか，現在のリージョナルマネージャの負荷が減って正常になるのを待ち，領域ごとに間隔をおいてスクリプトを実行してから，次のリージョナルマネージャの再検出を始めます。

13.4　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 13–1　ネットワークの例



13.4.1　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。

参考

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後，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 13–1　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 Model 3500yl HP Procurve Switch	各リージョナルサイトの Model 3500yl HP ProCurve Switch すべて
リージョナルサイト 1	regional1	15 台の Model 3500yl HP Procurve Switch	該当なし
リージョナルサイト 2	regional2	15 台の Model 3500yl HP Procurve 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 Model 3500yl ProCurve 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とグローバルマネージャの接続

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と通信する 1 つ以上のグローバルマネージャを設定できます。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，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。

13.4.2　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 80 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。

参考

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`
- UNIX：`$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 13-2　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	TCP ポート
非 SSL	nmsas.server.port.web.http	80
	nmsas.server.port.hq	4457
SSL	nmsas.server.port.web.https	443
	nmsas.server.port.hq.ssl	4459

(2)　自己署名証明書の設定

global1 と 2 つのリージョナル NNMi 管理サーバー（regional1 と regional2）間で SSL（Secure Sockets Layer）を使用してグローバルネットワーク管理機能を使用する場合は，追加の作業が必要です。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。「8.6　自己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する」に示した手順を実行してください。

(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。グローバルネットワーク管理機能は，以前の NNM 製品で使用されていた分散ソリューションとは異なります。グローバルネットワーク管理機能を使用すると，リージョナルシステムによるポーリングノードの管理が回避されるため，ネットワーク帯域幅やコンピュータリソースを考慮する必要がなくなります。

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，マニュアル「*JP1/Cm2/Network Node Manager i　インストールガイド*」，「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とリージョナルマネージャ）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，UNIX（HP-UX/Linux/Solaris）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「*トラブルシューティンググローバルネットワーク管理*」と「13.11.2　クロック同期」を参照してください。

> **参考**
>
> 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。「8.6　自己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する」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。「8.7　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」を参照してください。

(8) 監視する重要な機器の一覧作成

global1 から監視する，regional1 と regional2 の管理機器一覧を作成します。この情報を転送フィルタ（これについてはあとで説明します）で使用します。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 のライセ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とリージョナルマネージャの管理ドメインの検討

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。

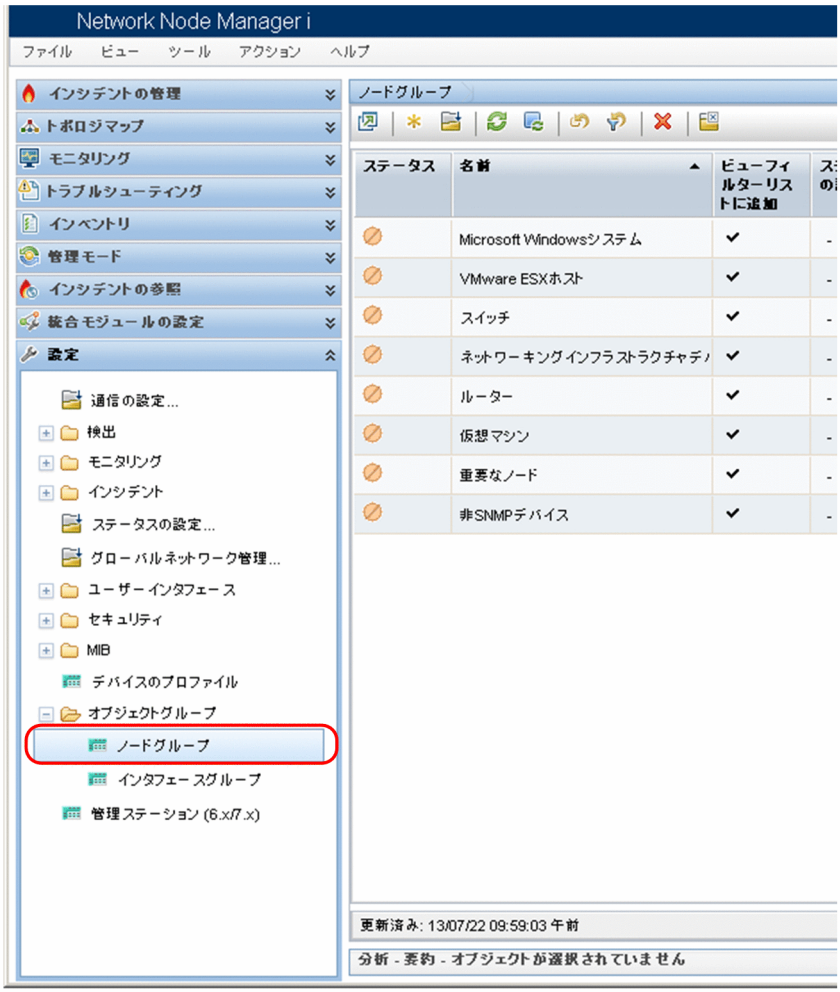
13.5　リージョナルマネージャで転送フィルタを設定する

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 global1 がリージョナルマネージャ 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。

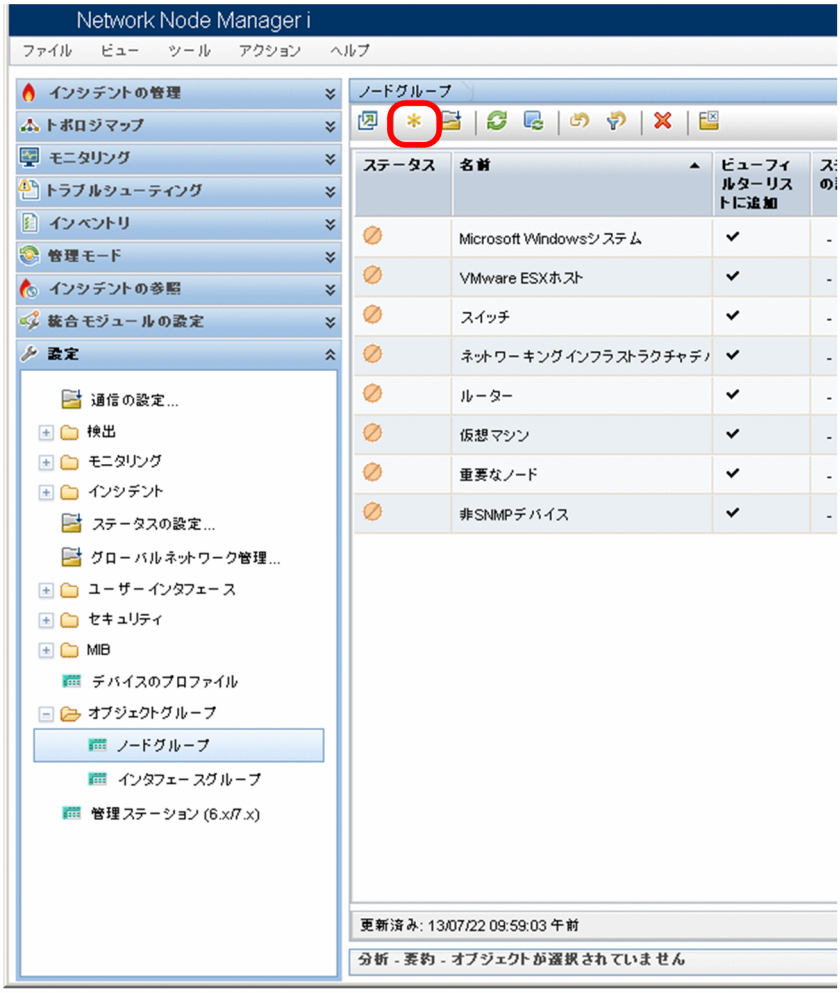
13.5.1　転送されるノードを制限する転送フィルタを設定する

ノードグループを設定し，regional1 から Model 3500yl ProCurve Switch のノード情報だけを global1 に転送するようにします。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。

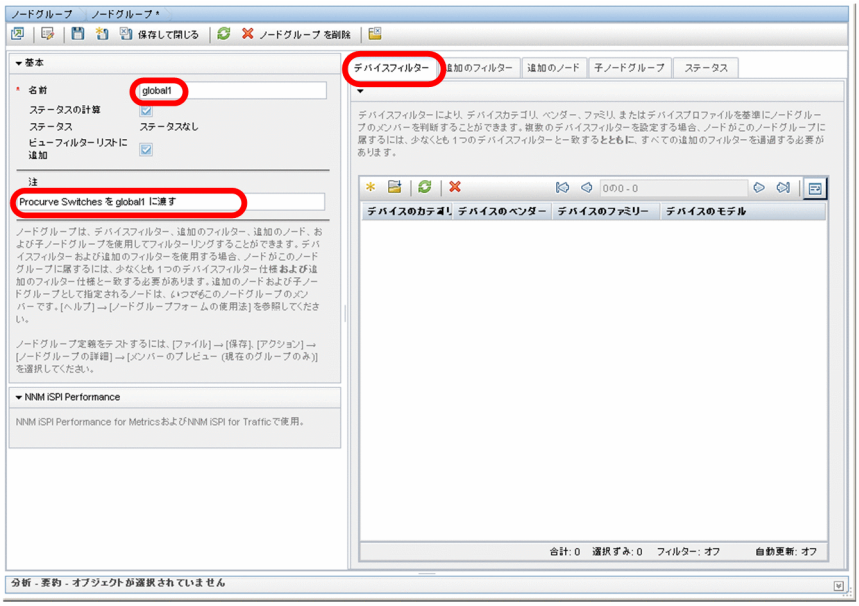


参考

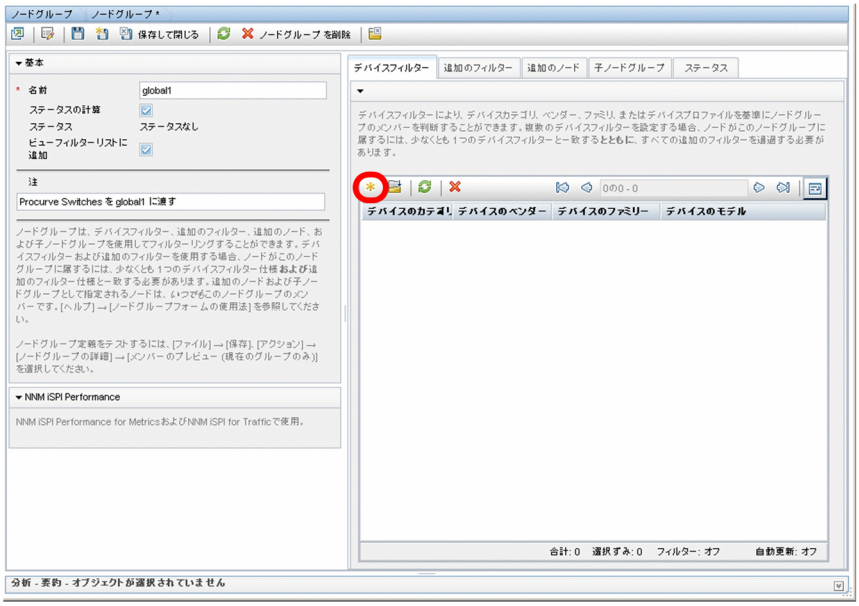
この例では，ノードフィルタの新規作成し，そのフィルタを使用して regional1 と regional2 の転送フィルタを作成する方法を説明していますが，既存のフィルタを使用して，リージョナル NNMi 管理サーバーからグローバル NNMi 管理サーバーへの転送フィルタを設定することもできます。

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。

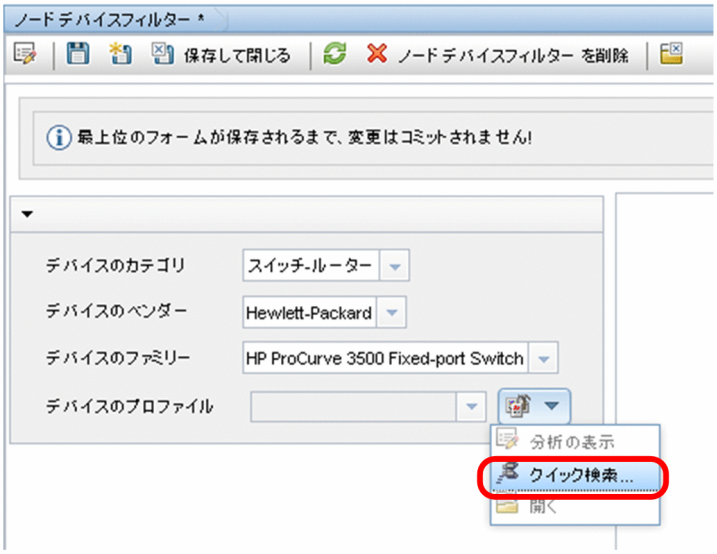


4. [デバイスフィルター] タブで [新規作成] アイコンをクリックして，[ノードデバイスフィルター] フォームを開く。

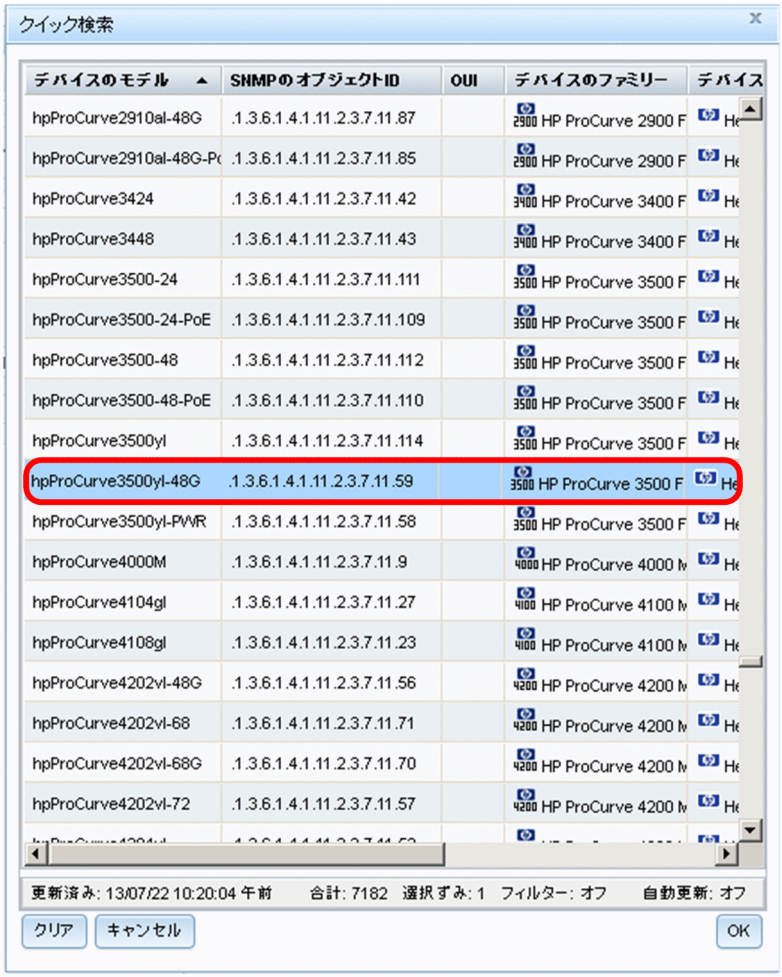


5. プルダウンメニューを使用して，[デバイスのカテゴリ] では [スイッチ-ルーター]，[デバイスのベンダー] では [Hewlett-Packard]，および [デバイスのファミリー] では [HP Procurve 3500 Fixed-port Switch] を選択する。

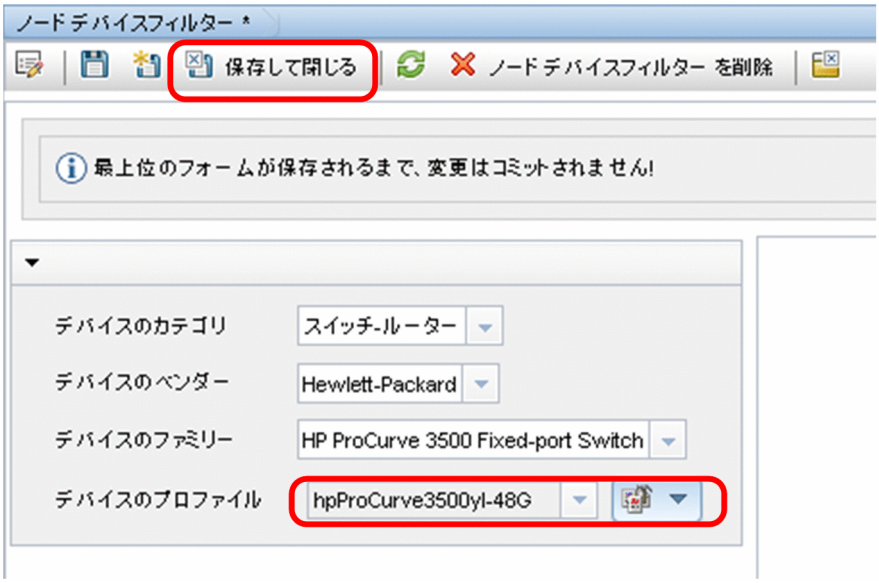
6. プルダウンメニューから，[クイック検索] をクリックして，[デバイスのプロファイル] フォームを開く。



7. 3500yl HP ProCurve Switch のプロファイルを検索して選択し，［OK］をクリックする。

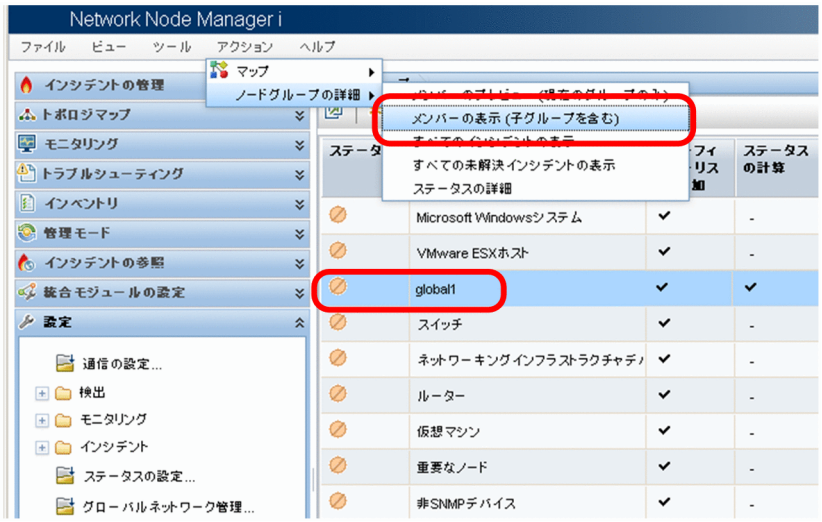


8. ［保存して閉じる］を 2 回クリックする。

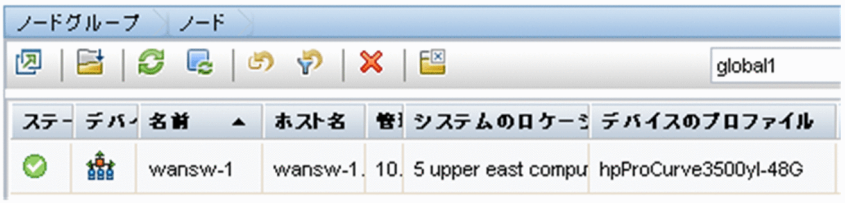


9. このフィルタをテストするため，［global1］を選択する。

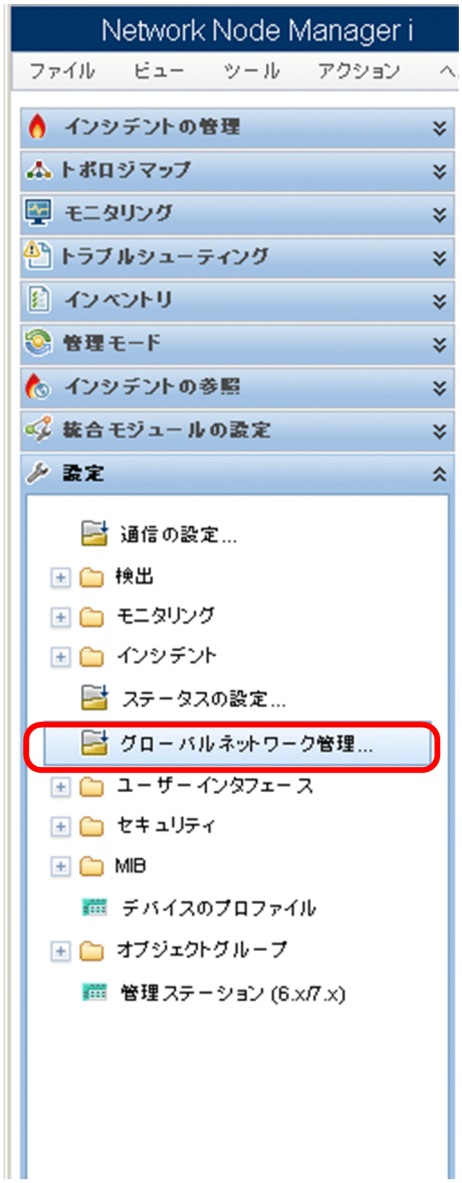
10. ［アクション］＞［ノードグループの詳細］メニューから，［メンバーの表示］をクリックする。



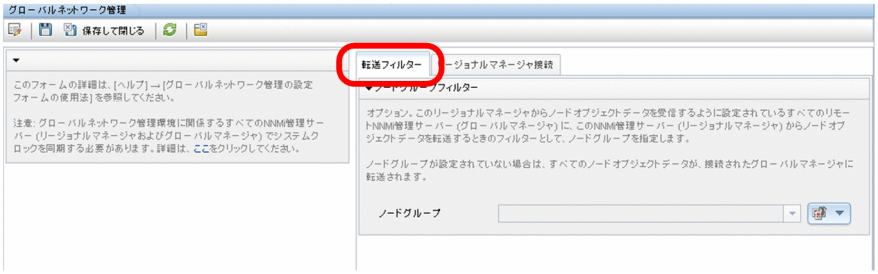
11. NNMi ではすでに HP 3500yl スイッチが 1 つ検出されている。これは，作成したフィルタが，設定した特定のスイッチモデルを検索していることを示している。次のステップでは，今作成したこのノードフィルタを使用して転送フィルタを設定する。



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。

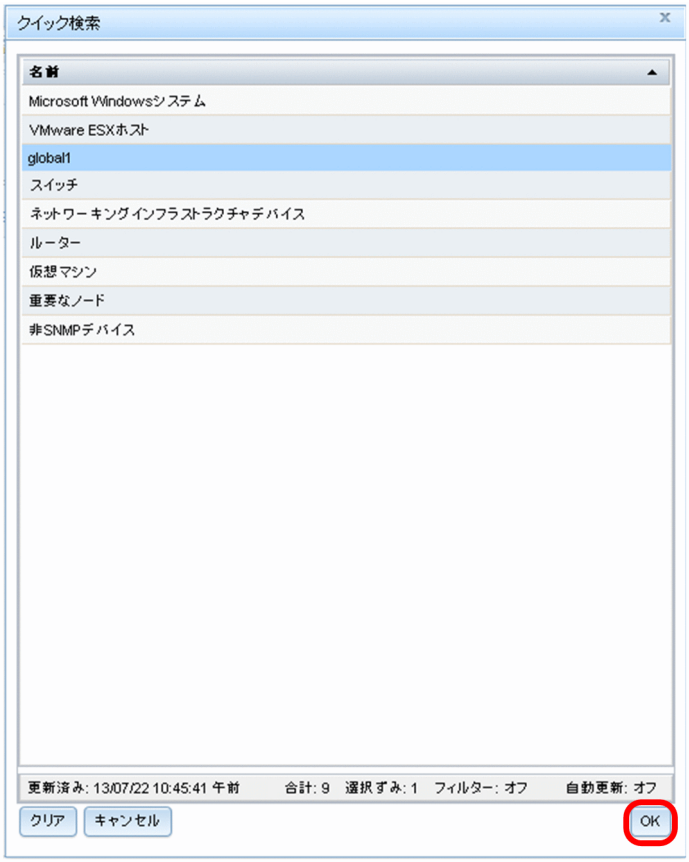


13. [転送フィルター] タブをクリックする。



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。

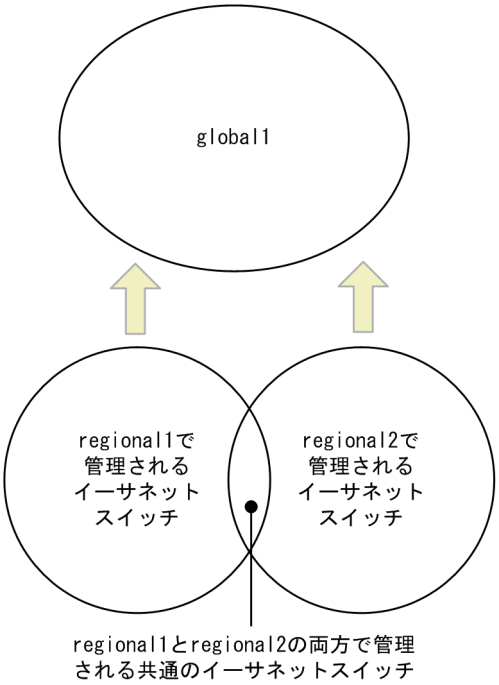


16. [保存して閉じる] をクリックする。

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.を実行したら，次のセクションに進み，global1 を regional1 と regional2 に接続します。

13.6　グローバルマネージャとリージョナルマネージャを接続する

すでに述べたように，regional1 と regional2 の両方で，共通のスイッチを複数管理しているとします。この共通のスイッチ情報を regional1 から global1 に転送します。



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。

> **参考**
>
> 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。

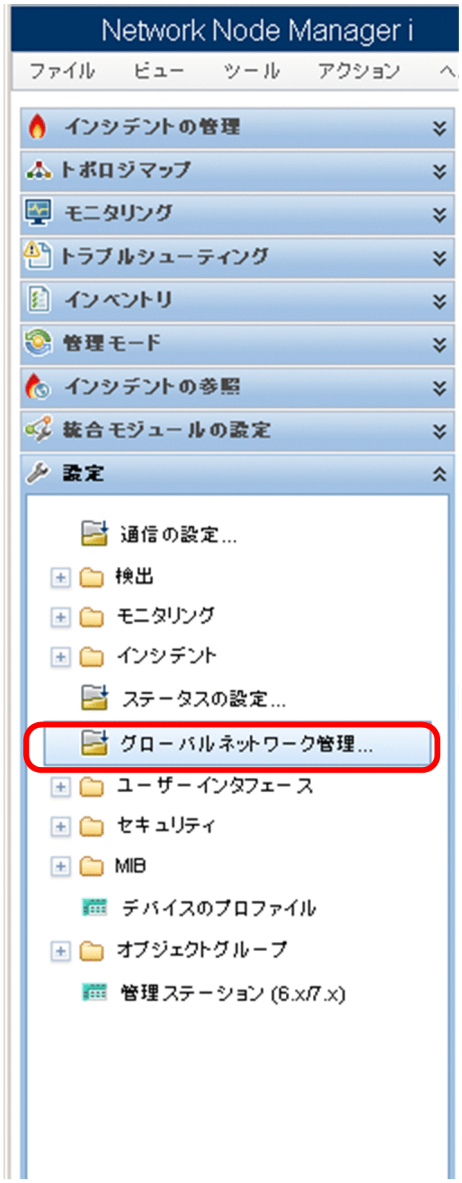
1. **すでに述べたように，NNMi 管理サーバーのクロックを global1，regional1，および regional2 と同期化してから，グローバルネットワーク管理設定内のこれらのサーバーを接続する。**

   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。

   > **参考**
   >
   > 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。

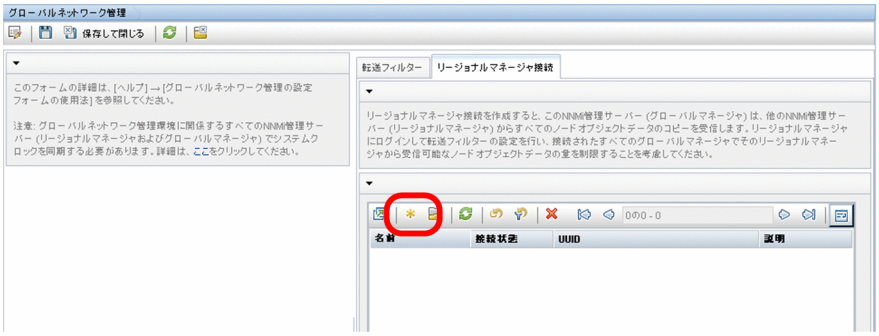
2. global1 から regional1 への接続を設定する。

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。

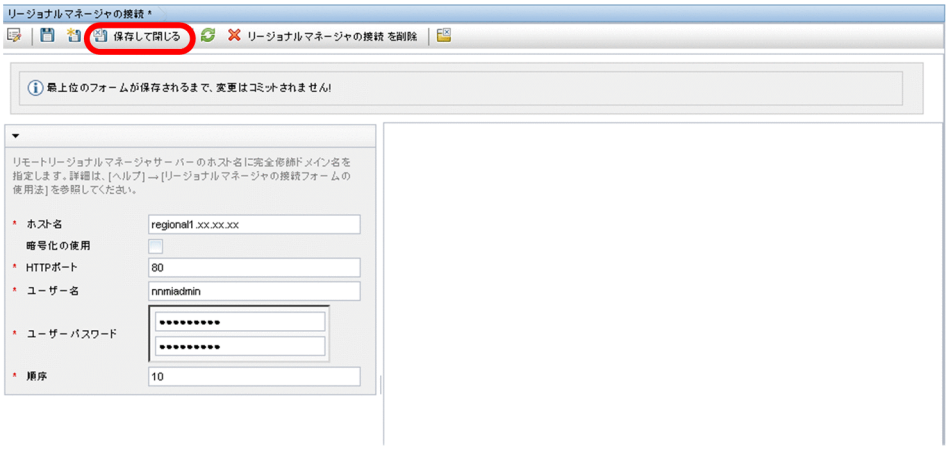
e　［接続］タブをクリックします。

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。

> **参考**
>
> このフォームで作成するエントリ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　［保存して閉じる］を 2 回クリックして作業を保存します。

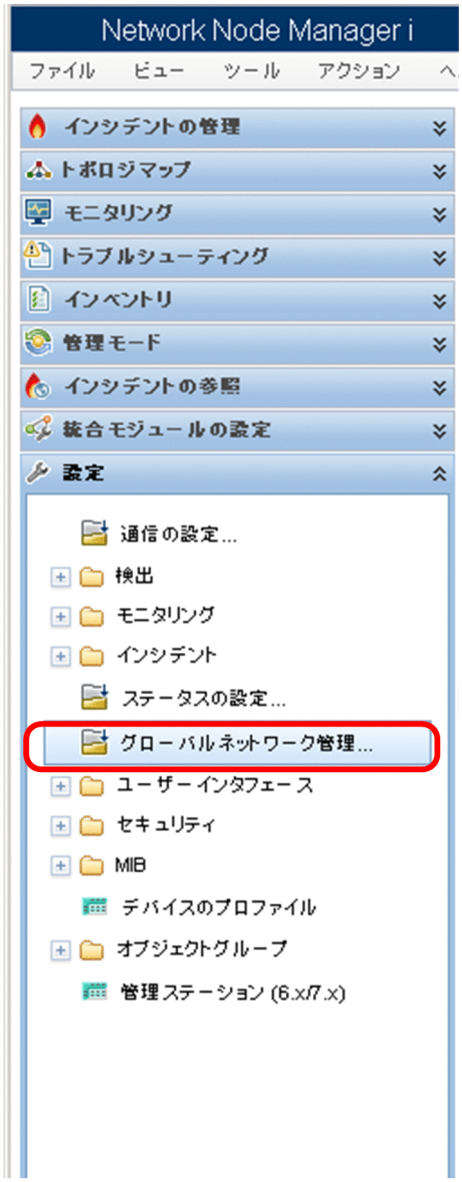


3. global1 から regional2 への接続を確立するため，手順 2.の a から h までを実行する。

13.7　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。

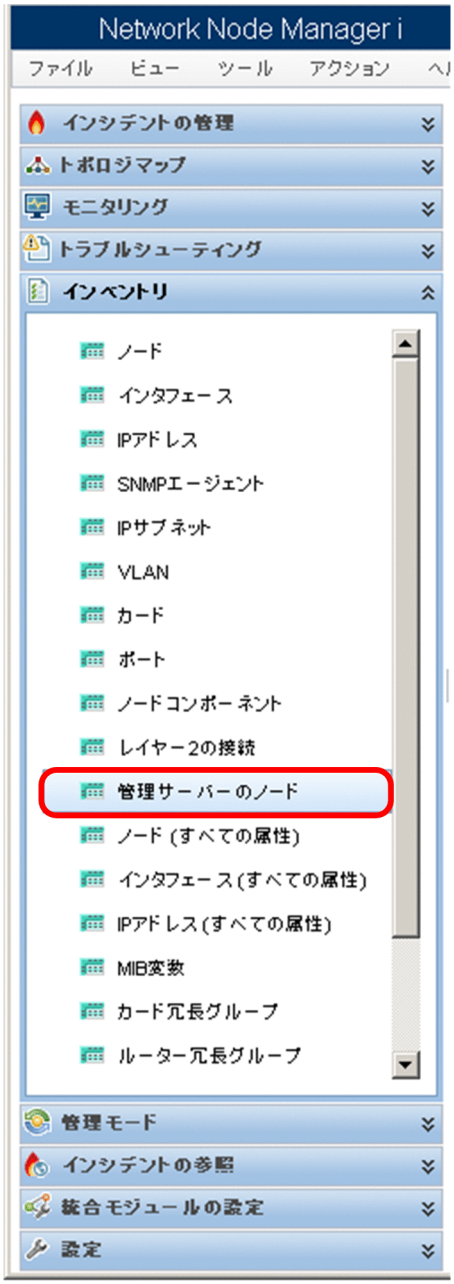
NNMi が検出を完了するまで，次のセクションには進まないでください。詳細については，マニュアル「*JP1/Cm2/Network Node Manager i　インストールガイド*」の「*4.3.3　検出の進行状況を確認する*」を参照してください。

13.8　global1 のインベントリを確認する

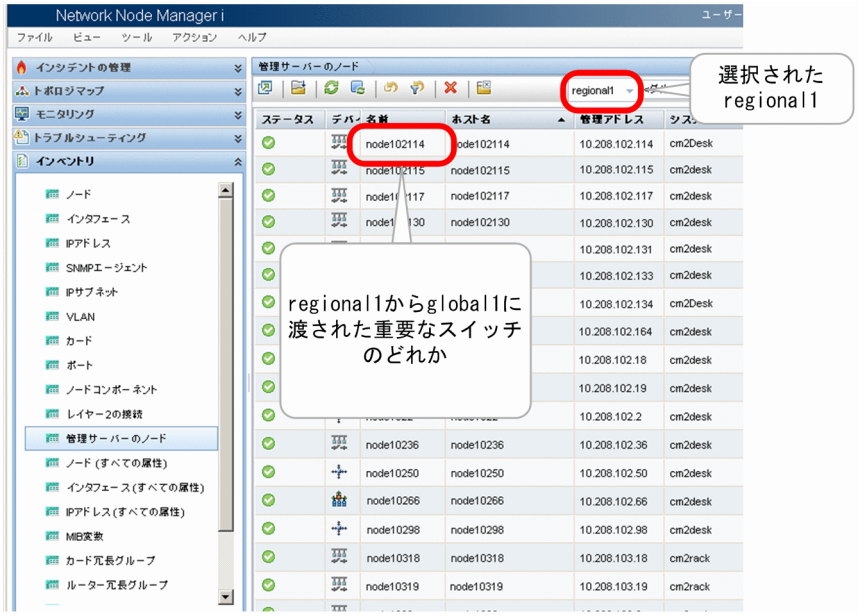
NNMi が検出を完了するまで，このセクションは実行しないでください。詳細については，マニュアル「*JP1/Cm2/Network Node Manager i　インストールガイド*」の「*4.3.3　検出の進行状況を確認する*」を参照してください。

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。

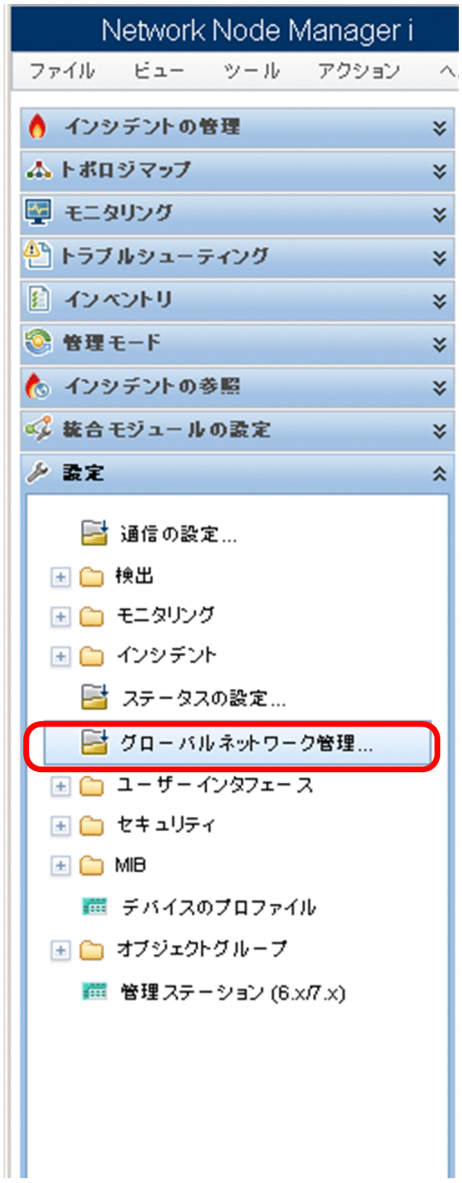


手順 1.から手順 2.を実行して，接続されているほかのリージョナルマネージャから global1 に転送されたデバイスインベントリも表示します。

13.9　global1 と regional1 との通信を切断する

global1 を完全にシャットダウンするか，何日間かシャットダウンする計画であることを想定します。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仮定します。シャットダウンを完了するには，追加の手順を実行する必要があります。

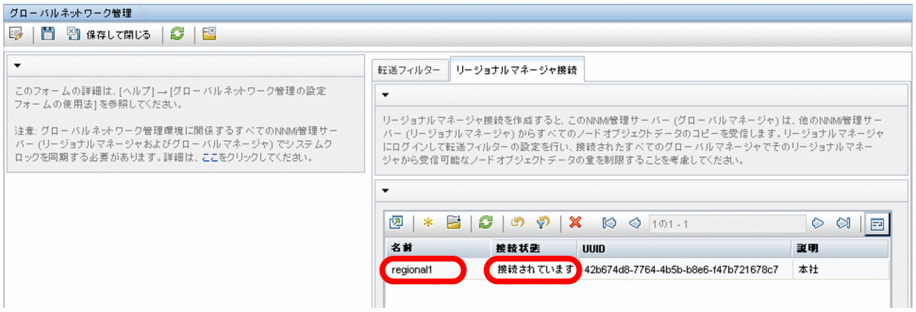
1. global1 の NNMi コンソールで，［設定］ワークスペースの［グローバルネットワーク管理］をクリックする。



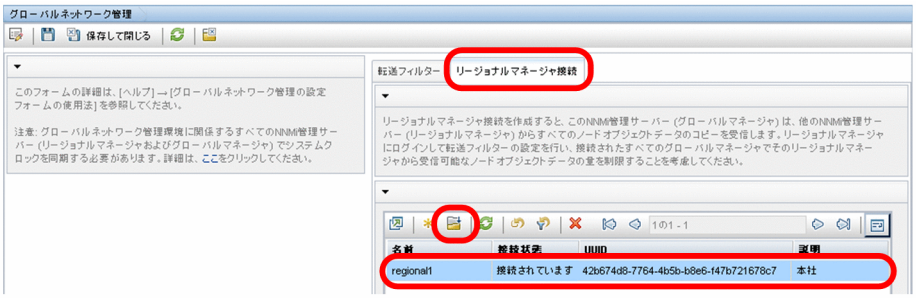
2. ［リージョナルマネージャ接続］をクリックする。

3. 接続状態が［接続されています］であることを確認する。

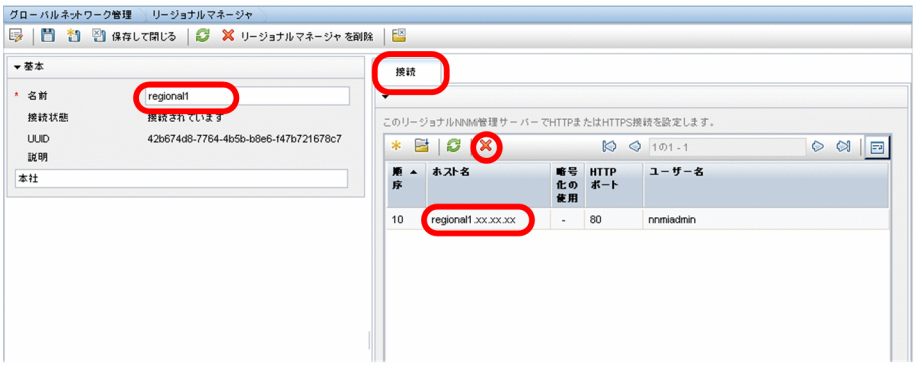
接続状態が［接続されています］ではない場合，処理を続行する前に，NNMi ヘルプの「*トラブルシューティンググローバルネットワーク管理*」を参照して問題を診断します。



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］をもう一度クリックする。

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。

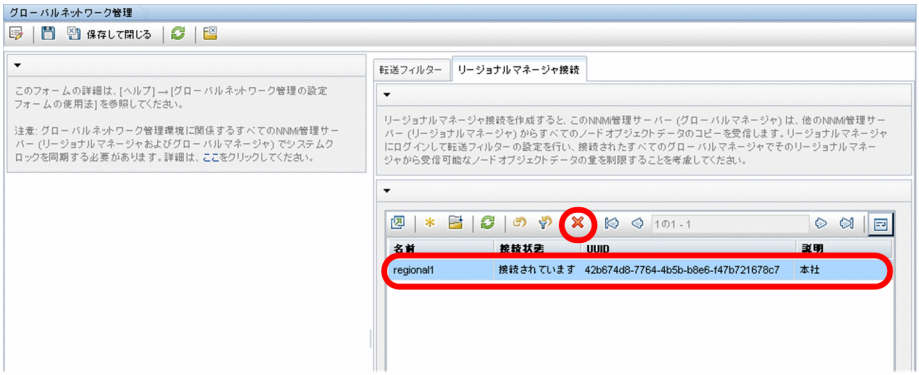
11. regional1 の設定レコードを削除するには，次を実行する。

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。

13.10　グローバルネットワーク管理の追加情報

13.10.1　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。

参考

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。

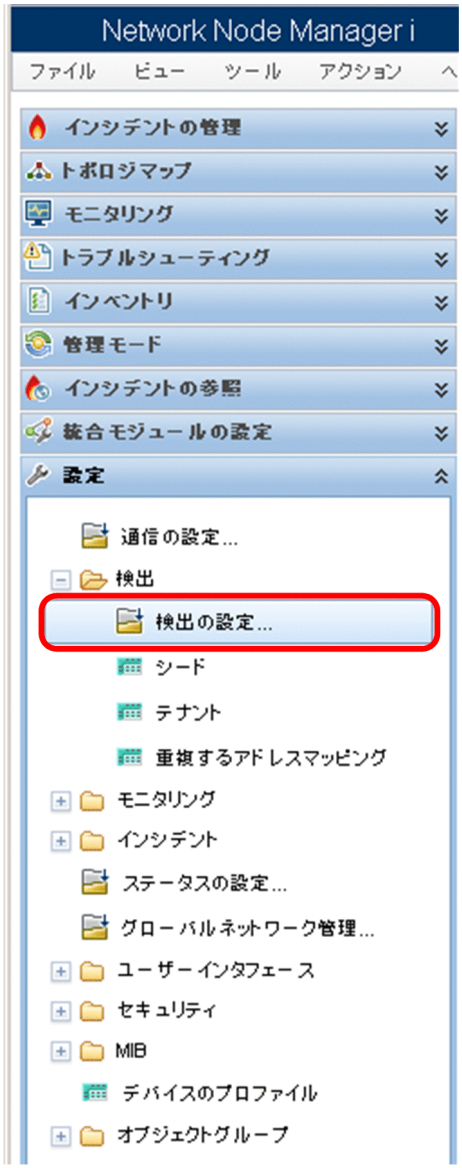
global1 は，regional1 または regional2 が検出を完了するたびに，regional1 と regional2 を同期します。NNMi は FDB（転送データベース）データを使用して，レイヤー 2 接続を計算します。FDB データは非常にダイナミックなもので，特に，1 つのグローバルサーバーに複数のリージョナルサーバーが接続しているような場合には，検出するごとに大きく異なります。

参考

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。

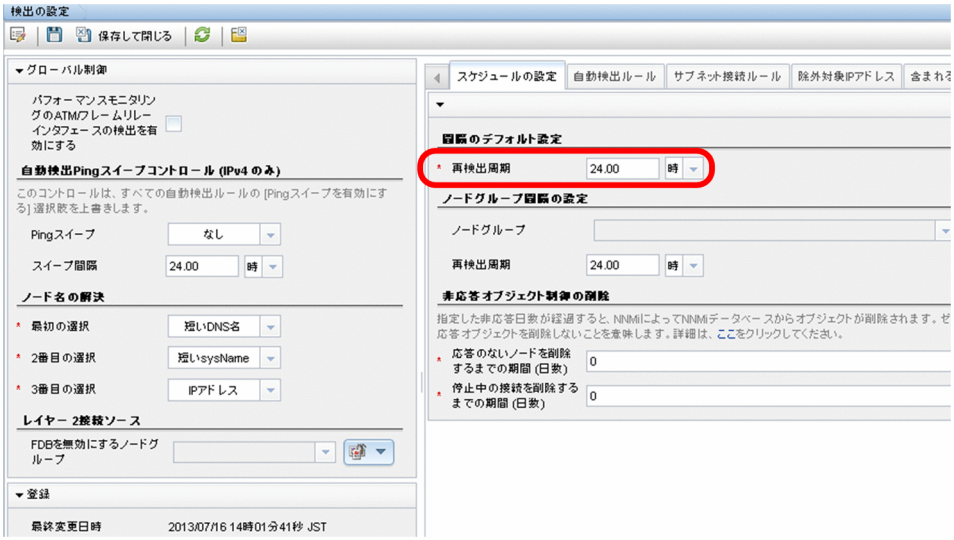
[再検出間隔] は，各リージョナルサーバーで調整でき，global1 とリージョナルマネージャとの間の検出の精度を変更できます。**[再検出間隔]** が短くなるほど，検出の精度が上がり，NNMi が行うネットワークトラフィックも増えます。**[再検出間隔]** が長くなるほど，検出の精度は下がり，NNMi が行うネットワークトラフィックも減ります。これは，ネットワークが大きくなるほど，ユーザーが行う再検出の頻度が少なくなることを意味します。**[再検出間隔]** を設定するには，次の手順を実行します。

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，［再検出周期］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。

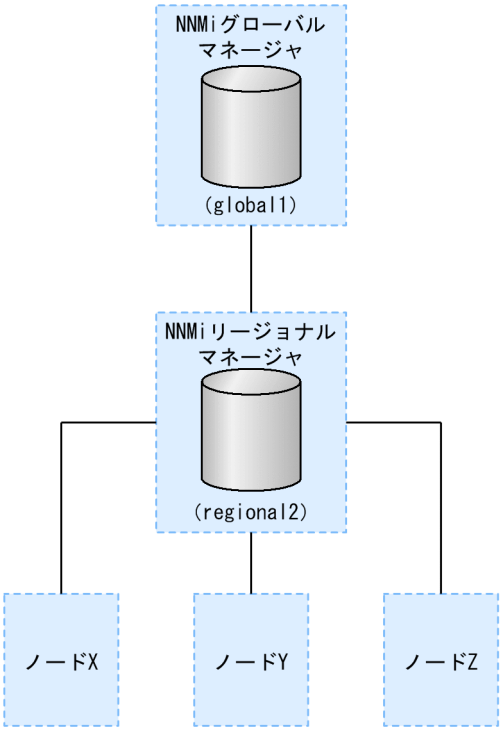


3. [保存して閉じる] をクリックする。

13.10.2 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

リージョナル NNMi 管理サーバー regional2 が Node X を検出して管理し，グローバル NNMi 管理サーバー global1 がリージョナル NNMi 管理サーバー regional2 に接続すると想定します。

図 13–2 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。
5. 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はない。
   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。

13.10.3 グローバルマネージャ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 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。

13.11　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

13.11.1　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「*トラブルシューティンググローバルネットワーク管理*」を参照してください。

13.11.2　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とリージョナルマネージャ）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，UNIX（HP-UX/Linux/Solaris）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 ’<name>’ への接続は停止しています。

13.11.3　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。

13.11.4　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

global1 と regional2 の間で情報に矛盾があることに気が付いたと想定します。それを解決するため，global1 から nnmnoderediscover.ovpl スクリプトを実行し，global1 と regional2 を同期化します。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。

「図 13-2　ノードのステータスポーリングまたは設定ポーリング」に示したネットワークについて考えます。regional2 をノード X，Y，および Z とそのノードセット全体を global1 を使用して同期化すると想定します。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。

13.12　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「22.3　NNMi 10-00 および 10-10 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」を参照してください。

13.13　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が必要です。「11.　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。

14 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。

14.1 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。

- IPv6 サブネット接続の検出
- 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

14.2 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 は，Windows オペレーティングシステムではサポートされていません。IPv6 をサポートするオペレーティングシステムの詳細については，「リリースノート」を参照してください。そのほかに，次の要件があります。

- 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。

> **参考**
>
> 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。

14.3 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，マニュアル「*JP1/Cm2/Network Node Manager i インストールガイド*」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。

14.4　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。

14.4.1　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 14–1　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

14.4.2　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

14.5 NNMi のインストールと IPv6 管理機能の有効化

NNMi のインストールでは，インストールスクリプトに IPv6 機能が含まれますが，これらの IPv6 機能は手動で有効化する必要があります。IPv6 機能を有効化するには，まず NNMi Advanced ライセンスを購入して適用する必要があります。次に，nms-jboss.properties ファイルを編集して，IPv6 が機能するよう手動で設定する必要があります。

14.6 IPv6 管理機能を有効にする

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）が設定されている必要があります。

次に示す手順は，IPv6 機能を有効にする方法を説明しています。

- NNMi Advanced ライセンスのインストール
- nms-jboss.properties ファイルにある IPv6 マスタースイッチの有効化

先に進む前に，前のセクションで説明した必要条件すべてについてレビューと確認を行います。

1. **NNMi に同梱されたインスタントオンライセンスを使用，または NNMi Advanced ライセンスをインストールする。**

   NNMi ライセンスの取得とインストールの詳細については，マニュアル「JP1/Cm2/Network Node Manager i インストールガイド」を参照してください。IPv6 機能は，基本 NNMi ライセンスでは使用できません。

2. **nms-jboss.properties ファイルを編集する。**

   次の場所を探してください。

   - UNIX：$NNM_PROPS/nms-jboss.properties

3. **# Enable NNMi IPv6 Management で始まるテキストを探す。**

   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。

   a NNMi で IPv6 通信を有効化するには，次のプロパティのコメントを解除します。

   ```
   java.net.preferIPv4Stack=false
   ```

   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 NNMi で IPv6 通信全体を有効化するには，次のプロパティのコメントを解除します。

   ```
   com.hp.nnm.enableIPv6Mgmt=true
   ```

   c nms-jboss.properties ファイルを保存して閉じます。

4. **NNMi 管理サーバーを再起動する。**

   a NNMi 管理サーバーでovstop コマンドを実行します。

   b NNMi 管理サーバーでovstart コマンドを実行します。

5. **任意で，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 どちらかを使用して通信できるノードです。これには，次の手順を実行します。

a　NNMi コンソールで，［設定］ワークスペースの［通信の設定］をクリックします。

b　［IP バージョン設定］フィールドで，［IPv4］，［IPv6］または［Any］を選択します。

c　変更を保存します。

6. 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。

```
オブジェクトマネージャ名: ovjboss
 状態:                実行中
 PID:                 <Process ID #>
 最後のメッセージ:        Initialization complete.
 終了ステータス:         -
 追加情報:
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
NmsApa                                  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
NmsTrapReceiver                         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
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
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
TrapPropertiesService                   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
```

7. IPv6 を有効化すると，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。

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］＞［ポーリング］＞［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。

14.7 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。

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。

14.7.1 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。

14.7.2 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。
- NNMi Advanced ライセンスが期限切れ，または誰かがライセンスを削除した。

  NNMi は，NNMi の基本ライセンスを使用します。基本ライセンスの管理ノードライセンス数は，すべての検出済みノードの管理を続行するのに十分な容量があります。NNMi は SNMP IPv6 でないノードすべてをインベントリからすぐに削除します。NNMi は SNMP ノードをすべて再検出し，IPv6 データはすべて削除します。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。この場合，対象ノードを NNMi インベントリから手動で削除する必要があります。
- NNMi Advanced ライセンスが期限切れ，または誰かがライセンスを削除した。

  NNMi は，NNMi 基本ライセンスを使用します。基本ライセンスの管理ノードライセンス数は，すべての検出済みノードの管理を続行するのに十分な容量がありません。NNMi はすぐに，SNMP IPv6 で

ないノードを削除します。Licensing サービスは，ライセンスを受けたインベントリ能力を超える SNMP ノードに［非管理対象］状態のマークを付けます。NNMi はすぐに，管理対象 SNMP ノードから得た IPv6 データを削除します。

管理対象外の SNMP ノードの場合は，次の手順を実行します。

a　追加ライセンス機能をインストールします。

b　NNMi コンソールの［アクション］＞［管理モード］＞［管理］コマンドを使用して，Licensing サービスによって「非管理対象」とマークされているノードの管理モードを変更します。nnmmanagementmode.ovpl スクリプトを使用して，これらのノードも管理できます。詳細については，nnmmanagementmode.ovpl のリファレンスページを参照してください。

c　NNMi コンソールにある［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，NNMi で検出できるようにします。nnmnoderediscover.ovpl スクリプトを使用して，これらのノードも検出できます。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 Advanced ライセンスの期限が切れた，または誰かがライセンスを削除した。NNMi 基本ライセンスをインストールしなかった。

  NNMi によって直ちに SNMP IPv6 以外のノードがすべて削除され，残りのノードが自動的に管理対象外となります。この状況を解決するには，次の手順を実行します。

  a　有効なライセンスをインストールします。

  b　NNMi コンソールの［アクション］＞［管理モード］＞［管理］コマンドを使用して，Licensing サービスによって「非管理対象」とマークされているノードの管理モードを変更します。nnmmanagementmode.ovpl スクリプトを使用して，これらのノードも管理できます。詳細については，nnmmanagementmode.ovpl のリファレンスページを参照してください。

  c　NNMi コンソールにある［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，NNMi が「非管理対象」から「管理」に変更したノードを検出できるようにします。nnmnoderediscover.ovpl スクリプトを使用して，これらのノードも検出できます。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。

  d　IPv6 リストを作成してから IPv6 インベントリを削除するには，［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，各管理対象ノードから設定情報を取得します。

14.7.3　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。

15 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。

15.1 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，「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」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，「17. 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。

15.2 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 15–1 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，リージョナルマネージャを設定できません。 HA • HA 用に各グローバルマネージャ，リージョナルマネージャを設定できます。 • 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またはリージョナルマネージャがフェイルオーバーすると，NNMi は，グローバルマネージャとリージョナルマネージャ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。

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

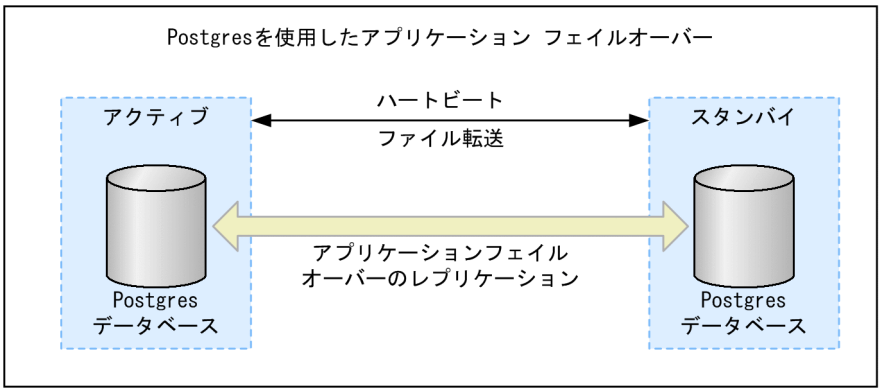
重要なネットワーク機器の障害発生を知らせ，その障害の根本原因を示す NNMi は，多くの IT プロフェッショナルから信頼を寄せられています。NNMi 管理サーバーに障害が発生した場合でも，引き続き NNMi がネットワーク機器の障害発生を知らせてくれる必要があります。このニーズを満たすのが NNMi のアプリケーションフェイルオーバーで，NNMi プロセスのアプリケーションコントロールをアクティブな NNMi 管理サーバーからスタンバイ NNMi 管理サーバーに引き渡すことで，NNMi の機能は中断なく提供されます。

16.1　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（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。

16.2 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

16.2.1 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 管理サーバーが同じ種類かつ同じバージョン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 HP-UX 11iV3 を実行している場合，スタンバイサーバーも HP-UX 11iV3 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 10-10-01 を実行している場合，スタンバイサーバーでも同一の NNMi 10-10-01 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。
- （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。

注意事項

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，「4.4　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（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
- NNMi を，LDAP を通じてディレクトリサービスに統合する予定があり，両方の NNMi 管理サーバーのパスワードがアプリケーションフェイルオーバーの構成前に暗号化されていた場合には，`ldap.properties` ファイルがセカンダリ NNMi 管理サーバーにコピーされていることを確認してください。

  このコピー処理は，セカンダリ NNMi 管理サーバーが初めてクラスタに追加されたあとに行われているはずです。

  このコピー処理が正常に行われたかどうかを確認するには，クラスタを有効にして，5 分間待ちます。次に，`ldap.properties` ファイルを調べて，セカンダリ NNMi 管理サーバーにコピーされたことを確認します。詳細については，「16.3　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。
- 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，「16.3　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 C　NNMi が使用するポートの一覧」を参照してください。

16.2.2　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「16.4.2　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「16.4.1　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。

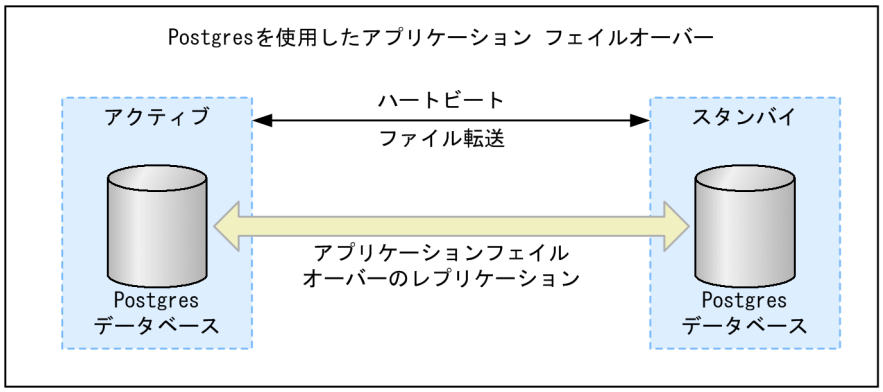
16.3　アプリケーションフェイルオーバー構成の NNMi を設定する

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。

16.3.1　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. マニュアル「JP1/Cm2/Network Node Manager i　インストールガイド」に記載のとおり，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. マニュアル「*JP1/Cm2/Network Node Manager i　インストールガイド*」の「*3.3　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
   ```

   UNIX

   ```
   $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
```

5. NNMi の証明書（nnm.keystore および nnm.truststore ファイル，または認証機関を使用する）を設定する。

選択した方法によって，「8.3　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」に示した指示，または「8.4　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」に示した指示を実行します。

注意事項

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.keystore ファイルおよびnnm.truststore ファイルをマージして，nnm.keystore ファイルおよび nnm.truststore ファイルをそれぞれ 1 つのファイルに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
```

UNIX

```
$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，「16.3.3　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。

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: 000000001000000005
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
============================================================
```

注意事項

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。

- クラスタ名が，サーバー X とサーバー Y で異なっているかどうか。
- 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
  UNIX
  $NnmDataDir/shared/nnm/conf/nnmcluster/cluster.keystore
- 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。
- nnm.keystore ファイルとnnm.truststore ファイルを確実にマージしたかどうか。
- 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 アドレスが一致しているかどうか。
  一致していない場合は，NIC から取得できる IP アドレスに解決できるホスト名を com.hp.ov.nms.cluster.member.hostnames に指定してください
- 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。

- 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 NNMi10-10 を実行しており，サーバー Y が NNMi10-10 の修正版を実行している場合などです。

9. サーバー X で，NNMi クラスタマネージャを開始する。

```
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。

16.3.2 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.keystore` および`nnm.truststore` ファイルの内容をマージして，それぞれ 1 つの `nnm.keystore` および`nnm.truststore` 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。

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。

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。

9. 最後の概要には，設定が成功したかどうかが示される。
   設定が成功していない場合，[前へ] をクリックして問題を修正します。
   クラスタのセットアップに成功している場合，ブラウザを閉じます。

10. 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。

11. 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，「16.3.3　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。

12. 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，「16.3　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，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 「16.3　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。

14. nnmcluster -daemon コマンドを使用して，スタンバイノードを起動する。

16.3.3　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（クラスタが通信に使用するネットワークインタフェース）を選択できます。

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
   - UNIX：$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。

   UNIX 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。

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 パラメータは，次のファイルで設定します。

   - Windows：%NnmDataDir%¥shared¥nnm¥conf¥props¥nms-cluster.properties
   - UNIX：$NnmDataDir/shared/nnm/conf/props/nms-cluster.properties

16.4　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。

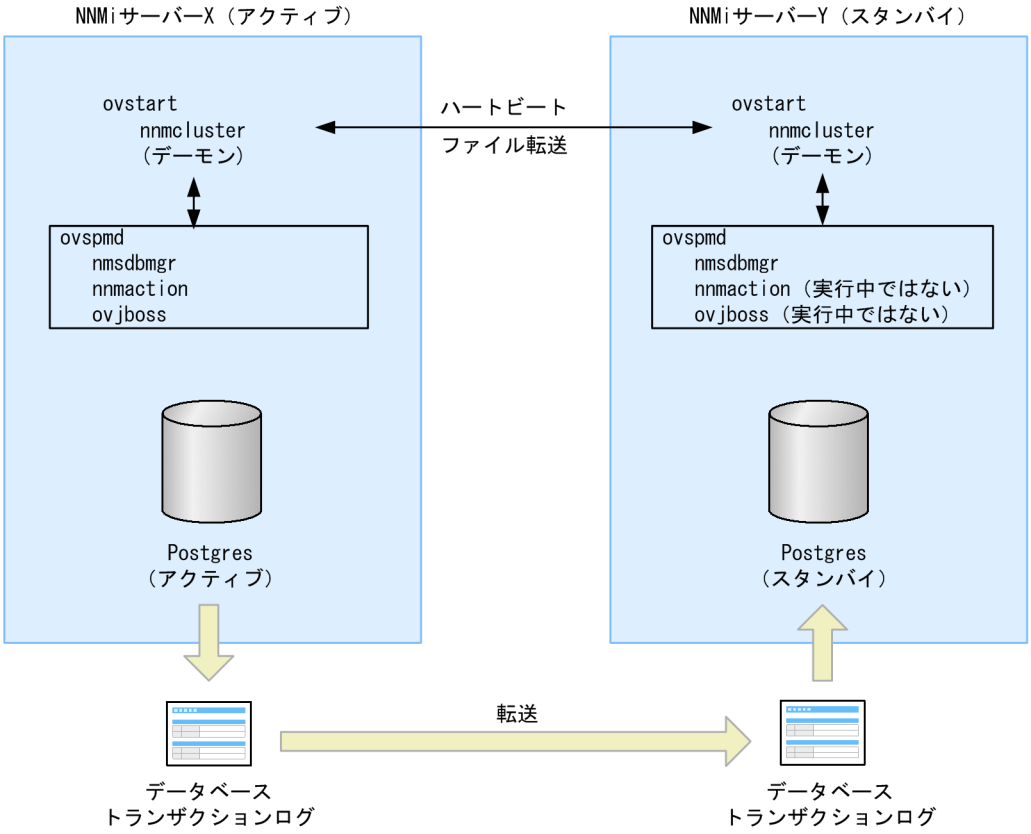
- デーモンモード：クラスタマネージャのプロセスはバックグラウンドで実行し，ovstop およびovstart コマンドを使用して NNMi サービスを開始および停止します。
- インタラクティブモード：クラスタマネージャは，NNMi 管理者がクラスタの属性を表示および変更できるインタラクティブセッションを実行します。例えば，NNMi 管理者はこのセッションを使用して，アプリケーションフェイルオーバー機能を有効または無効にしたり，デーモンプロセスをシャットダウンしたりできます。
- コマンドラインモード：NNMi 管理者は，コマンドプロンプトでクラスタの属性を表示および変更します。

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。

16.4.1　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 16-1　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 10-50 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 の場合：%NnmDataDir%shared¥nnm¥database¥

UNIX の場合：$NnmDataDir/shared/nnm/database/

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。

注意事項

- 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。

16.4.2 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 16–1 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 UNIX 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。

注意事項

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

「16.4.2(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（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（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。

注意事項

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。

16.4.3 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`

参考

- `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。

注意事項

UNIX 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。

16.4.4 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。

16.5　フェイルオーバーの問題解決後の設定

アクティブサーバーで障害が発生し，スタンバイサーバーがアクティブサーバーとして機能している場合，以前のアクティブサーバーで問題を解決したら，目的のアクティブなクラスタノードで次のコマンドを実行して，元の状態に戻します。

```
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。

16.6 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`

   UNIX

   `$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`

    UNIX

    `$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。UNIX 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します。このファイルは，Posgres 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
- UNIX：$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
- UNIX：$NnmDataDir/shared/nnm/databases/Postgres_standby

> **参考**
>
> 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
- UNIX：$NnmDataDir/shared/nnm/databases/Postgres.OLD

16.7　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。

16.7.1　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「22.4　アプリケーションフェイルオーバー構成の NNMi 10-50 へのアップグレード」を参照してください。

16.7.2　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，「16.4.3　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，「16.4　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. NNMi 管理サーバーを再起動する。
   a NNMi 管理サーバーで`ovstop` コマンドを実行します。
   b NNMi 管理サーバーで`ovstart` コマンドを実行します。
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。

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，「16.4.2 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（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。

16.7.3 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。

注意事項

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。

(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

- UNIX

  $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

UNIX

$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，「18.3　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

UNIX

$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。

16.7.4　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。

注意事項

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，「16.4　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

「2.8 設定をやり直す」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
   ```

   UNIX

   ```
   $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
   ```

   UNIX

   ```
   $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

- UNIX

  $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。

16.7.5　NNMi データベースパスワードの変更

1. 「16.6　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. 「16.3.2　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。

16.8 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，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。

16.8.1 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 16-2　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。

参考

- 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り，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。

17 高可用性クラスタに NNMi を設定する

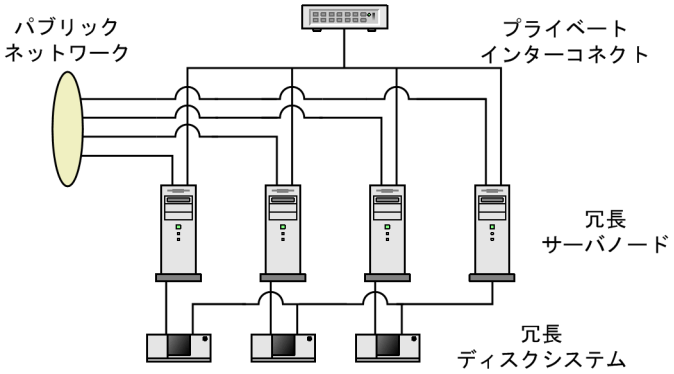
高可用性（HA）とは，構成された動作中のハードウェアおよびソフトウェアの一部に障害が発生しても中断されないサービスを提供するシステムです。HA クラスタは，フェイルオーバー発生時の機能とデータの継続性を保証するために，協調して動作するハードウェアとソフトウェアのグループ化を定義します。

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。HA 製品固有のコマンドの代わりに，この章で説明している NNMi のコマンドを使用できます。

17.1　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。

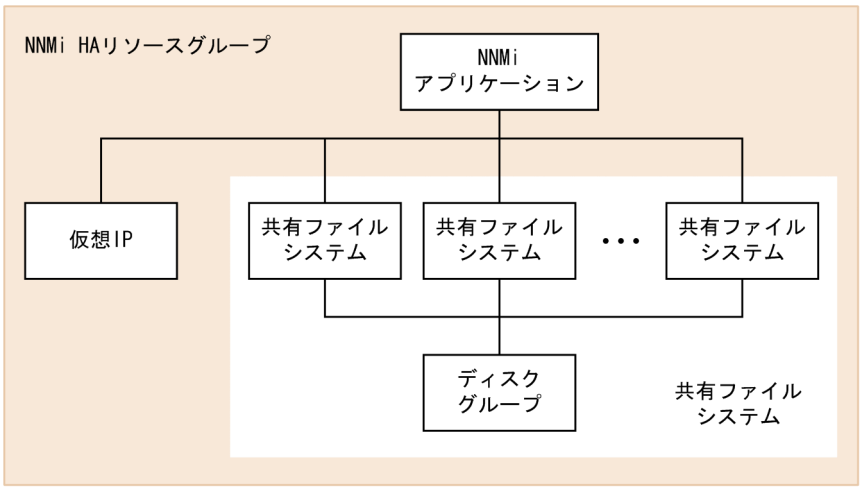
図 17–1　高可用性クラスタのアーキテクチャ



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。

HP Serviceguard，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 17–2　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 17–1　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
HP Serviceguard	SG，HP SG	パッケージ
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

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 17-1 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。

17.1.1　HA 用語集

次の表に，一般的な HA 用語の定義を示します。

表 17–2　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。
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の中で利用可能な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。

17.1.2　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）。

> **参考**
>
> 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。

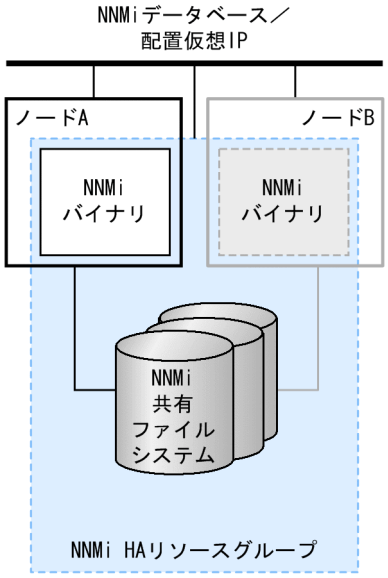
NNMi だけのシナリオ

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。

クラスタに三つ以上の NNMi ノードがある場合は，追加ノードには次の図のノード B と同様の設定を行います。

図 17-3　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，「17.4.2　HA 用に NNMi を設定する」を参照してください。

17.1.3　man ページ

NNMi の man ページには，HA 設定について，次の内容が含まれています。

- 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。

17.2 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 の場合：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。
- UNIX（HP Serviceguard）の場合：`scp -B` または`rcp` が実行可能な環境である必要があります。

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 の場合：<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）

  ・UNIX の場合：<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 の場合：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 UNIX を対象とした説明です。
- Windows の場合：Microsoft Cluster Services を使っている Windows Server 2008 または Windows Server 2012 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。

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，「17.4 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。

17.3 HA 設定の注意事項

HA 設定の注意事項を次に示します。

17.3.1 関連製品を使用する場合の注意

NNMi の関連製品（JP1/Cm2/SSO や JP1/IM - EG for NNMi）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。
- NNMi と，関連製品 JP1/Cm2/SSO，JP1/IM - EG for NNMi（および前提の JP1/Base）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。
  - JP1/Cm2/SSO は，NNMi を前提とする依存関係を設定します。
  - JP1/IM - EG for NNMi は，JP1/Base および 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。
  - JP1/Base 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。

  HP Serviceguard に登録するパッケージは次のように設定してください。
  - NNMi 用のパッケージに，関連製品の起動／停止／監視のコマンドを追加します。
  - 起動コマンドは，パッケージ制御スクリプト<resource_group>.cntl のcustomer_defined_run_cmds 部分に追加します。先に NNMi が起動するように設定してください。
  - 停止コマンドは，パッケージ制御スクリプト<resource_group>.cntl のcustomer_defined_halt_cmds 部分に追加します。NNMi があとに停止するように設定してください。
  - 監視コマンドは，監視スクリプト<resource_group>.mon を編集して追加してください。

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。

17.3.2 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。

- マニュアル「*JP1/Cm2/Network Node Manager i インストールガイド*」,「*JP1/Cm2/Network Node Manager i セットアップガイド*」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。

17.3.3 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。

  (Solaris の場合）起動のタイムアウトが発生する場合は，「17.8.3 一般的な HA のトラブルシューティング」の「(2) 製品の起動タイムアウト」を参照してください。
- (Windows 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 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。

17.4　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。

17.4.1　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，「8.5　自己署名証明書または CA 証明書を使用するように高可用性クラスタを設定する」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。

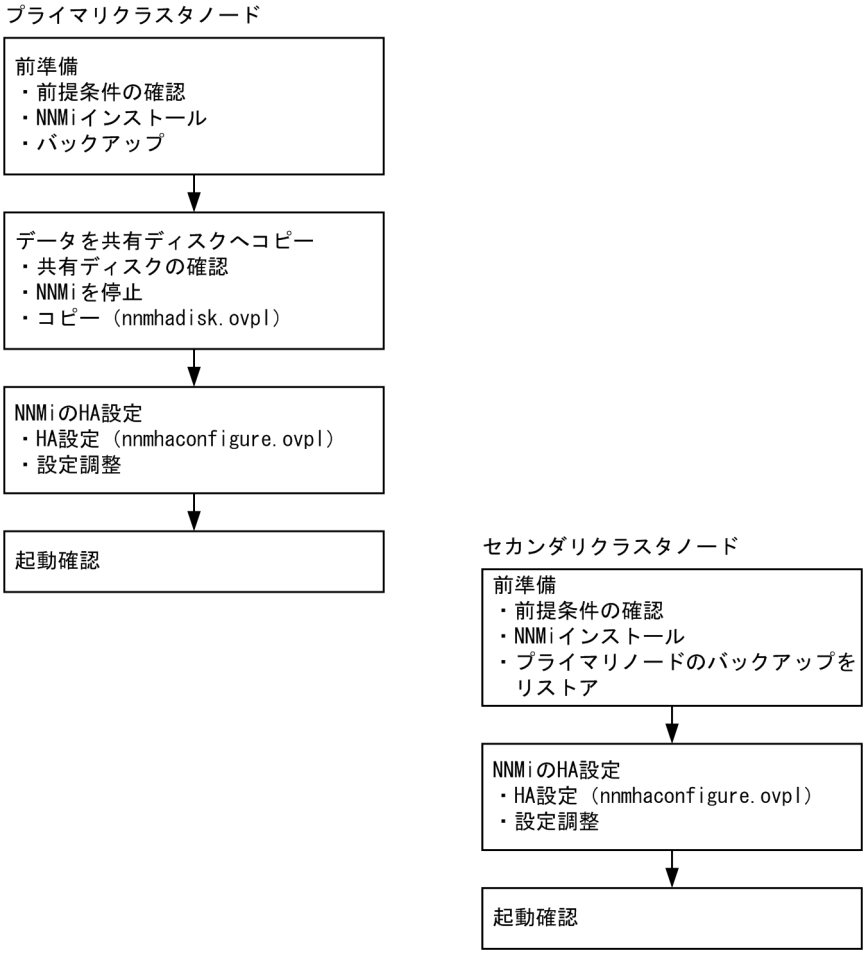
17.4.2　HA 用に NNMi を設定する

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すべて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。

注意事項

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，「8.5　自己署名証明書または CA 証明書を使用するように高可用性クラスタを設定する」を参照してください。

参考

- 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。1 つのクラスタノードで HA 設定プロセスが完了したあと，次の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，クラスタ環境内のすべてのノードで HA 用に NNMi を設定するまで，この作業を繰り返します。
- HA モニタの場合は，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。

(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 17-3 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の仕様に準じます。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。 **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，UNIX 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。

HA 設定項目	説明
	(例) • Windows の場合：ローカルエリア接続 • HP-UX の場合：lan0 • Linux の場合：eth0 • Solaris の場合：bge0
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/Cm2/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	(UNIX 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。 • HP-UX の場合：vxfs • Linux の場合：VCS または SCS には ext2，ext3，および vxfs • Solaris の場合：vxfs
ディスクグループ	(UNIX，VCS または SCS 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	(UNIX 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
論理ボリューム	(UNIX，HPSG だけ) NNMi 共有ファイルシステムの論理ボリュームの名前です。 例 lvol01
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 の場合： Y:またはY:¥JP1 • UNIX の場合： /nnmmount

HA 設定項目	説明
	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 の場合：　*<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 • UNIX の場合：　*<マウントポイント>*/NNM

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，クラスタの仕様に準じます。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
- Solaris VCS または Solaris SCS，および 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
- HP-UX HP SG 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（.）
    ただし先頭は英字
  - 文字数：31 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

17.4.3　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 17-4　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
<ネットワーク名リソース>	ネットワーク名	仮想ホスト名を制御する
<IP アドレスリソース>	IP アドレス	仮想 IP アドレスを制御する
<ディスクリソース>	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。
- *<ディスクリソース>*は，WSFC の場合は手動で設定します。

(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 17-5　<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) • 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

項目	詳細
[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 17-6　<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 17-7　<ディスクリソース>

項目	詳細
[全般]	• リソース名：クラスターディスク • 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 17-8　<NNMi 用の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：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※

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. 「17.2　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，インストールします。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，「18.　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。

注意事項

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
- 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
```

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>
```

注意事項

*<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。

WSFC の場合は，共有ディスクの所有者になっているノードで実施する必要があります。共有ディスクをノードが所有しているかについては，フェイルオーバークラスタ管理で確認できます。

(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，「17.9.2 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。「17.4.2 HA 用に NNMi を設定する」の表 17-3 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	jp1ha1
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

注意事項

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。
- WSFC の場合：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: ? jp1ha1
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 'jp1ha1-Name' を作成しています...

リソース             グループ             ノード          状態
-------------------- -------------------- --------------- ------
jp1ha1-Name          jp1ha1               NNMX64-33       オフライン


リソース 'jp1ha1-Name' 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
jp1ha1.xxx.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。

jp1ha1.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. WSFC の場合は，リソースグループにディスクリソースを登録し，汎用スクリプトリソースと依存関係を設定する。

ディスクリソースの登録は，フェイルオーバークラスタ管理コンソールを使用します。共有ディスクがクラスタサービスに登録されている場合の設定例を次に記載します。

- フェイルオーバークラスタ管理コンソールで，<*resource_group*>を開きます。
- ［記憶域の追加］を選び，適切なディスクリソースを登録します。
- <*resource_group*>-APP のプロパティの［依存関係］タブを開きます。
- 依存関係に，ディスクリソースを登録します。AND/OR は AND を指定します。

3. 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。

WSFC の場合

<*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。

注意事項

<*resource_group*>および<*resource_group*>に登録したリソースグループの設定によって，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。

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。

注意事項

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，「8.5　自己署名証明書または CA 証明書を使用するように高可用性クラスタを設定する」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，「17.8　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。

注意事項

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

セカンダリクラスタノードでは 1 つのノードごとに順番に次の手順を実行します。

(a)　前準備

1. 「17.2　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。

   「17.4.3(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，「18.　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
```

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
jp1ha1.xxx.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. 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。

4. 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。

5. 任意で，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。

注意事項

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）

17.4.4 HA 用に NNMi を設定する（UNIX の場合）

ここでは，UNIX 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 17-9　HP Serviceguard での NNMi 用リソースグループの構成

項目	ファイル名
パッケージ構成ファイル	/etc/cmcluster/<*resource_group*>/<*resource_group*>.conf
パッケージ制御スクリプト	/etc/cmcluster/<*resource_group*>/<*resource_group*>.cntl
監視スクリプト	/etc/cmcluster/<*resource_group*>/<*resource_group*>.mon

HPSG の場合，nnmhaconfigure.ovpl が上記のパッケージの設定ファイル一式を/etc/cmcluster/<*resource_group*>に配置して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。

表 17-10　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 DiskGruop 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，MultiNICA，MultiNICB など）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，「17.8　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=<mountpoint>"
  )

  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> -return 1 0"
     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 17-11　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（Cm2 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。

注意事項

HA モニタの場合は，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. 「17.2 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，インストールします。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，「18. 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。

   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 sys で作成され，パーミッションには 555 が設定されていることを確認します。

   （例）

   ```
   ls -l
   ```

2. NNMi HA リソースグループ用に，共有ディスクを用意する。

   注意事項

   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）

   - Solaris 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 -F vxfs /dev/vx/dsk/<disk_group>/<volume_group> <HA_mount_point>
```

- 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_group> <HA_mount_point>
```

- HP-UX HP SG の場合

```
vgchange -c n <volume_group>
vgchange -a y <volume_group>
mount /dev/<volume_group>/<logical_volume> <HA_mount_point>
```

4. NNMi を停止する。

```
/opt/OV/bin/ovstop -c
```

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>
```

注意事項

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>
```

- HP-UX HP SG の場合

```
umount <HA_mount_point>
vgchange -a n <volume_group>
vgchange -c y <volume_group>
```

操作時に HP SG が動作している必要があります。

(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，「17.9.2 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。「17.4.2 HA 用に NNMi を設定する」の表 17-3 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	jp1ha1
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
論理ボリューム（HP SG だけ）	lvol1
マウントするディレクトリ	/shdsk1

注意事項

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。

  HP-UX HPSG の場合：ja_JP.SJIS またはja_JP.eucJP

  Solaris VCS または Solaris SCS の場合：ja_JP.PCK またはja_JP.eucJP

  Linux VCS または Linux SCS の場合：ja_JP.UTF-8

  HA 構成の設定後にロケールを変更する場合は，「17.6 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。

- HPSG (HP-UX)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: ？ jp1ha1
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
none            lan3*
192.168.69.0            lan2
10.208.69.0             lan0

選択可能な値:
1: lan3*
2: lan2
3: lan0
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: ？ 3

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
質問: ディスク タイプの名前を入力してください: ？ 1
質問: ボリューム グループの名前を入力してください: ？ vg03

選択可能な値:
1: group
2: lvol1
3: rlvol1
```

```
質問: 論理ボリュームの名前を入力してください:  ? 2
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。
Completed the cluster update
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。
#
```

- 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:  ? jp1ha1
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
attribute 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 jp1ha1
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。

jp1ha1.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. 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。

注意事項

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，「8.5　自己署名証明書または CA 証明書を使用するように高可用性クラスタを設定する」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たずに，プロンプトが返ってくるため，HA クラスタソフトのコマンドを使用して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，「17.8　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。

注意事項

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

セカンダリクラスタノードでは 1 つのノードごとに順番に次の手順を実行します。

(a)　前準備

1. 「17.2　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。

   「17.4.4(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，「18.　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，「17.4.4(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
   ```

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
   jp1ha1.xxx.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. 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
```

5. 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。

6. 任意で，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。

17.5　共有 NNMi データ

HA 環境で実行する NNMi は，HA クラスタ内のすべての 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。

注意事項

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。

17.5.1　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 の場合
  －%NnmDataDir%は，%HA_MOUNT_POINT%¥NNM¥dataDir にマッピングされます。
- UNIX の場合
  －$NnmDataDir は，$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ディレクトリは，次のとおりです。

- Windows の場合
  －%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¥eventdb
  pmd イベントデータベース。
  －%NnmInstallDir%nonOV¥jboss¥nms¥server¥nms¥data
  ovjboss で使われるトランザクションストア。
- UNIX の場合
  －$NnmDataDir/shared/nnm/databases/Postgres
  組み込みデータベース。
  －$NnmDataDir/log/nnm
  NNMi のロギングディレクトリ。
  －$NnmDataDir/shared/nnm/databases/eventdb
  pmd イベントデータベース。

－$NnmInstallDir/nonOV/jboss/nms/server/nms/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。

17.5.2 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。

17.6　HA 設定のメンテナンス

17.6.1　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。

注意事項

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 の場合
  `%NnmDataDir%hacluster¥<resource_group>¥maintenance`
- UNIX の場合
  `$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，「17.6.1(1)　NNMi をメンテナンスモードにする」を参照してください。

17.6.2　HA クラスタ内の NNMi をメンテナンスする

(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 の場合

  %NnmInstallDir%misc¥nnm¥ha¥nnmhastoprg.ovpl NNM <resource_group>

- UNIX の場合

  /opt/OV/misc/nnm/ha/nnmhastoprg.ovpl NNM <resource_group>

リソースグループの停止を待たずにプロンプトが返ってくるため，HA クラスタソフトの管理コンソールやコマンドを使用してリソースグループの停止を確認してください。

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 の場合

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タブを選択し，新しい IP アドレスを入力します。

- UNIX の場合

  /opt/OV/misc/nnm/ha/nnmhargconfigure.ovpl NNM <resource_group> -set_value <resource_group>-ip Address <new_IP_address>

  （VCS または SCS の場合）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 の場合

     %NnmInstallDir%misc¥nnm¥ha¥nnmhastartrg.ovpl NNM <resource_group>

   - UNIX の場合

     /opt/OV/misc/nnm/ha/nnmhastartrg.ovpl NNM <resource_group>

   リソースグループの起動を待たずに，プロンプトが返ってくるため，HA クラスタソフトのコマンドを使用してリソースグループの起動を確認してください。

4. NNMi を正常に起動できたことを確認する。

   - Windows の場合

     %NnmInstallDir%bin¥ovstatus -c

   - UNIX の場合

     /opt/OV/bin/ovstatus -c

(c) 物理ホスト名の変更

手順については，「20.5 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」を参照してください。

(d) 物理 IP アドレスの変更

手順については，「20.4 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. 「17.6.1(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. 「17.6.1(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「17.6.1　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 の場合

     ```
     %NnmInstallDir%misc¥nnm¥ha¥nnmhaclusterinfo.ovpl -group <resource_group> -activeNode
     ```

   - UNIX の場合

     ```
     /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「17.6.1 NNMi をメンテナンスモードにする」を参照してください。

注意事項

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。

注意事項

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. nnmrestore.ovpl コマンドでのリストアを行う場合は，残りすべてのノードで手順 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) ロケールの変更

HA 構成構築後，ロケールを変更したい場合は次の手順を実施してください（HP-UX，Solaris の場合だけ）。

1. NNMi HA リソースグループを停止する。

```
/opt/OV/misc/nnm/ha/nnmhastoprg.ovpl NNM <resource_group>
```

上のコマンドを実行後，HA クラスタソフトのコマンドを使用してリソースグループの停止を確認してください。

2. 次のコマンドを実行する。

```
/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -config NNM -set HA_LOCALE <LANG>
```

<*LANG*>は下記のどちらかを指定してください

- HP-UX HPSG の場合：ja_JP.SJIS または ja_JP.eucJP
- Solaris VCS または Solaris SCS の場合：ja_JP.PCK または ja_JP.eucJP

3. 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>
```

HA クラスタソフトのコマンドを使用してリソースグループの起動を確認してください。

4. NNMi を正常に起動できたことを確認する。

```
/opt/OV/bin/ovstatus -c
```

(8) 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 の場合
     %NnmInstallDir%misc¥nnm¥ha¥nnmhaclusterinfo.ovpl -group <*resource_group*> -activeNode
   - UNIX の場合
     /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 の場合
     %NnmInstallDir%bin¥nnmresetembdb.ovpl
   - UNIX の場合
     /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。

17.7　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。

高可用性用の NNMi の設定を維持するには，HA クラスタに，実行中の 1 つの NNMi ノードと，少なくとも，1 つのパッシブなクラスタノードが必要です。HA クラスタから NNMi を完全に削除するには，クラスタ内のすべてのノードで HA 機能の設定を解除します。

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。

- 17.7.1 アクティブなクラスタノードの特定
- 17.7.2 パッシブなクラスタノードでの設定解除
- 17.7.3 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。

> **参考**
>
> HA モニタの場合は，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。

17.7.1　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 の場合
     %NnmInstallDir%misc¥nnm¥ha¥nnmhaclusterinfo.ovpl -group <*resource_group*> -activeNode
   - UNIX の場合
     /opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -group <*resource_group*> -activeNode

17.7.2　パッシブなクラスタノードでの設定解除

パッシブなクラスタノードが複数ある場合は，解除するノードごとに次の手順を実施してください。

1. パッシブなクラスタノードごとで，HA クラスタから NNMi の設定を解除する。
   - Windows の場合
     %NnmInstallDir%misc¥nnm¥ha¥nnmhaunconfigure.ovpl NNM <*resource_group*>

- UNIX の場合

  /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。

- HP-UX HP SG の場合

  Unable to perform the security token exchange with cmclconfd on node xxxxx

  Cannot connect to configuration daemon (cmclconfd) on node xxxxx

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。

   <*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 の場合

     %NnmInstallDir%bin¥nnmsetofficialfqdn.ovpl -force <*FQDN*>

   - UNIX の場合

     /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。

   - 「新しい証明書を生成できません。自己署名されたエイリアス xxx.xxx.xxx はすでにキーストアに存在します。」が表示された場合

     nms-local.properties ファイルを編集してください。

   ファイルのパス

   Windows の場合：%NnmDataDir%conf¥nnm¥props¥nms-local.properties

   UNIX の場合：/var/opt/OV/conf/nnm/props/nms-local.properties

   編集内容（自己署名証明書を使用する場合）

   com.hp.ov.nms.ssl.KEY_ALIAS = <*FQDN*>.selfsigned

   <*FQDN*>は nnmsetofficialfqdn.ovpl で指定した FQDN を小文字で記述します。

   証明書の詳細については，「8. 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。

- WSFC の場合

  エクスプローラで，%NnmDataDir%hacluster¥<*resource_group*>¥フォルダを削除します。

- HP Serviceguard の場合

  ```
  cd /var/opt/OV/hacluster/
  rm -r <resource_group>
  cd /etc/cmcluster/
  rm -r <resource_group>
  ```

- VCS または SCS の場合

  ```
  cd /var/opt/OV/hacluster/
  rm -r <resource_group>
  ```

  ほかに解除するパッシブなクラスタノードがなければ，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，すべての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。

17.7.3 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 リソースグループを停止する。

- Windows の場合

  %NnmInstallDir%misc¥nnm¥ha¥nnmhastoprg.ovpl NNM <*resource_group*>

- UNIX の場合

  $NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl NNM <*resource_group*>

リソースグループの停止を待たずにプロンプトが返ってくるため，HA クラスタソフトの管理コンソールやコマンドを使用してリソースグループの停止を確認してください。

2. NNMi を使用している環境によっては，次の手順を実施する必要がある。

- WSFC の場合

  リソースグループを停止したあと，フェイルオーバークラスタ管理コンソールから次の手順を実施します。

  <*resource_group*>-APP の依存関係からディスクリソースを削除します。

  <*resource_group*>からディスクリソースを削除します。

3. 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 の場合

%NnmInstallDir%misc¥nnm¥ha¥nnmhaunconfigure.ovpl NNM <*resource_group*>

- UNIX の場合

  $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。

- HP SG の場合

  Unable to perform the security token exchange with cmclconfd on node xxxxx

  Cannot connect to configuration daemon (cmclconfd) on node xxxxx

- VCS または SCS の場合

  VCS WARNING V-16-1-10133 Group does not exist: <*resource_group*>

4. NNMi ノードの FQDN 設定を物理ホスト名に変更する。

<*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 の場合

  %NnmInstallDir%bin¥nnmsetofficialfqdn.ovpl -force <*FQDN*>

- UNIX の場合

  /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。

- 「新しい証明書を生成できません。自己署名されたエイリアス xxx.xxx.xxx はすでにキーストアに存在します。」が表示された場合

  nms-local.properties ファイルを編集してください。

ファイルのパス

・Windows の場合：%NnmDataDir%conf¥nnm¥props¥nms-local.properties

・UNIX の場合：/var/opt/OV/conf/nnm/props/nms-local.properties

編集内容

- 自己署名証明書を使用する場合

  com.hp.ov.nms.ssl.KEY_ALIAS = <*FQDN*>.selfsigned

  <*FQDN*>は nnmsetofficialfqdn.ovpl で指定した FQDN を小文字で記述します。

証明書の詳細については，「8. NNMi での証明書の使用」を参照してください。

5. 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。

- WSFC の場合

  エクスプローラで，%NnmDataDir%hacluster¥<*resource_group*>¥フォルダを削除します。

- HP Serviceguard の場合

  cd /var/opt/OV/hacluster

  rm -r <*resource_group*>

  cd /etc/cmcluster

  rm -r <*resource_group*>

- VCS または SCS の場合

  cd /var/opt/OV/hacluster

  rm -r <*resource_group*>

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。

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 の場合

  %NnmInstallDir%misc¥nnm¥ha¥nnmhadisk.ovpl NNM -from <HA_mount_point>

- UNIX の場合

  /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。

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

10. 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 の場合

  %NnmInstallDir%bin¥nnmbackup.ovpl -type offline -scope all -target <directory>

- UNIX の場合

  /opt/OV/bin/nnmbackup.ovpl -type offline -scope all -target <directory>

11. 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，1 つのノードごとに順番に，手順 10.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 の場合

  %NnmInstallDir%bin¥nnmrestore.ovpl -force -source <backup_data>

- UNIX の場合

  /opt/OV/bin/nnmrestore.ovpl -force -source <backup_data>

このコマンドの詳細については，「18. 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。

17.8 HA 設定のトラブルシューティング

17.8.1 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 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，共有ディスクのマウントポイントに存在している。
  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。
- 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，他方では無効です。
- タイムアウトが短すぎる。
  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設定され，その後すぐにフェイルオーバーを実行する場合，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。

17.8.2 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。

17.8.3 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 2008 R2)

Windows Server 2008 R2 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
```

注意事項

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) 製品の起動タイムアウト

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-16-1-13012 Thread(...) Resource(<resource group>-app): online procedure did not
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完全には起動できなかったことを示しています。NNMi に付属した HA 設定スクリプトでは，タイムアウトは 30 分と定義されています。

Solaris オペレーティングシステムでの 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 -modify <resource_group>-app OnlineTimeout <value in seconds>
/opt/VRTSvcs/bin/haconf -dump -makero
```

(3) 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
not 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。

Solaris オペレーティングシステムでの 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
```

(4) 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。

(5) 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。
- Serviceguard の場合

<*resource_group*>.cntl.log ファイルとsyslog ファイルにエラーが記録されていないか調べてください。良くある原因は，リソースを追加できない状態（例えば，ディスクグループの設定を誤っているため，アクティブにできない）のままで，システムが放置されていることです。

－HP-UX の場合/etc/cmcluster/<*resource_group*>/<*resource_group*>.cntl.log

－Linux の場合/usr/local/cmcluster/conf/<*resource_group*>/<*resource_group*>.cntl.log

- 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。

   - Serviceguard の場合

     －HP-UX の場合：/usr/sbin/cmmodnet を実行して，IP アドレスを追加します。

     －Linux の場合：/usr/local/cmcluster/bin/cmmodnet を実行して，IP アドレスを追加します。

   - 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 の場合

     %NnmInstallDir%¥misc¥nnm¥ha¥nnmhastartrg.ovpl NNM -start <*resource_group*>

   - UNIX の場合

     $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。

(6) 「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。

17.8.4 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) すべてのクラスタノードを設定解除したあとの HA 用の NNMi を再び有効にする

すべての NNMi HA クラスタノードの設定を解除した場合は，NNMi の共有ディスクのマウントポイントへのリンクが，`ov.conf` ファイルから削除されます。共有ディスク内のデータを上書きすることなく，マウントポイントへのリンクを作成し直すには，プライマリクラスタノードで次の手順を実行します。

1. NNMi が実行中であれば，停止する。
   ```
   ovstop -c
   ```
2. 共有ディスクへのリンクを削除する。
   - Windows の場合
     `%NnmInstallDir%misc¥nnm¥ha¥nnmhadisk.ovpl NNM -setmount <HA_mount_point>`
   - UNIX の場合
     `$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM -setmount <HA_mount_point>`
3. `ov.conf` ファイルの HA マウントポイント関連のエントリを確認する。
   `ov.conf` ファイルの場所は，「17.9.1 NNMi HA 設定ファイル」を参照してください。

(2) 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

   UNIX：$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

   UNIX：$NnmDataDir/hacluster/<*resource_group*>/maintenance

(3) 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_EVENTDB_DIR=<HA_mount_point>/NNM/dataDir/shared/nnm/databases/eventdb
- 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/installDir/nonOV/jboss/nms/server/nms/data

- HA_LOCALE=<*ロケール*>（UNIXだけ）
- HA_PERFSPI_ADAPTER_DIR=<HA_mount_point>/NNM/dataDir/shared/perfSpi/datafiles

ov.conf ファイルの場所は，「17.9.1　NNMi HA 設定ファイル」を参照してください。

(4)　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
   - UNIX：$NnmDataDir/hacluster/<*resource_group*>/maintenance
2. NNMi データベースを共有ディスクにコピーする。
   - Windows：%NnmInstallDir%¥misc¥nnm¥ha¥nnmhadisk.ovpl NNM ¥
     -to <*HA_mount_point*>
   - UNIX：$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*>
   - UNIX：$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

- UNIX：$NnmDataDir/hacluster/<*resource_group*>/maintenance

(5) 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。

(6) 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。

(7) 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戻されません。

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。

(8) フェイルオーバー後にセカンダリクラスタノードが共有ディスクファイルを見つけられない

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

UNIX：$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
```

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*>

UNIX：$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*>

UNIX：$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

UNIX：$NnmDataDir/hacluster/<*resource_group*>/maintenance

17.9 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。

17.9.1 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 の場合
  `%NnmDataDir%shared¥nnm¥conf`
- UNIX の場合
  `$NnmDataDir/shared/nnm/conf`

表 17-12 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。

17.9.2 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 の場合
  `%NnmInstallDir%misc¥nnm¥ha`
- UNIX の場合
  `$NnmInstallDir/misc/nnm/ha`

表 17–13　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。
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 17-14 に示した NNMi 付属のスクリプトは，表 17-13 に示したスクリプトで使用します。表 17-14 に示したスクリプトは直接実行しないでください。

表 17–14　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。 Serviceguard 設定では，<*resource_group*>.cntl で使用します。 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

17.9.3 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

- UNIX 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。

- HP-UX 実行時

  －/etc/cmcluster/<*resource_group*>/<*resource_group*>.cntl.log

  これは，リソースグループ用のログファイルです。

  －/var/adm/syslog/syslog.log

  －/var/adm/syslog/OLDsyslog.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

- VCS または SCS 用の Linux または Solaris の場合

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

リソース	ログファイル
	• 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。

18 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，次の 2 種類です。

ファイルシステム内のファイル

リレーショナルデータベースのデータ

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。

18.1 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。

18.2 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。

18.2.1 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。

18.2.2 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 18-1　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 18-1　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 18-1　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 18-2　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 18-1　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	
%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 18–2　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

18.3 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。

参考

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。

注意事項

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。

参考

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。

18.3.1　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

18.3.2　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。

ポイント

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。

参考

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。

18.4　バックアップとリストアの方針

18.4.1　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「18.3.1　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，「18.3.2　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。

18.4.2　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，「18.2.2　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
```

18.4.3 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，「18.3.2　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。

18.4.4 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
```

18.5　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。

ポイント

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。

19 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。

19.1 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，UNIX 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，UNIX の 3 つのすべてのプラットフォーム（RedHat，HP-UX，および Solaris）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。これら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。

RedHat Linux，および Solaris：

```
setfacl -m user:<USER>:rwx /var/opt/OV/log/nnm/public
```

HPUX：

```
setacl -m user:<USER>:rwx /var/opt/OV/log/nnm/public
```

Solaris ZFS：

```
chmod A+user:<USER>:read_data/add_file/write_data/
list_directory:allow /var/opt/OV/log/nnm/public
```

setfacl，setacl，またはchmod 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。

19.2　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「*カスタムポーリングを設定する*」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。

19.2.1　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
- UNIX：$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
   - UNIX：$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
   ```

   注意事項

   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. NNMi 管理サーバーを再起動する。
   a　NNMi 管理サーバーでovstop コマンドを実行します。
   b　NNMi 管理サーバーでovstart コマンドを実行します。

19.2.2 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
   - UNIX：$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. NNMi 管理サーバーを再起動する。
   a NNMi 管理サーバーでovstop コマンドを実行します。
   b NNMi 管理サーバーでovstart コマンドを実行します。

19.2.3 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
   - UNIX：$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. NNMi 管理サーバーを再起動する。
   a NNMi 管理サーバーでovstop コマンドを実行します。

b　NNMi 管理サーバーでovstart コマンドを実行します。

19.3 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。

19.3.1 同時アクション数を設定する

Solaris NNMi 管理サーバーで同時アクションの数を増加すると，NNMi のパフォーマンスが低下します。

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
   - UNIX：$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. NNMi 管理サーバーを再起動する。
   a NNMi 管理サーバーでovstop コマンドを実行します。
   b NNMi 管理サーバーでovstart コマンドを実行します。

19.3.2 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
   - UNIX：$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. NNMi 管理サーバーを再起動する。
   a NNMi 管理サーバーでovstop コマンドを実行します。
   b NNMi 管理サーバーでovstart コマンドを実行します。

19.3.3 アクションサーバー名のパラメータを設定する

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。

HP-UX，Solaris および 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。
   a ovstop nnmaction
   b ovstart nnmaction

19.3.4　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。

Solaris NNMi 管理サーバーの場合，NNMi の稼働状態情報でアクションキューサイズが大きくなっていることが示されると，パフォーマンスを上げるために最大メモリサイズが削減されます。

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。
   - %NNM_PROPS%¥nnmaction.properties
   - $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をニーズに合わせて調整する。
4. 変更を保存する。
5. NNMi 管理サーバーを再起動する。
   a　NNMi 管理サーバーでovstop コマンドを実行します。
   b　NNMi 管理サーバーでovstart コマンドを実行します。

19.3.5　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 19–1　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）

項目	説明
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
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

19.4 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 がブロックすることになったとします。インシデントのブロックが発生すると，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。また，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。

インシデントの数が一定のレートに達しないように，次の方法でインシデントをブロックします。

nnmtrapd.conf ファイルによる方法

nnmtrapd.conf ファイルを使用し，NNMi が監視するインシデントをブロックして，インシデントトラフィックを減らしてください。ただし，nnmtrapd.conf ファイルによる方法では，NNMi は依然として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 ファイルによる方法

nnmtrapd.conf ファイルを使用する方法より適切な方法です。NNMi にはフィルタリングメカニズムがあり，NNMi イベントパイプラインで早期にインシデントがブロックされ，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が回避されます。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，インシデントの量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。

19.5 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
   - UNIX：$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. NNMi を再起動する。

```
ovstop
ovstart
```

19.6　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

デフォルトの NNMi では，NNMi 管理者がノードを削除できます。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，ノードを削除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ノードを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
   - UNIX：$NNM_PROPS/nms-ui.properties
2. 次の行を含んでいるテキストブロックを検索する。

```
#!com.hp.nnm.ui.level2NodeDelete = true
```

3. 次の行のコメントを解除して，次のように編集する。

```
com.hp.nnm.ui.level2NodeDelete = true
```

4. 変更を保存する。
5. 次のどちらかを実行して，NNMi を構成し，ノード削除のための正しいパーミッションを設定する。
   - オプション 1：NNMi レベル 2 オペレータの NNMi 管理権限を持つセキュリティグループを作成します。このセキュリティグループを構成して，NNMi レベル 2 オペレータが削除できるノードのセットを含むようにします。
   - オプション 2：次のようにして，nms-topology.properties ファイルにエントリを追加します。

   a

   次のファイルを編集します。

   Windows：%NNM_PROPS%¥nms-topology.properties

   UNIX：$NNM_PROPS/nms-topology.properties

   b

   ファイルの最後までスクロールして，次の行を追加します。

   permission.override.com.hp.nnm.DELETE_OBJECT=com.hp.nnm.ADMIN,com.hp.nnm.LEVEL2

   c

   変更を保存します。
6. 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。

手順 1.から手順 6.までを行うと，NNMi コンソールは次のように変化します。

- NNMi レベル 2 オペレータユーザーグループメンバーのノードビューに［アクション］＞［削除］メニュー項目が，ツールバーに削除ボタン（アイコン）が含まれます。
- ノードフォームに［アクション］メニューが，ツールバーに［ノードを削除］ボタンが含まれます。

19.7　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
   - UNIX：$NNM_PROPS/nms-ui.properties
2. 次の行を含んでいるテキストブロックを検索する。

```
#!com.hp.nnm.ui.level2MapEditing = true
```

3. 次の行のコメントを解除して，次のように編集する。

```
com.hp.nnm.ui.level2MapEditing = true
```

4. 変更を保存する。
5. 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。

- NNMi レベル 2 オペレータの場合，［インベントリ］＞［ノードグループ］メニューに作成および削除ツールバーアイコンが表示されます。
- NNMi レベル 2 オペレータの場合，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］ボタンが含まれます。また，［デバイスフィルター］などのタブが含まれます。

ノードグループマップの場合，NNMi コンソールに［レイアウトの保存］ツールバーボタンと，［ファイル］＞［レイアウトの保存］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

のユーザーがノードグループマップの設定を作成するパーミッションを持つように，NNMi を構成できます。

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. 変更しなければならないノードグループを開く。
3. ［ファイル］＞［ノードグループマップの設定を開く］を開く。
4. ［レイアウトの保存のための最小 NNMi ロール］を［オペレータレベル 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。

19.8 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。

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。

   このステップによって，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。

4. [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。

5. [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。

6. 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。

   このステップによって，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。

7. 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
   - UNIX：$NNM_PROPS/nms-topology.properties

8. 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
```

9. 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
```

10. 変更を保存する。

11. NNMi 管理サーバーを再起動する。

   a　NNMi 管理サーバー上でovstop コマンドを実行します。

   b　NNMi 管理サーバー上でovstart コマンドを実行します。

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。

19.9 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

NNMiが管理対象外のデバイスからSNMPv3トラップを受信することがあります。NNMiを構成して，これらのデバイスのSNMPv3 engineIDをSNMPv3キャッシュに追加できます。

このようにNNMiを構成することによって，NNMiはこれらのSNMPv3トラップを認証し，格納できます。

次のようにして，これらのSNMPv3トラップを受信し，格納するようにNNMiを構成します。

1. NNMiコンソールで，［設定］＞［通信の設定］を選択する。

   NNMiが受信する各トラップの情報と一致するように，通信の設定を構成します。詳細については，NNMiヘルプの「*デフォルトのSNMPv3を設定する*」を参照してください。

   SNMPv3ノードのアドレス範囲を含む領域を使用するか，それぞれに特定のノード設定を構成するとよいです。

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］を選択する。

   ［未解決のSNMPトラップおよびSyslogメッセージを破棄する］の選択を解除します。

   ［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は管理対象外のノードから送信されたトラップを保持します。

3. NNMi管理サーバー上でovstopコマンドを実行する。

4. 次のファイルを編集する。

   - Windows：%NNM_PROPS%¥nms-communication.properties
   - UNIX：$NNM_PROPS/nms-communication.properties

5. ファイルの末尾に次の行を追加する。

   ```
   com.hp.nnm.snmp.engineid.file=<path to file>file.txt
   ```

   <path to file>file.txtエントリは，デバイスを含んでいるファイルのフルパスとファイル名です。これらの構成変更によって，NNMiプロセスを再起動するたびに，NNMiはこのファイルのエントリをSNMPv3キャッシュに読み込みます。

   Linux NNMi管理サーバーでは，ファイルパスは/var/opt/OV/etcなど，通常の形式になります。

   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。

6. 変更を保存する。

7. <path to file>file.txtファイルを編集する。

8. デバイスの IP アドレス，ポート，およびエンジン ID をコンマで区切って追加する。各デバイスのエントリを各行に 1 つずつ追加する。

エンジン ID は，一連の 16 進数のバイトです。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,
8000000903000001f9ea33000
ff06::c3,161,80 00 00 09 03 00 00 1f 9A A3 30 00
```

9. NNMi 管理サーバー上でovstart コマンドを実行して NNMi を起動し，<path to file>file.txt ファイルを読み込む。

10. Boot.log ファイルを開き，NNMi がファイルを読み込んだことを確認する。

手順 8.で準備したファイルが正しく読み込まれている場合には，次のようなメッセージが記録されます。

```
2012-10-17 14:44:44.876 情報 [NnmTrapService] Start: Populate
engineIDs from file
2012-10-17 14:45:08.017 情報 [SnmpV3EngineIdCachePopulator]
Successfully loaded 3 V3 Engine IDs from file /temp/patch2/
v3hosts.txt
```

ノードから有効な構成へのマッピングが失敗した場合には，次のようなメッセージが記録されます。

```
2012-10-17 14:45:03.485 警告 [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表示された場合は，このノードの［設定］＞［通信の設定］を調整する必要があります。

キャッシュと<path to file>file.txt ファイルからエントリを削除する必要がある場合は，<path to file>file.txt に変更を加えてエントリを削除してから，次のように NNMi を再起動します。

a NNMi 管理サーバー上でovstop コマンドを実行します。

b NNMi 管理サーバー上でovstart コマンドを実行します。

19.10 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

NNMi のデフォルト構成を使用しているときは，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オリジナルトラップアドレスが表示されないことがあります。管理者は，オリジナルトラップアドレスを特定するように NNMi を構成できます。

NNMi には，cia.originaladdress というカスタムインシデント属性があります。

NNMi は，com.hp.nnm.trapd.useUdpHeaderIpAddress プロパティと組み合わせて，cia.originaladdress 属性の意味を特定します。com.hp.nnm.trapd.useUdpHeaderIpAddress パラメータの値のデフォルトはfalse なので，NNMi は通常cia.originaladdress 属性を無視します。
com.hp.nnm.trapd.useUdpHeaderIpAddress の値をtrue に設定すると，cia.originaladdress 属性から SNMP エージェントアドレスの値がわかります。

UDP ヘッダーアドレスを NNMi でのソースとして使用して，さらに管理対象デバイスの実際の SNMP アドレスへアクセスする必要があるときは，com.hp.nnm.trapd.useUdpHeaderIpAddress の値をtrue に設定すると便利です。

com.hp.nnm.trapd.useUdpHeaderIpAddress 属性がfalse（デフォルトの設定）のときには，
cia.originaladdress およびcia.address 属性の両方が同じ値を含みます。

cia.originaladdress の値を使用してオリジナルトラップアドレスを特定するように NNMi を構成するには，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
   - UNIX：$NNM_PROPS/nms-jboss.properties
2. 次の行を含んでいるテキストブロックを検索する。

```
#!com.hp.nnm.trapd.useUdpHeaderIpAddress=false
```

3. 次のように行のコメントを解除して，編集する。

```
com.hp.nnm.trapd.useUdpHeaderIpAddress=true
```

4. 変更を保存する。
5. NNMi を再起動する。
   a ovstop
   b 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。

6. 手順 1.から手順 5.までを完了すると，NNMi はcia.originaladdress の値を使用してオリジナルトラップアドレスを特定する。

19.11　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
   - UNIX：$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. NNMi を再起動する。

```
ovstop
ovstart
```

参考

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。

19.12　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

管理者は，ネットワークから NNMi への HTTP アクセスを無効にできます。

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。

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
   - UNIX：$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. NNMi 管理サーバーを再起動する。
   a　NNMi 管理サーバーでovstop コマンドを実行します。
   b　NNMi 管理サーバーでovstart コマンドを実行します。

19.13　厳格に SNMPv3 インフォームを処理するように NNMi を構成する

厳格に SNMPv3 インフォームを処理するように NNMi を構成できます。新しいプロパティを構成すると，NNMi は厳格に SNMPv3 インフォームを処理できるようになります。NNMi は，トラップ転送設定で構成されたクレデンシャルに一致しないクレデンシャルを持つ SNMPv3 インフォームを処理しません。この構成では，NNMi 通信の設定画面でノードに対して構成された認証またはプライバシが無視されます。

NNMi は，（この新しいプロパティを持つ）SNMPv3 インフォームを検証するときとは異なる方法で SNMPv3 トラップを検証します。SNMP トラップの場合，NNMi はトポロジ内のノードの監視に現在使用されている通信の設定を使用します。

新しいプロパティを構成するには，次のようにします。

1. 次のファイルを編集する。
   - Windows
     `%NNM_DATA_DIR%¥shared¥nnm¥conf¥props¥nms-communication.properties`
   - UNIX
     `$NNM_DATA_DIR/shared/nnm/conf/props/nms-communication.properties`
2. 次の行を追加する。

```
com.hp.ov.nms.comm.snmp.enforcestrictv3traps=true
```

3. ファイルを保存する。
4. NNMi 管理サーバーを再起動する。
   a　NNMi 管理サーバー上で`ovstop` コマンドを実行します。
   b　NNMi 管理サーバー上で`ovstart` コマンドを実行します。

   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。

   構成したばかりのプロパティが誤っているか，`false` に設定された場合，NNMi は SNMPv3 インフォームをトラップ転送設定で設定された構成と照合して検証しません（この機能を追加する前の NNMi の動作）。NNMi は，拒否された SNMPv3 インフォームおよびトラップに関するメッセージを`nnm-trace*.log` ファイルに記録します。

19.14 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。varbind リスト内のvarbind の位置は，OID が SNMPv2 仕様に従ってリスト内に位置付けられていることを意味します。OID がトラップパラメータとして使用される場合は，OID が特定の MIB にリストされていることを意味します。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。

例えば，NNMi がvarbind が次のような位置にあるトラップを受信したとします。

```
Varbind 1: .1.3.6.1.2.1.1.3.0 (sysUpTime)
Varbind 2: .1.3.6.1.6.3.1.1.4.1.0 (snmpTrapOID)
Varbind 3: .1.3.6.1.2.1.2.2.1.1.92
Varbind 4: .1.3.6.1.4.1.11.2.17.20.20.1
Varbind 5: .1.3.6.1.4.1.11.2.17.20.20.2
Varbind 6: .1.3.6.1.4.1.11.2.17.20.20.3
Varbind 7: .1.3.6.1.4.1.11.2.17.20.20.4
Varbind 8: .1.3.6.1.2.1.1.3.0 (sysUpTime)
Varbind 9: .1.3.6.1.6.3.1.1.4.1.0 (snmpTrapOID)
Varbind 10: .1.3.6.1.4.1.11.2.17.20.20.3
Varbind 11: .1.3.6.1.4.1.11.2.17.20.20.4
```

NNMi 10-10 より前では，NNMi はトラップ 1 とトラップ 2 の両方にあるすべてのsysUpTime およびsnmpTrapOID のvarbind（下線部分）を削除します。これを次に示します。

```
Varbind 1: .1.3.6.1.2.1.1.3.0 (sysUpTime)
Varbind 2: .1.3.6.1.6.3.1.1.4.1.0 (snmpTrapOID)
Varbind 3: .1.3.6.1.2.1.2.2.1.1.92
Varbind 4: .1.3.6.1.4.1.11.2.17.20.20.1
Varbind 5: .1.3.6.1.4.1.11.2.17.20.20.2
Varbind 6: .1.3.6.1.4.1.11.2.17.20.20.3
Varbind 7: .1.3.6.1.4.1.11.2.17.20.20.4
Varbind 8: .1.3.6.1.2.1.1.3.0 (sysUpTime)
Varbind 9: .1.3.6.1.6.3.1.1.4.1.0 (snmpTrapOID)
Varbind 10: .1.3.6.1.4.1.11.2.17.20.20.3
Varbind 11: .1.3.6.1.4.1.11.2.17.20.20.4
```

NNMi 10-10 からは，NNMi は，次のように，最初と 2 番目の varbind（下線部分）位置にない sysUpTime および snmpTrapOID の varbind を保持します。

```
Varbind 1: .1.3.6.1.2.1.1.3.0 (sysUpTime)
Varbind 2: .1.3.6.1.6.3.1.1.4.1.0 (snmpTrapOID)
Varbind 3: .1.3.6.1.2.1.2.2.1.1.92
Varbind 4: .1.3.6.1.4.1.11.2.17.20.20.1
Varbind 5: .1.3.6.1.4.1.11.2.17.20.20.2
Varbind 6: .1.3.6.1.4.1.11.2.17.20.20.3
Varbind 7: .1.3.6.1.4.1.11.2.17.20.20.4
Varbind 8: .1.3.6.1.2.1.1.3.0 (sysUpTime)
Varbind 9: .1.3.6.1.6.3.1.1.4.1.0 (snmpTrapOID)
Varbind 10: .1.3.6.1.4.1.11.2.17.20.20.3
Varbind 11: .1.3.6.1.4.1.11.2.17.20.20.4
```

NNMi 10-10 までと同じ動作にしたい場合は，com.hp.nnm.events.preserveOldVarbindListOrder プロパティをtrue に設定してください。

NNMi 10-10 まで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
   - UNIX：$NNM_PROPS/nms-jboss.properties
2. 次の行を含んでいるテキストブロックを検索する。

   ```
   #!com.hp.nnm.events.preserveOldvarbindListOrder=false
   ```
3. 次のように行のコメントを解除して，編集する。

   ```
   com.hp.nnm.events.preserveOldvarbindListOrder=true
   ```
4. 変更を保存する。
5. NNMi を再起動する。
   a NNMi 管理サーバー上でovstop コマンドを実行します。
   b NNMi 管理サーバー上でovstart コマンドを実行します。

19.15　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。

SNMP トラップインシデントの自動トリム機能はデフォルトでは無効になっています。この機能を有効にすると，NNMi は古い SNMP トラップインシデントをデータベースから削除します。

注意事項

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。

19.15.1　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しない場合）

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 60,000 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，30,000 個の SNMP トラップ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。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 の場合：%NNM_PROPS%¥nms-jboss.properties
   - UNIX の場合：$NNM_PROPS/nms-jboss.properties
2. 次の行を探す。

```
#!com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=50
```

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=60
```

4. 次の行を探す。

```
#!com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=25
```

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=50
```

6. 次の行を探す。

```
#!com.hp.nnm.events.snmpTrapAutoTrimSetting=Disabled
```

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
```

8. 編集したファイルを保存してから，次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

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 60,000 を超えた場合に，次の式によって 30,000 個の SNMP トラップインシデントが削除されます。

```
com.hp.nnm.events.snmpTrapAutoTrimStartPercentage X
com.hp.nnm.events.snmpTrapMaxStoreLimit X
com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete
```

19.15.2 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 80,000 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，60,000 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。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 の場合：%NNM_PROPS%¥nms-jboss.properties
   - UNIX の場合：$NNM_PROPS/nms-jboss.properties
2. 次の行を探す。

```
#!com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=50
```

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=80
```

4. 次の行を探す。

```
#!com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=25
```

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=75
```

6. 次の行を探す。

```
#!com.hp.nnm.events.snmpTrapAutoTrimSetting=Disabled
```

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
```

8. 編集したファイルを保存してから，次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

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 80,000 を超えた場合に，次の式によって 60,000 個の SNMP トラップインシデントが削除されます。

```
com.hp.nnm.events.snmpTrapAutoTrimStartPercentage X
com.hp.nnm.events.snmpTrapMaxStoreLimit X
com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete
```

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 の場合：%NNM_TMP%¥incidentArchive.”*<日付>*”.csv.gz
- UNIX の場合：$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。

19.15.3 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。

注意事項

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 の場合：%NNM_PROPS%¥nms-jboss.properties
   - UNIX の場合：$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. 編集したファイルを保存してから，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 の場合：%NNM_PROPS%¥nms-jboss.properties
   - UNIX の場合：$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 の場合：%NNM_PROPS%¥nms-jboss.properties
   - UNIX の場合：$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，次のように修正し，ファイルを保存する。

```
com.hp.nnm.events.snmpTrapMaxStoreLimit=200000
```

7. 次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

19.15.4 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。

19.15.5 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 の場合：%NNM_PROPS%¥nms-jboss.properties
   - UNIX の場合：$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. ファイルを保存してから，次のコマンドを実行して NNMi を再起動する。

```
ovstop
ovstart
```

19.16 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
   - UNIX：$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. NNMi 管理サーバーを再起動する。

a NNMi 管理サーバーでovstop コマンドを実行します。

b NNMi 管理サーバーでovstart コマンドを実行します。

19.16.1 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後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。

19.17　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
   - UNIX：$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. NNMi 管理サーバーを再起動する。
   a　NNMi 管理サーバーでovstop コマンドを実行します。
   b　NNMi 管理サーバーでovstart コマンドを実行します。

19.18 NNMi 自己監視

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
   - UNIX：$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. NNMi を再起動する。
   a NNMi 管理サーバーでovstop コマンドを実行します。
   b NNMi 管理サーバーでovstart コマンドを実行します。

19.19　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，「19.19.1　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，「19.19.1　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。

19.19.1　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`
   - UNIX：`$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. NNMi 管理サーバーを再起動する。

   a NNMi 管理サーバーでovstop コマンドを実行します。

   b NNMi 管理サーバーでovstart コマンドを実行します。

注意事項

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。

19.20　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
   - UNIX：$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. NNMi を再起動する。
   a　NNMi 管理サーバー上でovstop コマンドを実行します。
   b　NNMi 管理サーバー上でovstart コマンドを実行します。

20 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。

20.1 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

20.2　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
```

「18.3.2　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
```

注意事項

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，「20.3　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。

異なるシステムへのリストアに関しては，ソースシステムへのリストア中に古い SSL 証明書が，サーバー上に（nnm.keystore.backup とnnm.truststore.backup として）保存されています。古い証明書と新しい証明書はリストア中にマージされます。

20.3 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。

注意事項

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。

参考

- 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。

20.4　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。

20.5　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

注意事項

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，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 管理サーバーのホスト名，ドメイン名，またはその両方を変更するには，次の手順を実行します。

1. システム名を変更する。

   必要に応じて，システムを再起動します。

2. NNMi を停止する。

```
ovstop
```

3. 新規 NNMi パブリックキー証明書を作成する。

   この NNMi 管理サーバーの新規証明書をnnm.keystore ファイルに作成します。次回，ovjboss プロセスが正常に起動すると，NNMi によって新規証明書を使用するデータベースへのアクセスが更新されます。

   a　NNMi 証明書が含まれるディレクトリに移動します。

   - Windows：%NnmDataDir%shared¥nnm¥certificates
   - UNIX：$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。

   b　次のコマンドを実行し，キーストアに新規パブリック/プライベートキーペア（証明書）を生成します。

   Windows

```
%NnmInstallDir%¥nonOV¥jdk¥nnm¥bin¥keytool.exe -genkey ¥
-alias "<unique_alias>" -keyalg rsa -keysize 2048 ¥
-dname "cn=<hostname>, dc=<domain_name_by_parts>" ¥
-keypass "nnmkeypass" -validity 36500 ¥
-keystore nnm.keystore -storepass "nnmkeypass"
```

   UNIX

```
$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -genkey ¥
-alias "<unique_alias>" -keyalg rsa -keysize 2048 ¥
-dname "cn=<hostname>, dc=<domain_name_by_parts>" ¥
-keypass "nnmkeypass" -validity 36500 ¥
-keystore nnm.keystore -storepass "nnmkeypass"
```

（凡例）

行の最後の¥は，行が続いていることを示します。

<unique_alias>を NNMi 管理サーバーの新規ホスト名などの一意値（例：newnnmi）に置き換えます。

<hostname>を NNMi 管理サーバーの新規完全修飾ドメイン名（例：newnnmi.servers.example.com）に置き換えます。

dc=<domain_name_by_parts>を NNMi 管理サーバーがある新規ドメインの各コンポーネントに置き換えます。例えば，NNMi 管理サーバーnewnnmi.servers.example.com の場合は，次を指定します。

dc=servers, dc=example, dc=com

keytool コマンドの詳細については，java.sun.com で「鍵と証明書の管理ツール」を検索してください。

4. NNMi 管理サーバーの完全修飾ドメイン名を変更する。

NNMi で NNMi 管理サーバーの新規完全修飾ドメイン名を使用するように設定するには，nnmsetofficialfqdn.ovpl コマンドを使用します。

（例）

```
nnmsetofficialfqdn.ovpl newnnmi.servers.example.com
```

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。

5. 新しい証明書で HTTPS 設定を更新する。

新しい証明書で HTTPS 設定を更新するには，次の手順を実施します。

a　次のファイルを編集します。

- Windows：%NNM_CONF%¥nnm¥props¥nms-local.properties
- UNIX：$NNM_CONF/nnm/props/nms-local.properties

b　手順 3.の新規 NNMi パブリックキー証明書に使用した<unique_alias>に一致するように com.hp.ov.nms.ssl.KEY_ALIAS 変数を変更します。

CA 証明書を使用する場合は，「8.2　認証機関証明書を生成する」の指示に従い，com.hp.ov.nms.ssl.KEY_ALIAS 変数を変更します。

6. NNMi を起動する。

```
ovstart
```

21 NNMi セキュリティ

セキュリティについて説明します。

21.1　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，UNIX 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。

21.2 NNMi が ovjboss バージョン番号を報告しないように設定する

「Error 404」または「Not Found」のエラーメッセージは，HTTP の標準的な応答であり，クライアントはサーバーと通信することができたが，サーバーは要求されたコンテンツを見つけることができなかったことを示します。NNMi 管理サーバーが，ovjboss 情報を含む「Error 404」メッセージを生成する場合があります。

NNMi 管理サーバーが ovjboss 情報を報告しないようにするには，次の手順を実行します。

1. NNMi 管理サーバーでovstop コマンドを実行する。
2. 次に示すディレクトリ以外の場所にserver.xml ファイルを保存する。
   - Windows：%NnmInstallDir%¥nmsas¥common¥deploy¥jbossweb.sar¥server.xml
   - UNIX：$NnmInstallDir/nmsas/common/deploy/jbossweb.sar/server.xml
3. 次に示すファイルを編集する。
   - Windows：%NnmInstallDir%¥nmsas¥common¥deploy¥jbossweb.sar¥server.xml
   - UNIX：$NnmInstallDir/nmsas/common/deploy/jbossweb.sar/server.xml
4. ファイルの中から次の行を探す。

   ```
   <Host name="localhost" ...
   ```

5. 最後の>（大なり）の記号の前に次の属性を追加する。

   ```
   errorReportValveClass="com.hp.ov.nms.as.server.tomcat.NmsErrorReportValve"
   ```

   （例）

   ```
   <Host name="localhost" workDir="${nmsas.product.dir.workDir}/web"
   errorReportValveClass="com.hp.ov.nms.as.server.tomcat.NmsErrorReportValve">
   ```

6. NNMi 管理サーバーで ovstart コマンドを実行する。
7. NNMi 管理サーバーをテストし，ovjboss 情報を報告する「Error 404」エラーメッセージが生成されないことを確認する。

22 バージョン 9・10-00・10-10 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。

バージョン 8 以前の NNM から NNMi への移行については，「24.　バージョン 8 以前の NNM からの移行」を参照してください。

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 10-00，および 10-10 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 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。

22.1 NNMi 管理サーバーをバージョンアップする

22.1.1 バージョン 10-00 および 10-10 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 10-00 および 10-10 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。

参考

NNMi 管理サーバーをバージョンアップする前に，マニュアル「*JP1/Cm2/Network Node Manager i　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. マニュアル「*JP1/Cm2/Network Node Manager i　インストールガイド*」の手順に従って，NNMi 10-50 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。

22.1.2 バージョン 9 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 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 10-00 の「*JP1/Cm2/Network Node Manager i インストールガイド*」，「*JP1/Cm2/Network Node Manager i セットアップガイド*」，「リリースノート」の手順に従って，バージョン 9 で実行中の NNMi 管理サーバーをバージョン 10-00 の NNMi 管理サーバーにバージョンアップする。

2. 「22.1.1　バージョン 10-00 および 10-10 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 10-00 で実行中の NNMi 管理サーバーをバージョンアップする。

22.2　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 10-50 にバージョンアップする手順について説明します。

> **参考**
>
> NNMi 管理サーバーをバージョンアップする前に，マニュアル「*JP1/Cm2/Network Node Manager i　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 10-00 および 10-10 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. マニュアル「*JP1/Cm2/Network Node Manager i　インストールガイド*」の手順に従って，ソースの NNMi 管理サーバー上に NNMi 10-50 をインストールする。

3. NNMi 10-50 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 10-50 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. マニュアル「*JP1/Cm2/Network Node Manager i　インストールガイド*」の手順に従って，ターゲットの NNMi 管理サーバー上に NNMi 10-50 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バージョン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。

22.3 NNMi 10-00 および 10-10 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

22.3.1 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 10-50 が実行されているグローバルマネージャに接続された，NNMi 10-10 以前が実行されているリージョナルマネージャはサポートしていません。グローバルマネージャとリージョナルマネージャの両方で，同一バージョンの NNMi を実行する必要があります。

22.3.2 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 10-50 にアップグレードする場合，グローバルネットワークマネージャとリージョナルマネージャ間の接続は，グローバルネットワークマネージャとリージョナルマネージャの両方が NNMi 10-50 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を NNMi 10-50 にアップグレードし，正しく動作することを確認する。
   リージョナルマネージャのアップグレード中，グローバルマネージャは切断されたままになります。
2. グローバルマネージャを NNMi 10-50 にアップグレードする。

注意事項

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。

22.4 アプリケーションフェイルオーバー構成の NNMi 10-50 へのアップグレード

22.4.1 アプリケーションフェイルオーバー構成の NNMi 10-00 および 10-10 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 10-00 および 10-10 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 10-00 および 10-10 からのアップグレード

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，「2.2　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。

   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します。詳細については，「18.2.2　バックアップ領域」を参照してください。

2. 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 6.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。

3. 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。

4. 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。

5. 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

     UNIX：$NNM_SHARED_CONF/props/nms-cluster.properties
   - b　com.hp.ov.nms.cluster.name パラメータをコメントにします。
   - c　変更を保存します。

6. 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。

7. マニュアル「*JP1/Cm2/Network Node Manager i インストールガイド*」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 10-50 にアップグレードする。

   以前のアクティブ NNMi 管理サーバーが NNMi 10-10 以前を実行し，以前のスタンバイ NNMi 管理サーバーが NNMi 10-50 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 10-50 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。

8. 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。

9. 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。

10. 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

      UNIX：$NNM_SHARED_CONF/props/nms-cluster.properties
    - b　com.hp.ov.nms.cluster.name パラメータをコメントにします。

11. マニュアル「JP1/Cm2/Network Node Manager i インストールガイド」の指示に従い，以前のアクティブ NNMi 管理サーバーを NNMi 10-50 にアップグレードする。

    2 つのサーバーで NNMi 10-50 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。

12. 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

      UNIX：$NNM_SHARED_CONF/props/nms-cluster.properties
    - c　10-00 からアップグレードした場合，com.hp.ov.nms.cluster.name パラメータの値を入力します。

      10-00 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 パラメータのコメントを解除します。
    - e　変更を保存します。

13. 10-00 からアップグレードした場合，以前のアクティブ NNMi 管理サーバーでクラスタ通信に使用する NIC を設定する。

    詳細については，「16.3.3　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。

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 8.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
      UNIX：$NNM_SHARED_CONF/props/nms-cluster.properties
    - c　10-00 からアップグレードした場合，com.hp.ov.nms.cluster.name パラメータの値を入力します。
    - d　com.hp.ov.nms.cluster.name パラメータのコメントを解除します。
    - e　変更を保存します。
17. 10-00 からアップグレードした場合，以前のスタンバイ NNMi 管理サーバーでクラスタ通信に使用する NIC を設定する。
    詳細については，「16.3.3　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。
18. 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 NNMi10-50 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，「2.2　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，「18.2.2　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

     UNIX：$NNM_SHARED_CONF/props/nms-cluster.properties

- b　com.hp.ov.nms.cluster.name パラメータ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
     UNIX：$NNM_SHARED_CONF/props/nms-cluster.properties
   - b　com.hp.ov.nms.cluster.name パラメータ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，「2.2　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，「18.2.2　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

    UNIX：$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
```

23 バージョン 8 以前の NNM との比較

この章では，バージョン 8 以前の NNM と NNMi との重要な違いについて説明します。以前バージョンを使用していた方は，この章を参照しながら NNMi の計画を立てたり設定したりしてください。NNMi を初めてお使いになる方は，この章を読む必要はありません。

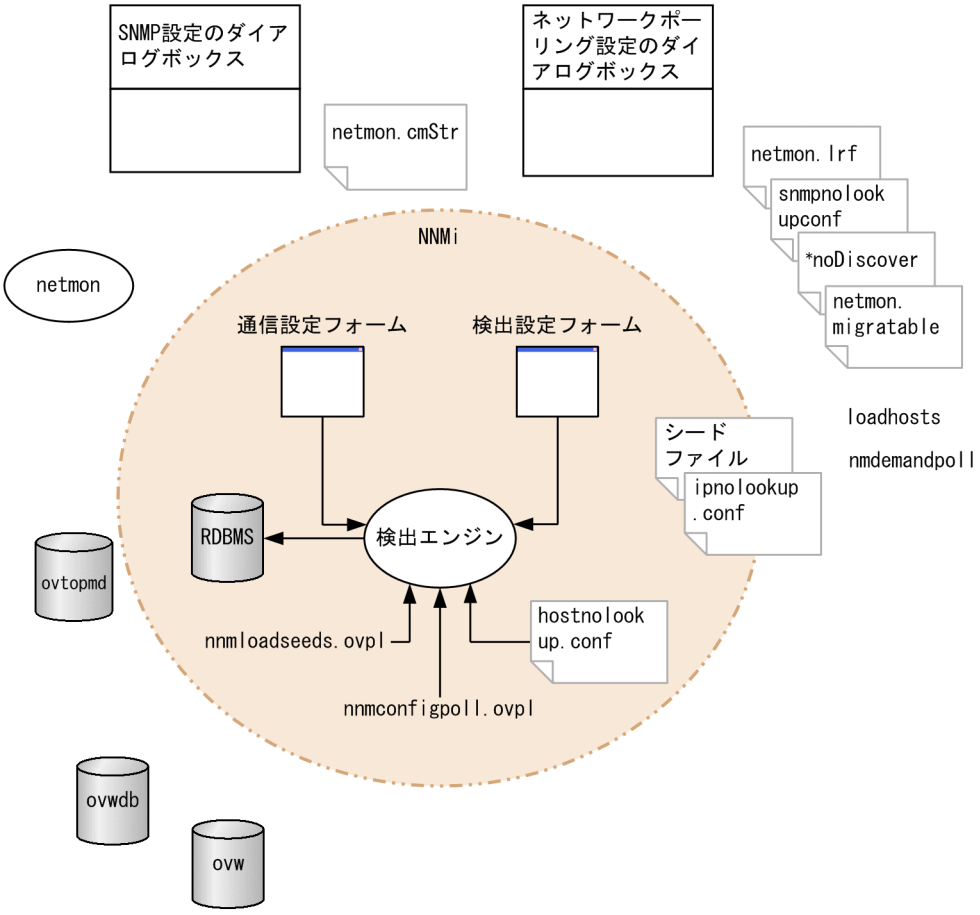
23.1　ネットワーク検出

検出は，データベースに追加されているネットワークの要素（デバイス，ノード，およびこれらのコンポーネント）に対して行われます。NNMi では，「インベントリ検出」とは新しいノードを探し出すことであり，「レイヤー 2 検出」とは接続性モデリングを指します。

NNM のデフォルトでは，起動すると自身のループバックアドレスをシードとして使用し，直接接続しているネットワークの自動検出を（自身の IP アドレスおよびサブネットマスクに基づいて）開始していました。NNMi では，最初から管理者制御が可能です。NNMi 自動検出では，検出を行う前に，検出領域を IP アドレス範囲に基づいて定義し，少なくとも 1 つのシードデバイス（通常はルーター）を指定します。

次の図の中央には，NNMi で検出を設定するために使用するツール，ファイルおよびコマンドを示しています。周囲には，NNM のツール，ファイルおよびコマンド等を示しています。

図 23–1　検出の設定要素



23.1.1　検出の重要概念

ここでは，NNM から NNMi への主な変更点を簡単に説明します。NNMi 検出についての詳細は，NNMi ヘルプの「*ネットワークの検出*」を参照してください。

- すべての情報を 1 つのリレーショナルデータベース内に保管します。
- 設定が容易な統合検出エンジンを使用します。
- スパイラル検出プロセスによって，ネットワークに変化が生じた際のトポロジ情報の継続的な更新ができます。定期的な再検出間隔よりトポロジの変化（インベントリとレイヤー 2 の両方）を，頻繁に検出できます。
- すべての検出対象ノードは，管理モード（管理対象，管理除外，またはサービス停止）にかかわらず，ライセンス限度に対してカウントされます。ライセンス限度を超えるノードは検出できません。
- 自動検出は，NNMi と NNM では同じ意味を持っていますが，設定アプローチは異なります。

  －NNMi では，自動検出境界を定義し，少なくとも 1 つの IP アドレスシードを指定してから，検出を実行させます。

  －NNMi 自動検出では，管理が容易な拡大式モデルを使用します。NNMi 自動検出では，指定された境界内のすべてのルーター，スイッチおよびサブネットを見つけ出して管理します。NNMi で検出して管理する追加デバイスタイプを指定します。

  参考

  デフォルトでは，SNMP 以外のノードは検出されません。
- シード検出は，NNMi と NNM では同じ意味を持っていますが，設定アプローチは異なります。

  －NNMi では，検出シードをユーザーインタフェースで指定します。

  －NNM のシードファイルを，NNMi でそのまま使用できます。

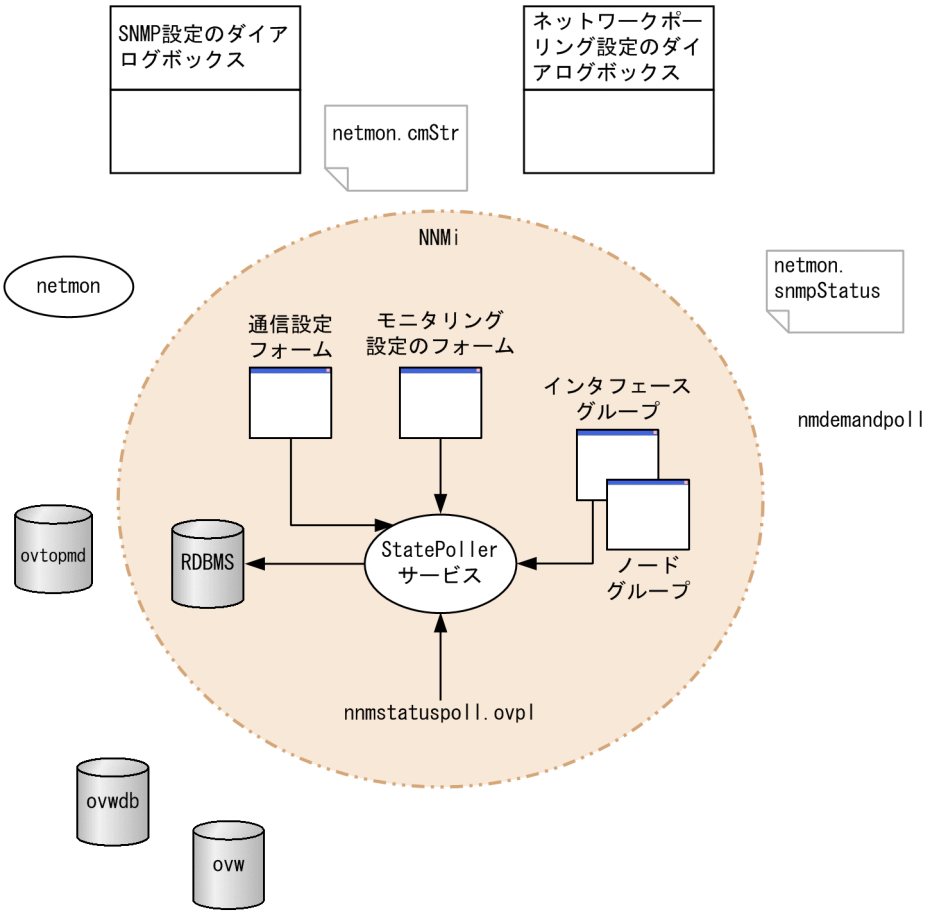
  －NNMi の`nnmloadseeds.ovpl` コマンドは，NNM の`loadhosts` コマンドに代わるコマンドです。
- NNMi 設定ポーリング（`nnmconfigpoll.ovpl`）は，デバイス設定情報を決定するための NNM のデマンドポーリング（`nmdemandpoll`）に代わるものです。

23.2　ステータス監視

ステータス監視を行うことによって，故障が起こり得るデバイスやコンポーネントに関して，最新のネットワークを可視化できます。ある構成要素のポーリングに失敗すると，NNMi は原因を調査して，根本原因アラームをインシデントブラウザに送出します。

次の図の中央に，ステータス監視を設定するために使用するツール，ファイルおよびコマンドを示しています。周囲には，NNM のツール，ファイルおよびコマンドなどを示しています。

図 23–2　監視の設定要素



23.2.1　ステータス監視の重要概念

ここでは，NNM から NNMi への主な変更点を簡単に説明します。NNMi ステータス監視に関する詳細は，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。

- 設定は，ユーザーインタフェースを通じて完了します。
- NNMi ノードグループおよびインタフェースグループは，トポロジフィルタに代わるものです。
  －グループは，定義済みの属性でだけフィルタリングできます。
  －グループをブール演算子で連結できません。

－ノードグループは，sysObjectId ワイルドカードを使用する代わりにデバイスフィルターを使用します。

－インタフェースグループを，ホストするノードのグループおよびインタフェースタイプに基づいて制限できます。

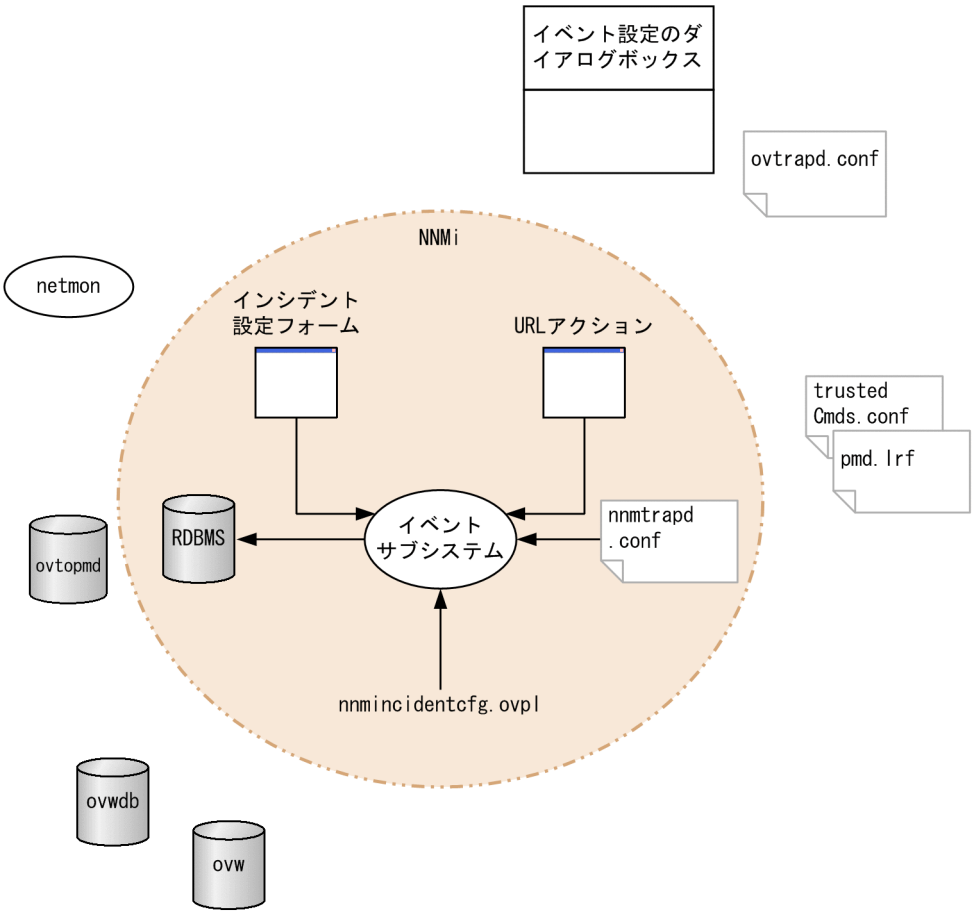
- 広範な制御機能によって，不要なインタフェースの除外が容易です。
- 監視設定は，（1）インタフェースの設定，（2）ノードの設定，（3）デフォルト設定のように，固有性の高いものから一般性的なものへ，順に適合します。
- 監視の動作をシステム全体で変更するには，すべての設定を全レベルで変更します。
- NNMi のステータスポーリング（**[アクション]** ＞ **[ポーリング]** ＞ **[ステータスのポーリング]** または`nnmstatuspoll.ovpl`）は，デバイスのステータスを判定するための NNM のデマンドポーリング（`nmdemandpoll`）に代わるものです。
- デフォルトでは，NNMi がポーリングするインタフェースは，レイヤー 2 接続を通じて別の既知のインタフェースに接続しているインタフェースだけです。接続していないインタフェースのポーリングと IP アドレスをホストしているインタフェースのポーリングを有効にできます。

23.3　イベント監視のカスタマイズ

NNMi にはインシデントビューという 1 つの中心となる場所があり，そこで管理イベント，および SNMP トラップを見ることができます。どの SNMP トラップをインシデントとして表示するかを制御してください。

次の図の中央に，NNMi でのイベント監視を設定するために使用するツール，ファイルおよびコマンドを示しています。周囲には，NNM のツール，ファイルおよびコマンドなどを示しています。

図 23–3　イベント監視の設定要素



23.3.1　イベント監視の重要概念

ここでは，NNM から NNMi への主な変更点を簡単に説明します。NNMi インシデントに関する詳細は，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。

- イベントサブシステムがプロセス間通信で使用されていません。また，イベントのボリュームが大きく削減されています。管理者は，それぞれの IPC メッセージを表示するかログするかを設定する必要がなくなりました。

- 受信設定したトラップだけを受信します。設定されていないトラップは，フィルタリングされてイベントパイプラインから除かれます。
- 受信するすべてのトラップを表示します。
- NNMi イベントサブシステムプロセスのトラップフィルタは，［インシデントの設定］フォームでの選択内容に基づいて設定されます。
- NNMi のnnmincidentcfg.ovpl コマンドは，指定された MIB モジュールのトラップ定義だけをロードします。
- イベントパイプラインで生じるペアワイズ，レート，および重複削除の相関を提供します（NNMi には，イベント相関システム（ECS）は含まれません）。
- インシデントのライフサイクルで生じるアクションを設定できます。あらゆるスクリプト，実行ファイル，または Jython アクションをアクションとすることができます。

24 バージョン 8 以前の NNM からの移行

この章では，NNM から NNMi への基本移行方法について説明します。この方法は，多くのユーザーに役立ちます。この章では，高度な移行のトピックまたはカスタマイズについては説明しません。

この章では，次の製品命名規約を使用します。

・NNM は，バージョン 8 以前の NNM のことです。

・NNMi は，バージョン 9 以降の NNMi のことです。

この章では，次の状態であることを想定しています。

・マニュアル「*JP1/Cm2/Network Node Manager i インストールガイド*」の指示に従って NNMi をインストール済みである。

・NNMi ヘルプとこのマニュアルの導入情報に説明してある概念，および NNMi 機能を全般的に理解している。

・NNMi コンソールの使用法を理解している。

24.1　移行手順

24.1.1　新しい NNM システム

NNM は，ソフトウェアの数世代にわたり，さまざまなネットワーク環境で使用されてきました。ルーター中心の世界でバージョン 5 以前の NNM からのユーザーは，現在のネットワーク構造には実際には適合しない，大きな荷物を抱えているでしょう。NNM システムが 2 年以上前のものである場合は，この機会を利用して新しいシステムを開始することをお勧めします。現在のネットワークをどのように管理するか改めて評価することで，NNM と比較して，オーバーヘッドの大幅な削減と，操作の効率化を実現する可能性があります。

NNMi を新たにインストールして使用し始める場合は，マニュアル「*JP1/Cm2/Network Node Manager i インストールガイド*」の指示に従って NNMi をインストールしてください。次に，このマニュアルのほかの章に説明してある導入作業を検討してください。その場合は，この章を読む必要はありません。

24.1.2　フェーズを分けて移行する

組織によっては，新規に構築するより，フェーズに分けた移行作業の方が，うまく機能する場合があります。このような組織では，新しい NNMi システムで，既存の NNM システムを完全に再現し，置き換えることを必要とするでしょう。移行の方法は多数ありますが，次のフェーズをお勧めします。

- 「24.2　フェーズ 1：SNMP 情報を移行する」
  使用中の環境の SNMP アクセス情報で NNMi を設定します。
- 「24.3　フェーズ 2：検出を移行する」
  NNM がオブジェクトを（自動）検出したのと似たような方法で，NNM が検出したオブジェクトを NNMi が検出するように設定します。
- 「24.4　フェーズ 3：ステータスモニタリングを移行する」
  使用中の環境に最も適切なステータスポーリング間隔とプロトコルを設定します。
- 「24.5　フェーズ 4：イベント設定とイベント削減を移行する」
  NNM で設定したように，イベントの重要度，カテゴリ，メッセージを表示し，自動アクションを実行するように NNMi を設定します。また，重複削除，レートのカウント，PairWise のキャンセルも設定する必要があるかもしれません。
- フェーズ 5：グラフィカルな視覚化を移行
  NNM のロケーションサブマップ，インターネットサブマップおよびセグメントサブマップの階層構造を NNMi のノードグループの設定として移行します。移行方法については，リリースノートを参照してください。

「表 24-1　移行の範囲」に，移行範囲について，最もシンプルな方法と，最も詳細で綿密な方法の概要を示します。

- 最もシンプルな方法では，環境に特有の情報は NNM からインポートし，そのほかの設定は，NNM から改善された NNMi のデフォルト値を使用します。
- 最も詳細で綿密な方法としては，NNM 設定を詳しく調べ，この設定を NNMi で再現します。

この章の残りの部分では，NNM の設定を NNMi に移行するプロセスを，順番に説明していきます。次に示す「NNM から収集」，「NNMi で再現」などの見出しは，特定の手順が移行プロセスのどの作業に当てはまるか示しています。

- 「NNM から収集」は，NNM 管理ステーションで行う作業を示します。
- 「NNMi で再現」は，NNMi 管理サーバーで行う作業を示します。
- 「NNMi での強化」は，追加項目として，NNMi 管理サーバーで行う作業を示します。移行プロセスの間，またはそのあといつでも強化できます。

幾つかのポイントでは，作業の難易度に応じて複数の方法を用意しています。

表 24–1　移行の範囲

フェーズ	最もシンプルな方法	最も詳細で綿密な方法
SNMP 情報	1. 現在使用中のコミュニティ文字列をすべてエクスポートします。 2. これらのコミュニティ文字列を NNMi にインポートします。NNMi がどのコミュニティ文字列がどのノードに一致するかを判断します。	1. 現在使用中のコミュニティ文字列をすべてエクスポートします。 2. エクスポートしたデータファイルを修正し，特定ノードのコミュニティ文字列として NNMi にインポートします。
検出	1. 検出された全ノードのリストをエクスポートします。 2. データファイルを変更し，ファイルの内容を，自動検出ルールのないシードとして NNMi にインポートします。	1. NNM と netmon がノードを検出する方法（シード，ロードホスト，フィルタ，そのほかのツール）を特定します。 2. シードおよび自動検出ルールを使用して，NNMi で可能な限り厳密にこの方法を再現します。
ステータスモニタリング	NNMi のデフォルト値は，ほとんどのユーザー要件に合うように改善されます。このデフォルト値を大幅に変更する必要はないので，改善されたデフォルト値で操作を開始します。	1. ノードの各グループについて，どのようなポーリング間隔とポーリングポリシーが，NNM および netmon で使用されているかを正確に調べます。 2. ポーリング間隔とポーリングポリシーを再現するように，NNMi のノードグループとインタフェースグループを作成します。
イベント設定とイベント削減	1. NNMi のデフォルト設定で開始します。 2. 管理対象デバイスのカスタムトラップの定義を追加します。 3. 必要に応じて，自動処理を追加します。	1. トラップとイベントの種類ごとに，何の NNM カスタマイズが行われたかを正確に調べます。 2. NNMi システム上で，一致するそれぞれのトラップとイベントの種類をカスタマイズします。

24.2　フェーズ 1：SNMP 情報を移行する

管理対象デバイスとの通信を確立するために NNMi が使う SNMP コミュニティ文字列情報を移行します。

NNM の設定に，名前解決から除外する IP アドレスまたはホスト名がある場合は，その情報を NNMi に移行します。

使用中のネットワークのカスタムデバイス用に NNMi デバイスプロファイルをカスタマイズします。

24.2.1　SNMP アクセスを設定する

NNMi 検出は，管理対象ノードの設定と接続に関する個別の情報を収集するため，それらノードに対する SNMP アクセスが必要です。SNMP は，ノードおよびそれに含まれるオブジェクトの稼働状態にアクセスするために，ステータスモニタリングの間も使用されます。

参考

NNM は，一致する領域の設定にリストされた順序で，コミュニティ文字列を 1 つずつ試してみて，利用可能なことを確認できた最初のコミュニティ文字列を使います。NNMi は，設定されたすべてのコミュニティ文字列を並行して試し，利用可能なことを確認できた最初のコミュニティ文字列を使います。利用可能な値が複数ある場合は，最も適したコミュニティ文字列を設定してください。

NNMから収集

NNM 管理ステーションには，使用中の環境の機器に SNMP がアクセスするための設定情報があります。

1. NNM SNMP 設定をエクスポートするには，次の操作の 1 つを実行する。
   - ユーザーインタフェースを開き，［オプション］ > ［SNMP 設定］を選択し，［エクスポート］をクリックします。ターゲットのファイル名にsnmpout.txt を指定します。
   - 次のコマンドを実行します。

     ```
     xnmsnmpconf -export > snmpout.txt
     ```

NNM SNMP 情報の例

出力は次の例のようなものになります。

```
10.2.126.75:public:*::::::
mytest57.mycorp.net:public:*::::::
127.0.0.1:public:*::::::
10.97.233.209:mycommstr:*::::::
mpls2950.mycorp.net:mycommstr:*::::::
```

mplsce04.mycorp.net:mycommstr:*::::::

..*.*:mycommstr:*:8:2:900:::

ターゲットファイルには，コロンで区切られた次のフィールドがあります。

target:community:proxy（*はプロキシでないことを示す）:timeout（1/10 秒単位）:retries:poll interval（秒単位）:port:set-community:

値の詳細情報を知るには次のコマンドを使います（ただし，インポートでは使わないでください）。

xnmsnmpconf -export -verbose

ovsnmp.conf ファイルフォーマットの詳細は，ovsnmp.conf のリファレンスページを参照してください。

2. 次のファイルで，設定されたコミュニティ文字列を確認する。

- Windows：%OV_CONF%¥netmon.cmstr
- UNIX：$OV_CONF/netmon.cmstr

NNMiで再現

コミュニティ文字列を NNMi に入力する方法を選択します。これらの各方法は，「NNM から収集」の手順 1.で作成した snmpout.txt ファイルの一意のコミュニティ文字列リストから開始します。

参考

［SNMP プロキシシステム］と［設定コミュニティ名］の設定エリアは移行できません。

(1) シンプルな方法

最もシンプルな方法としては，NNM コミュニティ文字列をすべて入力し，各デバイスに使う SNMP コミュニティ文字列を NNMi が解決できるようにします。コミュニティ文字列の検出はデフォルトで有効です。この機能によって迅速に移行できます。

1. ネットワークオペレーティングセンター（NOC）に，NNMi の最初の検出の間，認証エラーが発生することを予測するように通知する。

NOC の担当者は，その間，これらの認証エラーを無視できます。

2. 次の操作のうち 1 つを実行する。

- NNMi が使うフォーマットと一致するようにsnmpout.txt を変更します。次に，NNMi を使ってこれらの値をロードします。
- snmpout.txt ファイルをサンプルとして使用し，NNMi の入力ファイルを手作業で構築します。次に，NNMi を使ってこれらの値をロードします。
- 次の手順で，値を NNMi コンソールに入力します。

a snmpout.txt ファイルの一意のコミュニティ文字列値のリストを調べます。

－Windows：Excel でsnmpout.txt ファイルを開きます。データ行を選択してから，コラム B でソートします。

この例の場合は，次の 2 つの一意のコミュニティ文字列について考えます。

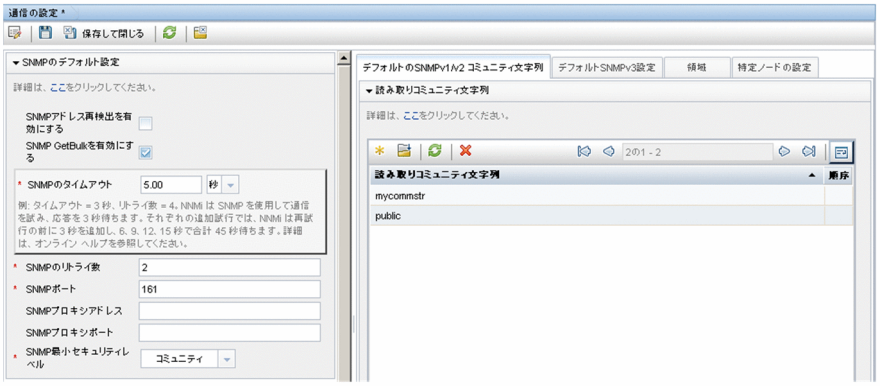
```
public
mycommstr
```

－UNIX：次のコマンドを実行します。

```
cut -f 2 -d ':' < snmpout.txt | sort -u
```

b　NNMi コンソールで，**[設定]** ワークスペースから **[通信の設定]** を選択します。**[デフォルトの SNMPv1/v2 コミュニティ文字列]** タブに一意の値をすべて入力します。

c　タイムアウト，リトライ数，およびポートを設定します。



(2)　修正したシンプルな方法

使用される IP 領域ごとのコミュニティ文字列をまとめます。領域ごとの値を NNMi コンソールで入力し，NNMi が各デバイスに使用する SNMP コミュニティ文字列を決定するようにします。前述の**シンプルな**方法よりも認証の失敗は少なくなります。

1. snmpout.txt ファイルで，NNM が使っている*IP 領域ごとの*一意の値のリストを調べる。
2. NNMi コンソールで，［設定］ワークスペースから［通信の設定］を選択する。
   IP 領域を作成してから，領域ごとにコミュニティ文字列を入力します。
3. タイムアウト，リトライ数，およびポートを設定する。

(3)　自動化された方法

snmpout.txt ファイルをnnmcommload.ovpl コマンドに必要なフォーマットに変換してから，各デバイスで使用中の個別のコミュニティ文字列をロードします。

1. NNMi ツールで使えるようsnmpout.txt ファイルを適合させるには，次の方法のうち 1 つを実行する。
   - エディタを使って NNMi に適切なファイルを作成します。結果は次のようなものになります。

     ```
     10.2.126.75,public
     mytest57.mycorp.net,public
     ```

```
127.0.0.1,public
10.97.233.209,mycommstr
mpls2950.mycorp.net,mycommstr
mplsce04.mycorp.net,mycommstr
```

- UNIX だけ：次のコマンドを実行します。

  ```
  awk 'BEGIN {FS = ":" };{printf"%s,%s¥n",$1,$2 }' ¥ <snmpout.txt> mysnmp.txt
  ```

  このコマンドはファイル内の個別のノードの設定にだけ有効です。範囲またはワイルドカードの設定は，手作業で削除します。

2. 次のコマンドを実行する。

   ```
   nnmcommload.ovpl -u username -p password -file mysnmp.txt
   ```

3. NNMi コンソールで，デフォルトのコミュニティ文字列，および IP 範囲用のコミュニティ文字列を設定する。

4. NNMi コンソールで，タイムアウト，リトライ数，ポートをすべて設定する。

(4) NNMi コンソールからの方法

NNMi コンソールで，［設定］ワークスペースから［通信の設定］を選択します。

snmpout.txt ファイルの設定された値を再現します。

NNMiでの強化

次の情報を使って，NNMi の通信アクセス設定を強化します。

- ホスト名ワイルドカード（IP 範囲より環境によく適合する場合）
- グローバルデフォルト，IP 範囲，および特定のノードに対する ICMP タイムアウトとリトライ数
- ネットワークの特定のエリアへの SNMP または ICMP のアクセスを有効化または無効化
- 特定のノードについて優先される管理アドレス

> **参考**
>
> NNM は，管理アドレスを選択するとき，最も小さいループバックアドレスを選択します。NNMi も最も小さいループバックアドレスを選択します。

24.2.2 名前解決を制限する

DNS（またはほかの名前解決）サービスの制限がわかっている場合は，NNM と NNMi にこれらのデバイスのルックアップを避けるよう指示できます。この作業がシステムに該当しない場合は，「24.2.1 SNMP アクセスを設定する」に進んでください。

NNMから収集

1. 次のファイルを確認し，NNM が「アドレスからホスト名への名前解決」から除外するアドレスを特定する。
   - Windows：%OV_CONF%¥ipnolookup.conf
   - UNIX：$OV_CONF/ipnolookup.conf
2. 次のコマンドを実行し，NNM が「名前からアドレスへの名前解決」から除外するホスト名を調べる。
   snmpnolookupconf dumpCache > snmpnolookup.out

   NNMiで再現
3. アドレスを手順 1.から次のファイルに追加する。
   - Windows：%NnmDataDir%shared¥nnm¥conf¥ipnolookup.conf
   - UNIX：$NnmDataDir/shared/nnm/conf/ipnolookup.conf
4. ホスト名を手順 2.から次のファイルに追加する。
   - Windows：%NnmDataDir%shared¥nnm¥conf¥hostnolookup.conf
   - UNIX：$NnmDataDir/shared/nnm/conf/hostnolookup.conf

   これらの設定ファイルのフォーマットについては，ipnolookup.conf とhostnolookup.conf のリファレンスページを参照してください。

   NNMiでの強化

   NNMi は検出の間だけルックアップを実行します。NNM 非ルックアップ設定を NNMi で再現すると，スパイラル検出の動作が自動的に改善されます。
5. NNMi では，表示する名前ラベルに，DNS ホスト名，IP アドレス，または MIB II sysName のどれかを選択して使用できる。次の手順で設定する。
   a　NNMi コンソールで，［設定］ワークスペースを開きます。
   b　［検出］＞［検出の設定］を選択します。
   c　［ノード名の解決］エリアでノード名優先を設定します。

24.2.3　デバイスプロファイルをカスタマイズする

NNM は，デバイスへの SNMP 通信によって，幾つかの設定情報を直接収集します。また，デバイスの**システムオブジェクト ID**（sysObjectID）から導出される情報もあります。

sysObjectID から NNMi の属性へのマッピングは，**デバイスプロファイル**を使って行われます。デバイスプロファイルは，モニタリング用にノードをグループにまとめたり，表示用にノードをフィルタしたり，検出のメンテナンス用にノードをカテゴリにまとめたりするときに使用されます。

次の設定エリアは移行できません。

- カスタマイズしたシンボル
- カスタマイズしたデータベースフィールドとデフォルト値

NNMから収集

1. 使用されている NNM のバージョンについて，OID ファイルのカスタマイズを特定する。
   - NNM 07-10 以前はファイルoid_to_sym，oid_to_type，HPoid2type を使って，システムの sysObjectID をデータベース属性と表示するシンボルにマッピングしています。
   - NNM 08-00 以降は，oid_to_sym ファイルがoid_to_sym_reg ディレクトリ構造に置き換えられています。

   NNMiで再現

   NNMi は，既知のシステムオブジェクト ID について，事前に設定した多数のデバイスプロファイルを提供しているので，必要なデバイスプロファイルをすぐに利用できます。最もシンプルな方法では，検出プロセスを開始し，結果を確認し，必要な場合だけ変更を行います。
2. NNMi コンソールでは，［設定］ワークスペースから［デバイスのプロファイル］を選択する。
   カスタマイズした値ごとにsysObjectID でエントリを見つけます。
3. 必要に応じてデバイスプロファイル設定を更新する。
   - NNMi が提供しているエントリについては，設定されている値が NNM での属性と一致することを確認します。
   - NNMi が提供していないエントリについては，sysObjectID 用に新しいデバイスプロファイルを作成します。
4. 最初の検出のあと，ノードインベントリで，［デバイスのプロファイル］列をソートして，［<No Device Profile>］であるノードを見つける。
   ［<No Device Profile>］というプロファイルタイプは，sysObjectID が NNMi でまだ設定されていないことを示しています。NNMi は，［<No Device Profile>］のノードにデフォルトのモニタリング設定を適用します。また，これらのノードはフィルタが困難です。
   NNMi データベース内のすべてのsysObjectID に対してデバイスプロファイルが定義されるように，新しいデバイスプロファイルを構築できます。

24.3　フェーズ 2：検出を移行する

検出のスケジュールと設定を移行します。NNMi スパイラル検出は，1 つまたは複数の検出シードを保存すると直ちに開始します。

注意事項

ネットワーク環境向けの適切なコミュニティ文字列を使用するよう NNMi を設定してから検出を開始します。

NNMi で最初の検出が終了したあとに，NNM で手動で設定したデバイス間の接続を移行します。

24.3.1　検出のスケジュールを設定する

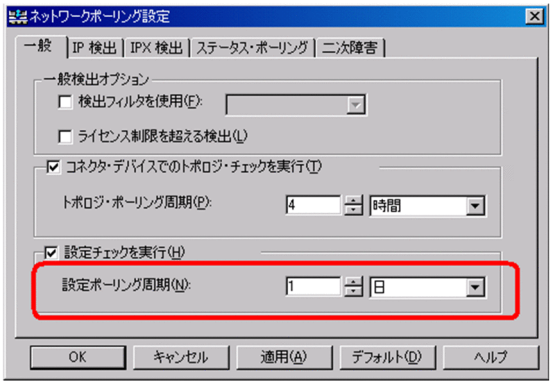
NNM 検出プロセスは独立して実行できます。検出を NNMi に移行するには，NNM がノードを検出する**間隔**を転送するだけで十分です。

次のスケジュール設定エリアは NNMi では使用されなくなっており，移行できません。

- コネクタデバイスのトポロジのチェック。現在は，NNMi が変更の可能性を示すトリガーを見つけるたびに，トポロジチェックが自動的に行われるようになりました。
- 設定チェック。NNMi では，設定チェックはスケジュールされた検出の時点，またはさまざまなトリガーによって行われるようになりました。
- レイヤー 2（拡張トポロジ）検出動作。NNMi は，各デバイスを見つけたときにレイヤー 2 検出を実行するので，この動作を別にスケジュールする必要はありません。
- 検出ポーリング間隔の自動調整。

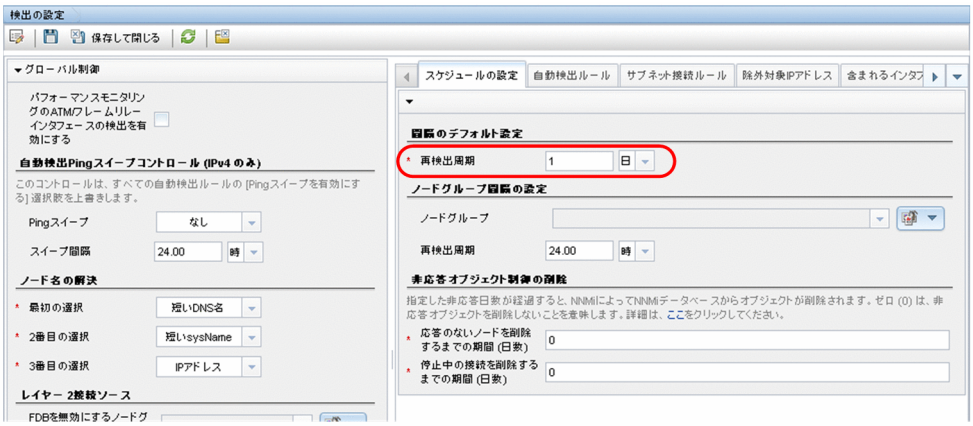
NNMから収集

1. NNM がいつ再検出を実施しているか特定する。
   a　ユーザーインタフェースで，**［オプション］＞［ネットワークポーリング設定］**を選択します。
   b　**［IP 検出］**ページで，**［検出ポーリング周期］**ボックスを確認します。
   －固定間隔を使っている場合は，NNMi で設定するために，その値を控えてください。
   －NNM で自動調整間隔を使っている場合，NNM は最大 24 時間待機します。NNMi では，デフォルト値である 24 時間のままにしておくこともできますし，新しい値を選択することもできます。
   －自動検出が有効になっていない場合は，**［一般］**ページの **［設定チェックを実行］**の周期を調べ，NNMi で設定するために，その値を書き留めてください。



NNMiで再現

2. NNMi コンソールで，［設定］ワークスペースから［検出］＞［検出の設定］を選択し，［再検出周期］を手順 1.で決定した値に設定する。



NNMiでの強化

ほかの設定更新はすべて自動的に追加されていくので，NNM よりも設定が簡単で，検出が効率的です。

24.3.2　検出方法を選択する

NNMi 検出に，次のどのモデルを使うか決定します。

- 自動検出ルールなしのシード検出。この種類の検出では，管理者が必要なノードだけをシードに追加するので，検出されるノードを制限できます。次の操作だけを実施してください。
  －「24.3.4　シード検出を追加する」
- シードと自動検出ルールに基づいた自動検出。次の両方の操作を実施してください。
  －「24.3.3　自動検出ルールを設定する」
  －「24.3.4　シード検出を追加する」

NNMi 検出方法の間の違いについては，NNMi ヘルプの「*検出のアプローチを決定する*」を参照してください。

参考

NNM のライセンスは，管理下にあるノード数に基づいて判断されます（ステータスをモニタリングされるノード）。NNMi のライセンスは，検出されたトポロジに配置されたノード数に基づいて判断されます（モニタリングされるノードとモニタリングされないノード）。

この違いがあるので，検出ノード数を少なくしようとする人もいるでしょうが，モニタリングされないノードをデータベースに入れると利点もあります。

例

- デバイスの管理を担当しない場合でも，サービスプロバイダのアクセスルーター，およびそれへの接続を表示できます。
- ステータスモニタリングアルゴリズムはデータベースに表示される接続に基づいています。リンクの他端のデバイスがデータベースにないインタフェースは，デフォルトでモニタリングされません。ステータスモニタリング設定でデフォルトを書き換えることもできますし，そのデバイスを検出することもできます。どちらを選択するかは，ご使用の環境についてどこに関心を置くかによって決まります。詳細については，「5.2.2(1)　監視されないノードへのインタフェース」を参照してください。

24.3.3　自動検出ルールを設定する

NNMi 検出設定は，NNMi の管理対象について考える良い機会です。NNM の検出設定とフィルタの変換を行う前に，現在のネットワーク環境を考察し，NNMi トポロジに組み込むものについて考えてください。

直接変換を行いたい場合，NNMi 検出ルールには NNM の次の 2 つのタスクセットが含まれています。検出のスコープの拡大，およびスコープ内で検出されるオブジェクトの制限です。

参考

NNMi 設定の場合，検出を拡大または制限する全ルールを定義してから，検出プロセスを開始するシードを入力することが重要です。

次のスケジュール設定エリアは NNMi では使用されなくなっており，移行できません。

- Windows からの IPX 検出
- ライセンスの制限を超える検出
- レイヤー 2 オブジェクトの検出の無効化（NNMi については常に有効）
- IP アドレスと`sysObjectID`（およびその派生物）以外の属性のフィルタによる検出の除外
- CDP プロトコルエリア（統合ポート，vlan など）に基づいたレイヤー 2 検出の制限

- 拡張トポロジゾーンの設定。NNMi のスパイラル検出には該当しなくなっています。

(1) スパイラル検出の設定

NNMi には，NNMi でスパイラル検出を設定する次の 2 つの方法があります。ノードの手動でのロード（例えば，ホストファイルから），および自動検出ルールの使用です。

(a) ノードの手動でのロード

NNMから収集

1. NNM で，loadhosts コマンドに入力した内容を含むファイルを見つける。

   このファイルには，各ノードの IP アドレスとホスト名，さらに指定されている場合はサブネットマスクがリストされています。

   NNM loadhosts の例

   loadhosts コマンドのファイルの例は次のとおりです。

   ```
   10.2.32.201 lnt04.mycorp.net # comment
   10.2.32.202 lnt07.mycorp.net # comment
   10.2.32.203 lnt03.mycorp.net # comment
   10.2.32.204 lnt02.mycorp.net
   10.2.32.205 lnt05.mycorp.net
   ```

NNMiで再現

2. NNMi では，NNM loadhosts コマンドと同じ方法で検出シードを使用できる。

   これを行うには，-f オプションとシードファイルを指定して，nnmloadseeds.ovpl コマンドを使用します。

> **参考**
>
> シードを NNMi に設定する前に，すべてのコミュニティ文字列の設定を完了してください。

> **参考**
>
> 検出の結果を NNM loadhosts と同じにするには，NNMi で設定されている自動検出ルールを無効にします。自動検出ルールを無効にするには，次の 1 つを実行します。
>
> - ［検出の設定］フォームからルールを削除します。
> - ［自動検出ルール］フォームで，［マッチングノードの検出］チェックボックスをオフにします。

NNMi のシードファイルのフォーマットでは，行ごとに IP アドレスまたはノード名（任意でコメント付き）があります。詳細は，*nnmloadseeds.ovpl* リファレンスページを参照してください。

NNMi シードファイルの例

次の例に，NNM loadhosts コマンドおよびホストファイルと同じ機能の NNMi シードファイルを示します。

```
10.2.32.201 # comment
10.2.32.202 # comment
lnt03.mycorp.net # comment
lnt02.mycorp.net
10.2.32.205
```

ポイント

NNMi では，管理アドレスとしてループバックアドレスが必ず優先されます。ループバックアドレスを使わない場合，NNMi では，管理アドレスとしてシードアドレスがおそらく使われます（必ずではありません）。したがって，優先される IP アドレスの書かれた hosts ファイルをコピーするのが良いやり方です。ホスト名を使う場合は，DNS が優先管理アドレスとして解決することを確認します。しかし，NNMi が管理アドレスとしてこのアドレスを使うことが保証されるわけではありません。管理アドレス選択の詳細は，NNMi ヘルプの「*検出ノード名の選択*」を参照してください。

(b) 自動検出ルールの使用

NNMから収集

1. NNM に検出フィルタが使われたかどうかを調べる。

NNM では，1 つの検出フィルタが検出のスコープ全体に適用されます。

a NNM ユーザーインタフェースを開きます。

b ［オプション］＞［ネットワークポーリング設定］を選択します。

c ［全般］ページで［検出フィルタを使用］チェックボックスを確認し，オンの場合は使用中の検出ファイルを書き留めてください。フィルタが使用されていない場合は「24.3.4 シード検出を追加する」を続けます。

d 次のファイル内で検出フィルタを見つけます。

－Windows：%OV_CONF%¥C¥filters

－UNIX：$OV_CONF/C/filters

ロジックを注意深く確認します。NNMi では，IP アドレスの範囲とシステムオブジェクト ID の範囲をフィルタできます。ホスト名のワイルドカードから IP 範囲への変換や，ベンダー名からシステムオブジェクト ID 範囲への変換のように，移行できるオブジェクトもあります。

NNM 検出フィルタの例

次の例に，NNM フィルタを示します。例えば，ルーター，ブリッジ，Nokia_Firewalls，NetBotz，NetsNSegs です。NetBotz ファイアウォールと Nokia ファイアウォールはsysObjectIDで定義されます。

```
Nokia_Firewalls "Nokia Firewalls"
{ ( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.1 ) ) ||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.9 ) ) ||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.10 ) ) ||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.10.11 ) ) ||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.10.12 ) ) ||
( isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.94.1.21.2.1.138 ) ) }
NetBotz "NetBotz"
{ isNode && ( "SNMP sysObjectID" ~ .1.3.6.1.4.1.5528.* ) }
My_NetInfrastructure "My Network Infrastructure"
{ Routers || Bridges || Nokia_Firewalls || NetBotz || NetsNSegs }
```

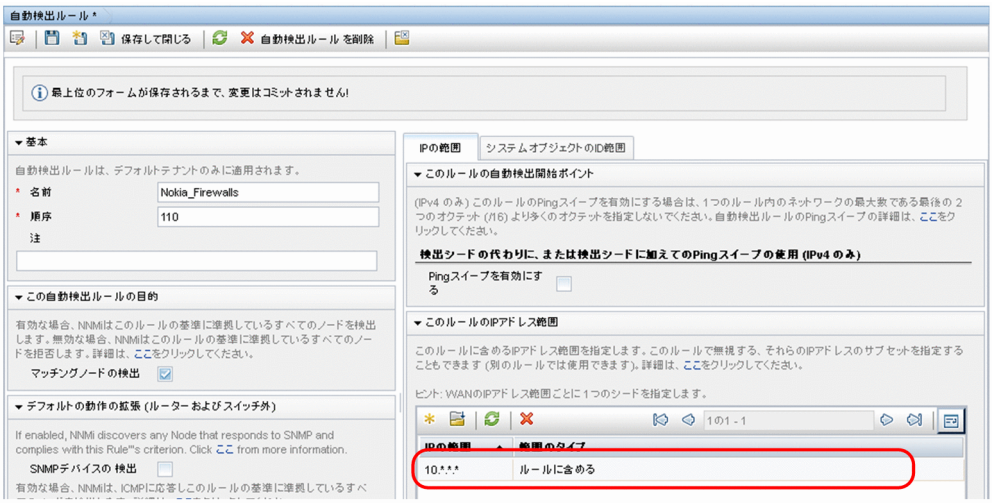
NNMiで再現

2. NNMi コンソールから，検出フィルタを入力する。

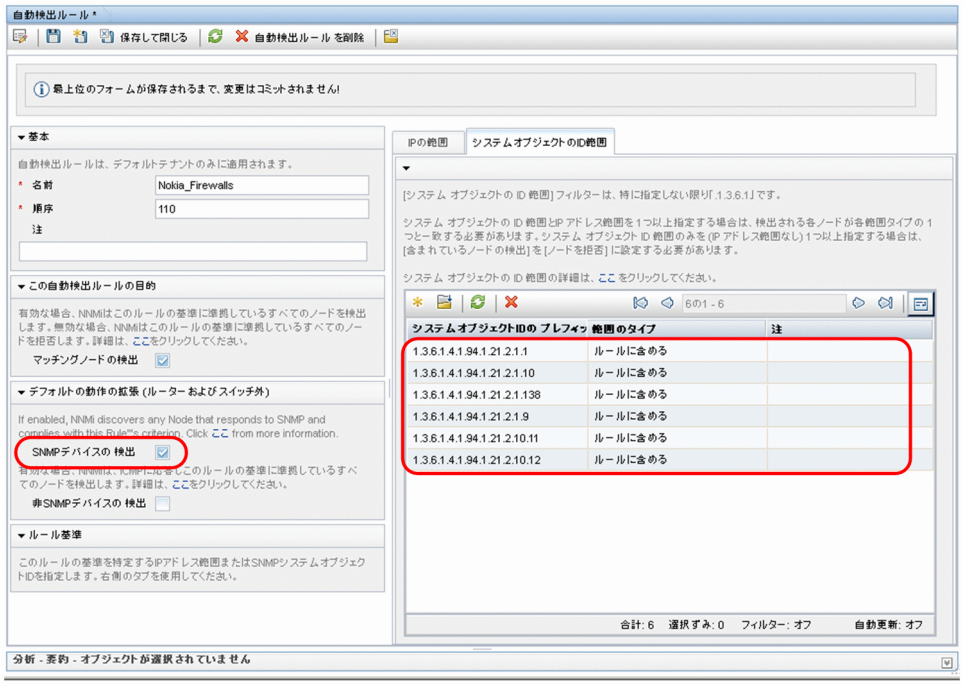
NNMi 検出フィルタエントリの例

例えば，「自動検出ルールの使用」の手順 1 .の「NNM 検出フィルタの例」に示す NNM フィルタを NNMi に移行するには，次の 3 つの自動検出ルールを定義します。1 つのルールは Nokia ファイアウォール用，1 つのルールは NetBotz デバイス用，最後の 1 つのルールはルーターとスイッチ用です（NNM 08-00 以降の Bridge と同じ）。NNMi では，NetsNSegs は不要です。この例の場合，検出されるネットワークの範囲は 10.*.*.* と仮定します。

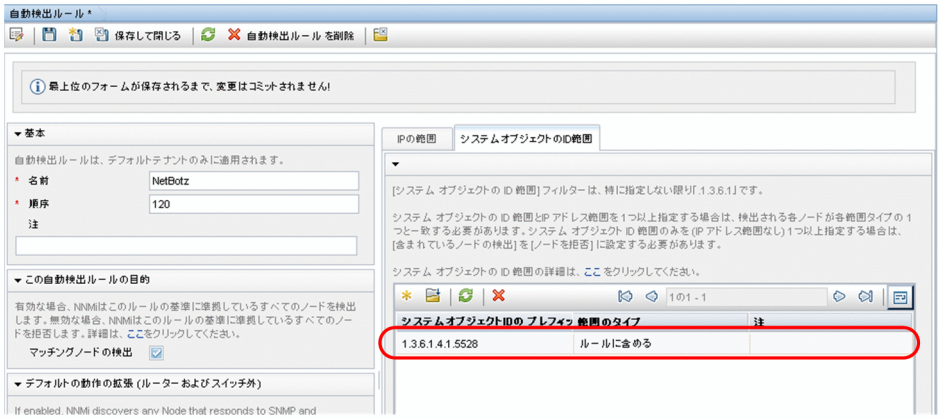
a　Nokia ファイアウォールの場合，ルール名（Nokia_Firewalls）を入力してから，ネットワーク IP 範囲 10.*.*.* を入力します。



b　各sysObjectIDを入力し（先頭のピリオドは入力しません），次に［SNMPデバイスの検出］チェックボックスをオンにします（デフォルトでは，NNMiはスイッチとルーターだけを検出します。これらのデバイスはスイッチまたはルーターとマークされていないこともあるので，sysObjectIDsを指定するときに［SNMPデバイスの検出］チェックボックスをオンにします）。



c　NetBotzルールを入力します。ここでもIP範囲の設定が必要です。このルールではNNM . 1.3.6.1.4.1.5528.*.にワイルドカードを使います。NNMiでは，アスタリスク（*）は黙示的なので，不要です。



d　最後のルールはスイッチとルーター用です。NNMiはデフォルトでこれらのデバイスを検出するので，オブジェクトID（OID）は指定しないでください。IP範囲だけを指定する必要があります。

24.3.4　シード検出を追加する

NNMから収集

1. 次のコマンドを実行して，NNM データベース内のデバイスの正確なリストを調べる。

```
ovtopodump > topology.out
```

NNMiで再現

2. NNM からtopology.out（エクスポート）ファイルをコピーおよび編集する。または NNMi にインポートするために，ファイルにエントリを再入力する。

   新しいファイルでは，行ごとに IP アドレスまたはホスト名を記載してください。NNMi がサブネットマスクを自動的に決定するので，サブネットマスクを指定する必要はありません。

```
10.2.32.201 # comment
10.2.32.202 # comment
lnt03.mycorp.net  # comment
lnt02.mycorp.net
10.2.32.205
```

> **参考**
>
> この代わりに，NNMi コンソールを使ってノードのリストを追加することもできます。

3. 次のコマンドを実行する。

```
nnmloadseeds.ovpl -f newSeedfile
```

詳細は，nnmloadseeds.ovpl のリファレンスページを参照してください。

NNMi は，これらのシードと関連づけられたデバイスの検出を直ちに開始し，既存のデバイスプロファイル（およびステータスモニタリング用のノードグループなど，ノードグループ）を実装します。NNMi スパイラル検出は継続します。検出シードの結果を知る方法については，マニュアル「*JP1/Cm2/Network Node Manager i インストールガイド*」の「*4.3.3　検出の進行状況を確認する*」を参照してください。

24.4　フェーズ 3：ステータスモニタリングを移行する

NNM では，netmon プロセスがステータスモニタリングを実行します。

- netmon プロセスは，デバイス（インタフェースを含むノードなど）をモデル化し，おもにノードレベルでポーリングパラメータを適用します。

NNMi では，ノード，インタフェース，またはアドレスのレベルでポーリングパラメータを適用できます。

24.4.1　ポーリング間隔を設定する

NNM netmon ポーリングプロセス

NNMから収集

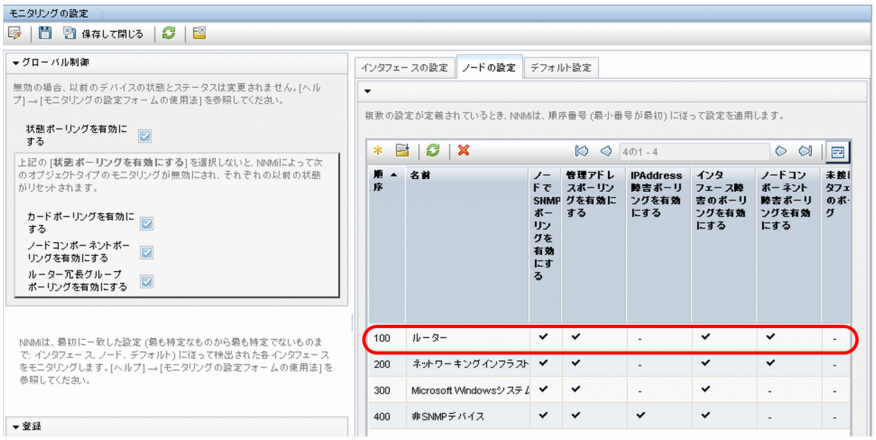
NNM ユーザーインタフェースからポーリング間隔を取得します。

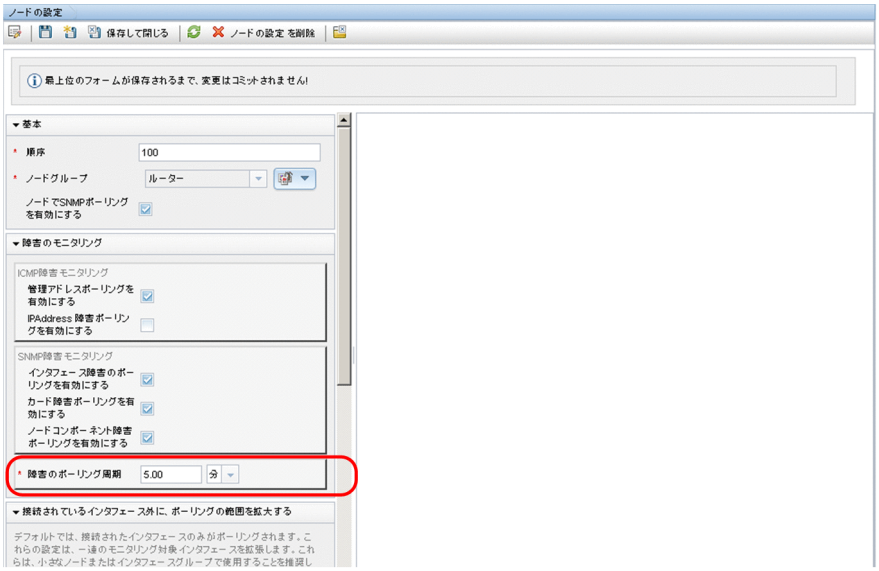
NNMi ポーリングプロセス

NNMiで再現

NNMi ステータスモニタリング設定はノードのグループまたはインタフェースのグループ（またはその両方）に基づいています。

NNMi コンソールで，**[設定]** ワークスペースから **[モニタリング]** ＞ **[モニタリングの設定]** を選択します。**[ノードグループの設定]** タブを選択してから，グループについて **[障害のポーリング周期]** を設定します。





24.4.2　ポーリングプロトコルを選択する

NNM netmon ポーリングプロセス

NNMから収集

デフォルトで，netmon プロセスは ICMP を使用して各アドレスをポーリングします（各アドレスはインタフェースと同一視されます）。netmon プロセスがデバイスによっては，ICMP でなく SNMP を使うように NNM を設定することもできます（両方を使うことはありません）。SNMP を使っているエリアがあるかどうか調べるには，次のファイルを確認します。

- Windows：%OV_CONF%¥netmon.snmpStatus
- UNIX：$OV_CONF/netmon.snmpStatus

NNMi ポーリングプロセス

NNMiで再現

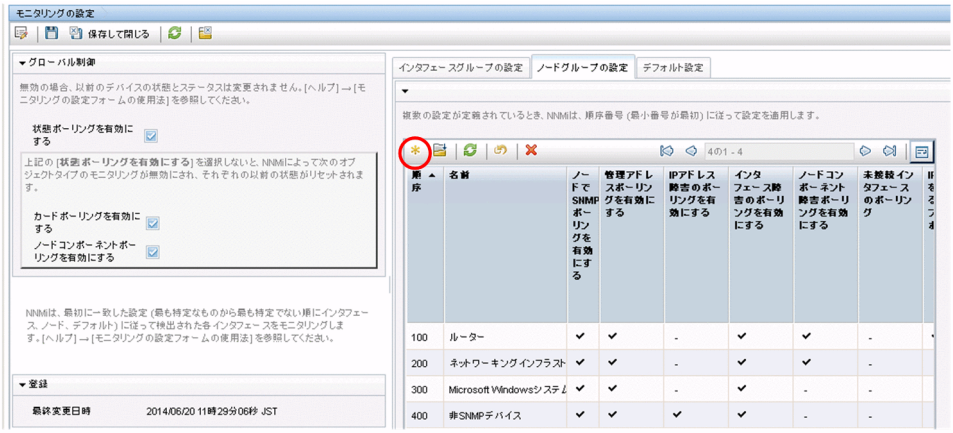
NNMi では，ノードとインタフェースの集合はノードグループとインタフェースグループとして定義します。ポーリング方針は［**モニタリングの設定**］フォームでノードグループとインタフェースグループに適用されます。

NNMi ポーリング設定の例

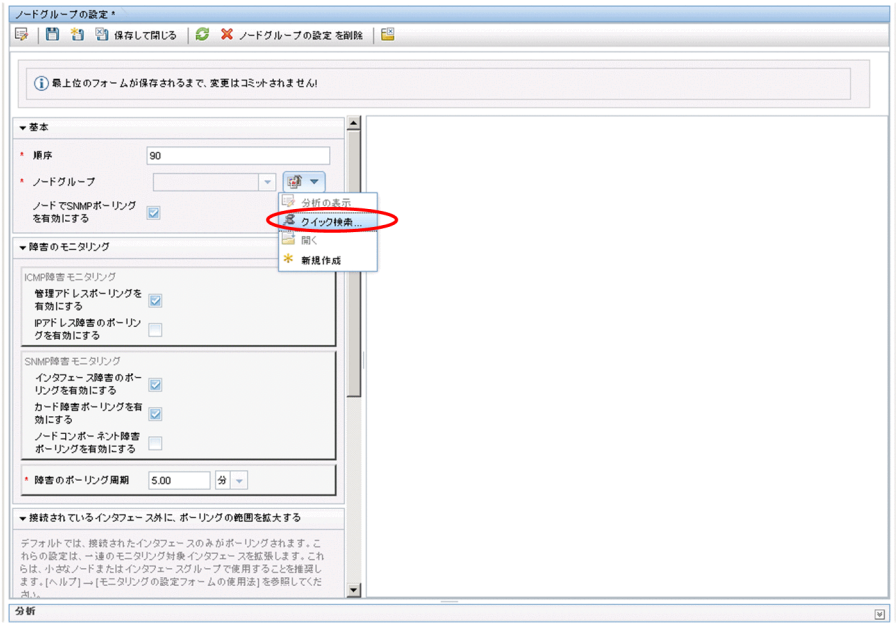
例えば，（SNMP と ping を使って）VOIP ルーターの集合にポーリングを設定するには，次の手順に従います。

1. ［**ノードグループ**］フォームを使って，VOIP ルーターを識別するノードグループを作成する。このフォームを保存し，閉じる。

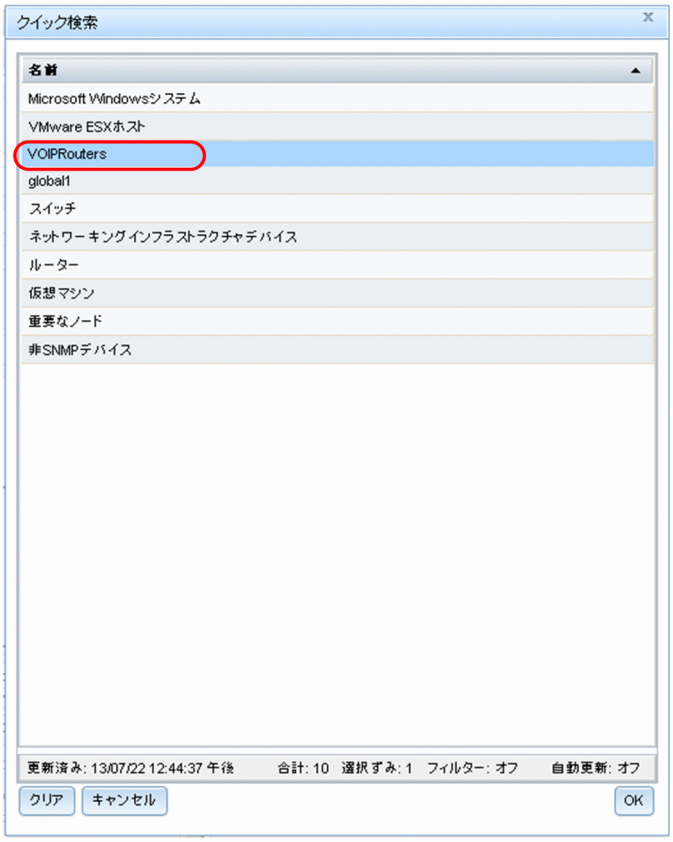
2. **[モニタリングの設定]** フォームで，次のように，新しいノードの設定を追加する。



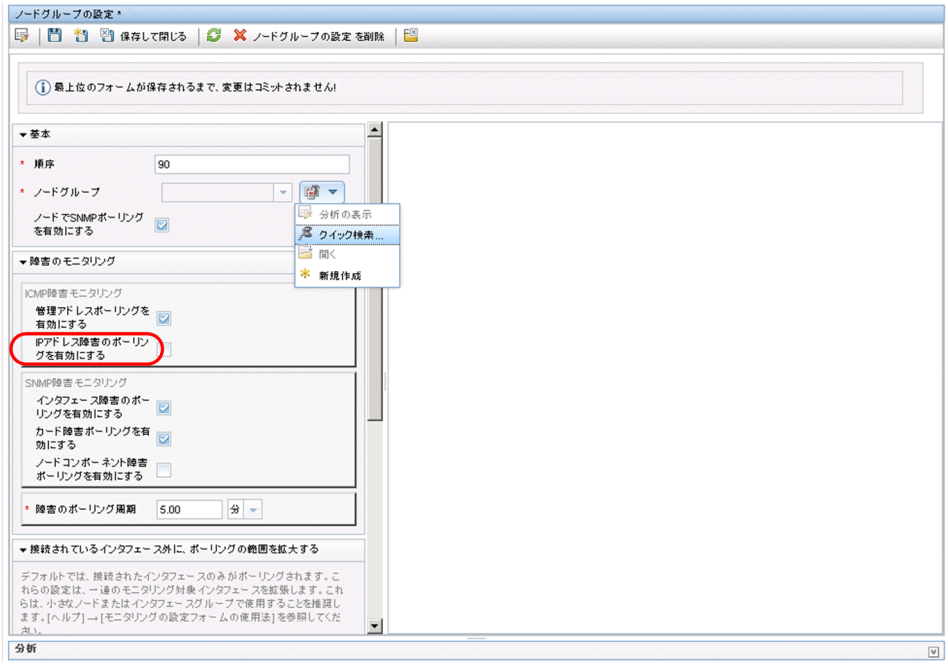
3. 順序づけの値を指定してから，次のように，**[ノードグループ]** フィールドの **[クイック検索]** を選択する。



4. 次のように，モニタリング設定用のノードグループを選択する。



5. 次のように，[IP アドレス障害のポーリングを有効にする] チェックボックスをオンにする。フォームを保存し，閉じる。



24.4.3　重要なノードを設定する

デフォルトで，NNMi には重要なノード用のノードグループがあります。

重要ノードが故障または到達できない場合，NNMi は，ノードステータスが危険域であると表示し，NodeDown インシデントを生成します。

NNM netmon ポーリングプロセス

NNMから収集

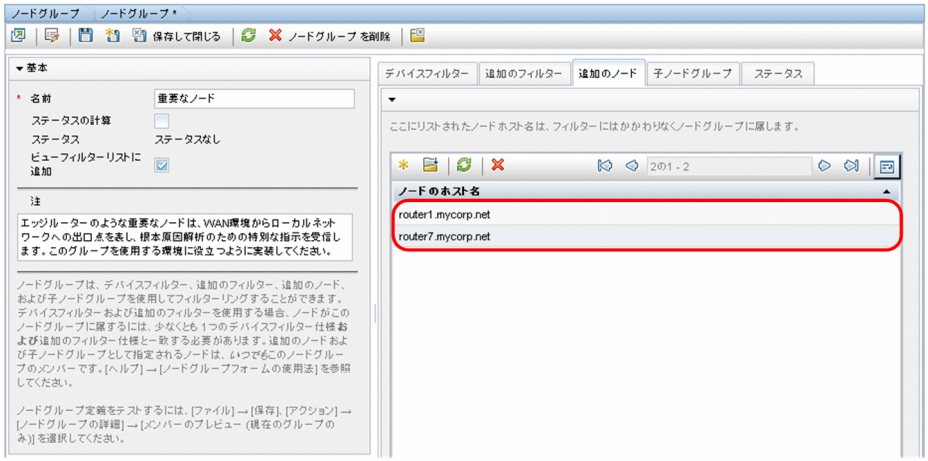
NNM は重要なノード用の設定はありません。NNMi に新しい重要なノードの設定を作成できます。

NNMi ポーリングプロセス

NNMiで再現

1. NNMi コンソールで，[設定] ワークスペースから [オブジェクトグループ] ＞ [ノードグループ] を選択する。
2. [重要なノード] ノードグループを開く。
3. 次のように，ホスト名ワイルドカード，デバイスフィルター，または特定のノードごとに，重要ノードをグループに追加する。
   a　デバイスフィルターを追加します。

   b　特定のノードを追加します。フォームを保存し，閉じます。

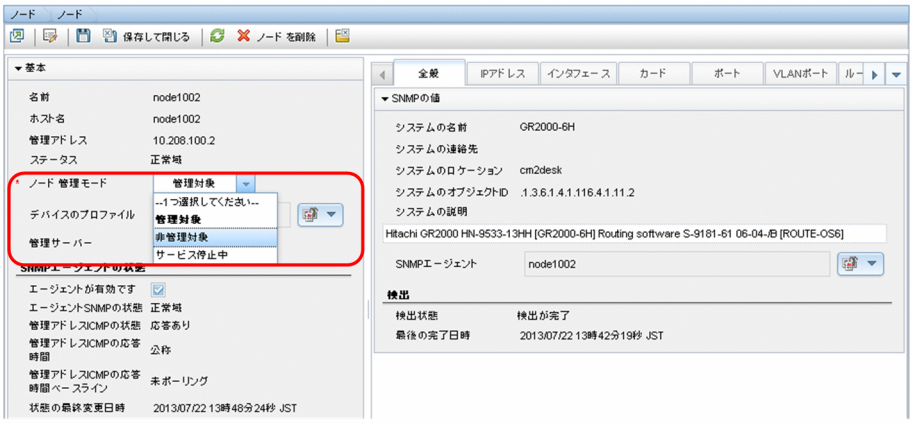


24.4.4　ステータスポーリングからオブジェクトを除外する

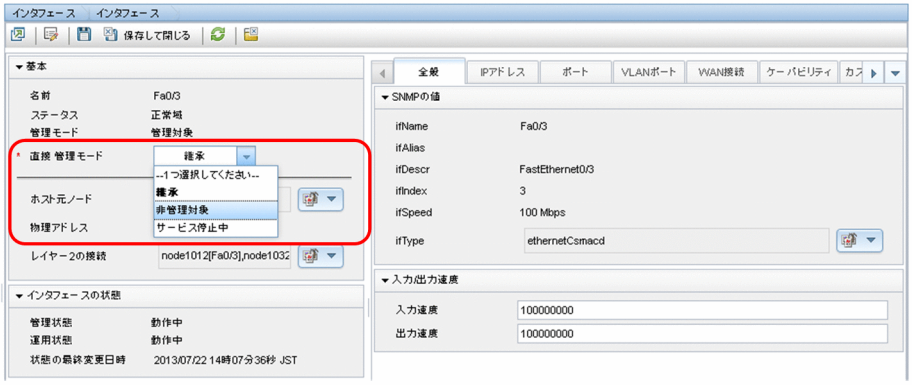
NNM では，ノードまたはインタフェースがモニタリングされるのを停止する（UNMANAGED（管理対象外）状態に設定する）ほとんどのアクティビティは，NNM ユーザーインタフェースによって手動で設定します。

NNMi はオブジェクトを管理対象外にするプロセスを簡単にします。新しい運用のデフォルトを，手動で実行していたものと一致させることはできます（例えば，アップリンクだけポーリングするなど）。しかし，ノードグループとインタフェースグループを使って設定を管理すれば，設定の自動更新が簡単になります。

ノードまたはインタフェースを Not Managed（管理対象外）とマークする必要がある場合もあります。次のように，個別のノードの管理モードを［ノード］フォームで設定できます。



次のように，個別のインタフェースの管理モードを［インタフェース］フォームで設定できます。


インタフェース
インタフェース
保存して閉じる
基本
名前
Fa0/3
ステータス
正常域
管理モード
管理対象
直接 管理モード
継承
--1つ選択してください--
継承
非管理対象
サービス停止中
ホスト元ノード
物理アドレス
レイヤー2の接続
node1012[Fa0/3],node1032
インタフェースの状態
管理状態
動作中
運用状態
動作中
状態の最終変更日時
2013/07/22 14時07分36秒 JST
全般
IPアドレス
ポート
VLANポート
WAN接続
ケーパビリティ
SNMPの値
ifName
Fa0/3
ifAlias
ifDescr
FastEthernet0/3
ifIndex
3
ifSpeed
100 Mbps
ifType
ethernetCsmacd
入力/出力速度
入力速度
100000000
出力速度
100000000

24.5　フェーズ 4：イベント設定とイベント削減を移行する

NNM は，拡張 SNMPv2 フォーマットを使って，受信イベント（管理対象デバイスからのトラップ，内部プロセス通信，転送されたイベント）の全ソースを分析します。イベントごとに，イベントオブジェクト識別子，名前，および設定パラメータがあります。

NNMi はイベントのさまざまなソースをそれぞれ異なるように処理します。デバイスからのトラップは SNMPv2 フォーマットです。さらに，NNMi 内部プロセス通信は新しい（トラップでない）メカニズムを使って，全般的なパフォーマンスを大幅に向上させています。NNMi では，認識されないイベントに関する「no format in trapd.conf」メッセージがありません。認識されないメッセージはデフォルトでは破棄されるようになりました。

次のイベント設定エリアは NNMi では使われなくなっており，移行もできません。

- 構成要素相関処理の種類の幾つか：suppress（抑制），enhance（強化），transient（過渡的），multisource（複数ソース）

24.5.1　デバイスからのトラップを表示する

NNM 環境に類似した方法で，デバイスからのトラップを表示するよう NNMi を設定できます。

NNMi には，NNM に同梱されている一般的な SNMP トラップおよびベンダートラップの多くのデフォルト設定があります。これらトラップのカスタマイズによって，NNMi を更新できます。

メッセージと自動アクションに利用できる変数のリストについては，NNMi ヘルプの「*インシデントメッセージを設定するための有効なパラメーター（SNMP トラップインシデント）*」と「*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（SNMP トラップインシデント）*」を参照してください。

NNMから収集

1. NNM 設定にカスタマイズされたトラップがあるかどうか調べる。

   カテゴリ，重要度，表示メッセージ，または自動アクションについて行われたカスタマイズに注意してください。

   NNMiで再現

2. ベンダー MIB ファイルを NNMi 管理サーバーにダウンロードする。

3. MIB ごとに次のコマンドを実行する。

```
nnmloadmib.ovpl -load mibFile
nnmincidentcfg.ovpl -loadTraps mibModule
```

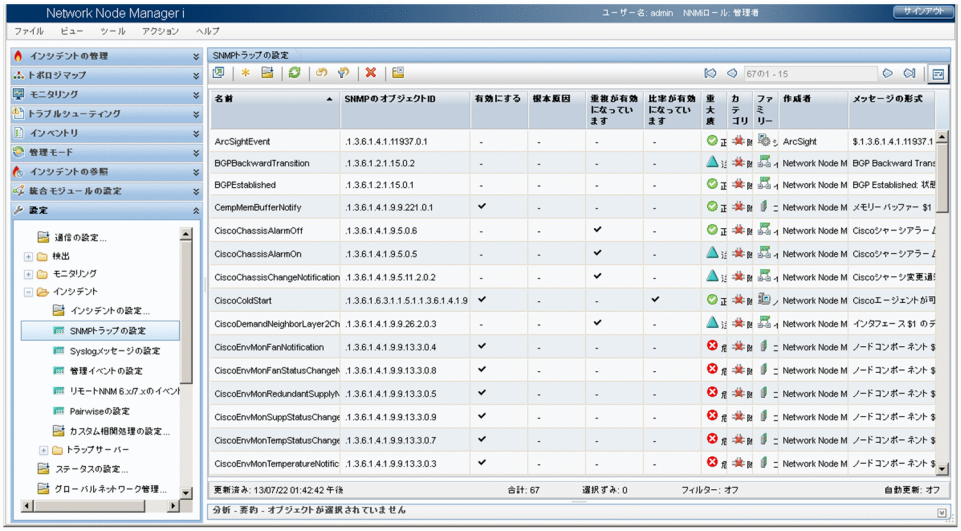
   - どの MIB がすでにロードされているか知るには，次のコマンドを使用します。

```
nnmloadmib.ovpl -list
```

詳細は，nnmincidentcfg.ovpl と nnmloadmib.ovpl のリファレンスページを参照してください。

> **参考**
>
> これらの手順では，TRAP-TYPE と NOTIFICATION-TYPE の MIB エントリをロードするだけです。NNMi はほかの MIB 変数を使いません。

4. NNMi コンソールで，［設定］ワークスペースから［インシデント］＞［SNMP トラップの設定］を選択する。



5. トラップ表示が NNM での表示と一致するようにカスタマイズする。

［SNMP トラップの設定］フォームで，必要に応じてカテゴリを作成できます。

NNMiでの強化

6. (任意) デフォルトの Severity (重大度), Category (カテゴリ), および Message (メッセージの形式) の設定に加えて, デフォルトの Family (ファミリ) を設定する。

7. (任意) トラップが [根本原因インシデント] ビューに表示されるように, トラップを根本原因として分類する。

24.5.2 NNMi で生成された管理イベント表示をカスタマイズする

NNMi では, イベント設定は簡単になっています。NNMi Causal Engine は NNM よりも簡潔な根本原因を生成します。

NNMi で生成されたインシデントを変更し, NNM アラームと類似した外見にします。例えば, NNMi NodeDown インシデントメッセージを NNM NodeDown アラームメッセージに類似するようカスタマイズできます。

NNMから収集

1. NNM で, イベント設定のカスタマイズを特定する。

   NNMiで再現

2. NNMi コンソールで, [設定] ワークスペースから [インシデント] > [管理イベントの設定] を選択する。

3. イベント番号ではなく名前で, 新しいインシデント設定を見つける。

4. (任意) イベント表示を NNM のイベント表示と一致するようカスタマイズするには, 管理イベントの設定フォームでカテゴリを作成する。

5. デフォルトの Severity (重大度), Category (カテゴリ), および Message (メッセージの形式) 設定に加えて, デフォルトの Family (ファミリ) を設定できる。

24.5.3 トラップのブロック／無視／無効化を設定する

NNM にはさまざまなレベルのイベント処理が備わっています。

- トラップがovtrapd に入ってくるときにトラップをブロックする。
- IGNORE というラベルのトラップまたはイベントの処理はするが, 保存または表示はしない。
- LOGONLY というラベルのイベントの保存および処理 (相関) をするが, 表示はしない。
- イベントをカテゴリに保存, 処理, 表示する。
- 設定なしに到着するトラップは,「No format in trapd.conf for…」としてアラームブラウザに表示され, データベースに保存される。

NNMi にはもっとシンプルな方法があります。*disabled（無効）*イベントまたはトラップは保存，処理，または表示されません。*enabled（有効）*イベントまたはトラップは完全に保存，処理，表示されます。NNMi に設定がないイベントはブロックされます。

NNMから収集

1. トラップを無視するカスタマイズまたはトラップをLOGONLY に設定するカスタマイズを特定する。
2. NNM がトラップフィルタメカニズム（ovtrapd.conf，NNM 08-00 で新規）を使用するかどうか調べる。

   NNMiで再現

3. NNMi コンソールで，［設定］ワークスペースから［インシデントの設定］を選択する。

   受信または表示したくないイベントを見つけ，これらイベントの［有効にする］チェックボックスをオフにします。
4. 特定の IP アドレスからトラップをブロックするには，次のファイルを編集し，NNM からのトラップフィルタリング情報を使用して NNMi をアップデートする。
   - Windows：%NnmDataDir%shared¥nnm¥conf¥nnmtrapd.conf
   - UNIX：$NnmDataDir/shared/nnm/conf/nnmtrapd.conf
5. nnmtrapconfig.ovpl コマンドを使用してトラップブロッキングを有効にし，トラップブロッキングのレートとしきい値を設定する。

   このコマンドの使用法の詳細は，nnmtrapconfig.ovpl のリファレンスページを参照してください。

24.5.4 自動アクションを設定する

NNMから収集

1. NNM 用に設定された自動アクションを決定する。

   NNMiで再現

2. NNM 管理ステーションのアクションスクリプトを NNMi 管理サーバーにコピーする。

   この場合，ファイルの位置は重要ではありません。
3. NNMi コンソールで，［設定］ワークスペースから［インシデントの設定］を選択する。
4. 自動アクションのある NNM イベントごとに，対応する NNMi インシデントをそのアクションで設定する（[アクション］タブ）。

   アクションを有効にするためには，［有効にする］チェックボックスをオンにする必要があります。
5. NNM の動作と一致させるために，［ライフサイクル状態］を［登録済み］に設定する。

NNMiでの強化

6. 次の NNMi 設定に注意する。
   - イベント到着時に発生する複数の自動処理を設定できます。
   - ほかのライフサイクル状態ごとに，1 つまたは複数の追加処理を設定できます（ライフサイクル状態は，In Progress（進行中），Completed（完了），Closed（解決済み））。
   - NNM より多くのインシデント属性をコマンドに渡せます。
   - NNMi がコマンドを実行する前に，別の設定ファイルにコマンドを登録する必要はないので，手順は簡単になっています。

24.5.5 追加（手動）アクションを設定する

NNM には，アラームブラウザのメニューから利用できるオペレータのアクションまたは追加のアクションが用意されています。NNMi コンソールメニューから利用できる URL アクションで NNM のアクションをシミュレートすることもできます。

NNMから収集

1. NNM にあるカスタムオペレータアクションを決定する。

   NNMiで再現

2. これらのカスタムアクションについて，URL として利用できるように移行する方法を特定する。
3. NNMi コンソールで，［設定］ワークスペースから［ユーザーインタフェース］＞［メニュー項目］を選択する。
4. ［新規作成］をクリックする。
5. アクションについて［メニュー項目ラベル］，［一意のキー］，［順序］，［選択タイプ］，［メニュー項目コンテキスト］をすべて用意する。

24.5.6 イベント相関処理：イベントの繰り返し

NNM では，イベントを複製するときに，最初のイベントまたは最後のイベントのどちらかを親として使用します。

NNMi では新しい親が作成され，［インシデントの参照］ワークスペースの［すべてのインシデント］を選択すると表示されます。またオリジナルのイベントが，設定されたビューに表示されます。

NNMから収集

1. RepeatedEvents 相関処理が NNM に使われるかどうか調べる。

2. Repeated 相互関係が NNM に使われるかどうか調べる。

3. 複製が使われているかどうか調べる（dedup.conf ファイル）。

NNMiで再現

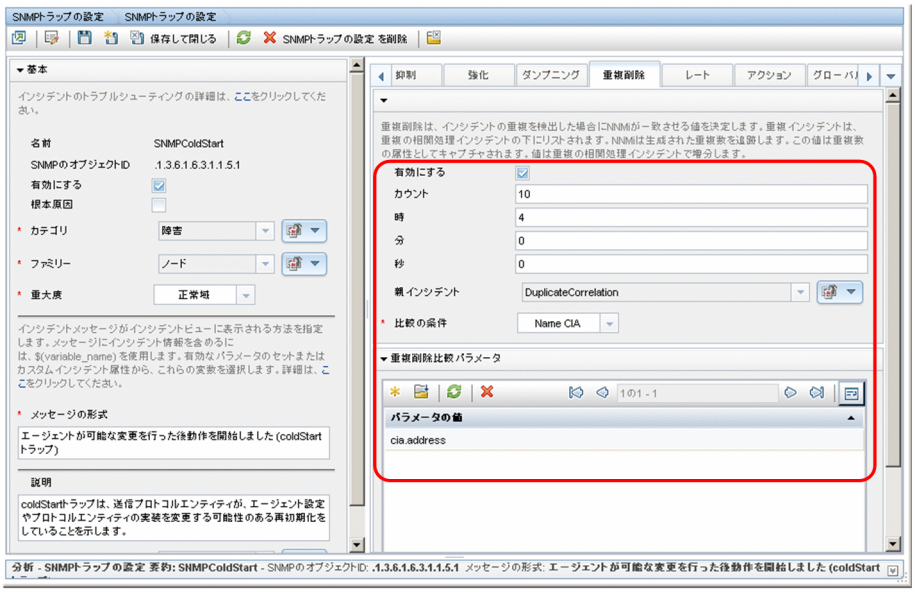
4. NNMi コンソールで，［設定］ワークスペースから［インシデントの設定］を選択する。

5. 複製するイベントを開く。

6. ［重複削除］タブを選択し，重複削除を有効にし，新しい親イベントを選択し，一致基準を定義する。

参考

NNMi での複製には時間の制限がありません。



24.5.7　イベント相関処理：レート計算

NNM では，イベントを複製するときに，最初のイベントまたは最後のイベントのどちらかを親として使用します。

NNMi では新しい親が作成され，［インシデントの参照］ワークスペースの［すべてのインシデント］を選択すると表示されます。またオリジナルのイベントが，設定されたビューに表示されます。NNMi は，レートの動作を NNM の定期的時間ウィンドウと同じになるようにしました。

NNMから収集

1. レート相関処理が NNM に使われるかどうか調べる。

NNMiで再現

2. NNMi コンソールで，［設定］ワークスペースから［インシデント］＞［管理イベントの設定］を選択する。

3. カウントされるイベント識別子を開く。

4. ［レート］タブを選択し，次を実行する。
   a ［有効にする］を選択してモニタリングを有効にします。
   b カウントの範囲を設定します。
   c 時間の範囲を設定します（［時］，［分］，および［秒］の各フィールド）。
   d 新しい親イベントを選択します（［相関処理インシデントの設定］）。
   e ［比較の条件］を定義します。
   詳細は，NNMi ヘルプの「*［管理イベント］フォーム*」を参照してください。

24.5.8 イベント相関処理：Pairwise のキャンセル

NNMi では，キャンセルは特定の期間に制限されません。

NNMから収集

1. NNM で，PairWise（ペアイベント）相関処理が使われるかどうか調べる。

2. NNM で，過渡状態コリレータが使われるかどうか調べる。

NNMiで再現

3. NNMi コンソールで，［設定］ワークスペースから［インシデント］＞［Pairwise の設定］を選択する。

4. 既存のペアを選択するか，または［新規作成］をクリックする。

5. ペアにされるイベント識別子および一致基準を設定する。
   詳細は，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。

24.5.9 イベント相関処理：ScheduledMaintenance（計画保守）

NNMi では，使用不能ノードのモニタリングを抑制できます。これを行うには，「サービス停止中」モードを使います。NNM とは異なり，「サービス停止中」メンテナンスを前もってスケジュールすることはできません。手動でオブジェクトを「管理対象」モードに戻す必要があります。

参考

「サービス停止中」モードのデバイスが送信した SNMP トラップは NNMi で抑制されます。

組織が ScheduledMaintenance（計画保守）相関処理を使っている場合は，一緒にオフラインになるシステムのリストを使用できます。

NNMから収集

1. ScheduledMaintenance 相関処理が NNM に使われるかどうか調べる。

   NNMiで再現

2. NNMi コンソールで，［設定］ワークスペースから［オブジェクトグループ］＞［ノードグループ］を選択する。
3. NNM メンテナンスリスト内のノードのセットごとにノードグループを作成する。ノードグループをビューフィルタとして利用できるように設定する。
4. メンテナンスのときは，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］を選択する。
5. ビューを特定のノードグループにフィルタするには，上端の［ノードグループのフィルタの設定］セレクタを使用する。
6. 全ノードを選択してから，［アクション］＞［管理モード］＞［サービス停止中］を選択する。
7. メンテナンスが完了したあと，ノードを選択してから，［アクション］＞［管理モード］＞［管理］を選択する。

25 NNMi Northbound インタフェース

JP1/Cm2/Network Node Manager i（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. **この操作はオプションです。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します。**

   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 はトラップを正しく再構成して NNMi インシデントデータを追加できません。したがって，NNMi はこのトラップを転送しません。

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，「6.1.2　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，「［HP 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，NNMiNorthbound 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
	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
	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
   - UNIX：/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/Integrated Management - Universal CMDB 10.1 Full

JP1/Integrated Management - Universal CMDB 10.1 Full（UCMDB：ユニバーサル設定管理データベース）は，検出および依存関係マッピングへのネイティブ統合によって，インフラストラクチャとアプリケーションの関係についての最新で正確な情報を自動的に維持します。

この章では，NNMi と UCMDB の統合について説明します。

26.1 NNMi と UCMDB の統合

NNMi と UCMDB との間で NNMi トポロジ情報を共有します。UCMDB は，設定項目（CI）として NNMi トポロジに各デバイスを保存したり，Discovery and Dependency Mapping（DDM：検出と依存関係マッピング）パターンを NNMi トポロジ用の CI に適用し，デバイス障害の影響を予測したりします。デバイス障害の影響分析は，UCMDB ユーザーインタフェース，および NNMi コンソールから入手できます。

連携できる UCMDB のバージョンについては，NNMi のリリースノートを参照してください。

NNMi と UCMDB は同じコンピュータにインストールできません。これらの製品は，次の構成のどちらかで，異なるコンピュータにインストールする必要があります。

- 異なるオペレーティングシステム

  例えば，NNMi 管理サーバーを Linux オペレーティングシステムにし，UCMDB サーバーを Windows オペレーティングシステムにします。
- 同じオペレーティングシステム

  例えば，NNMi 管理サーバー，UCMDB サーバーを共に Windows オペレーティングシステムにします。

注意事項

UCMDB と連携する場合は，次のことに注意してください。

- NNMi の IPv6 管理機能を無効にする必要があります。
- アプリケーションフェイルオーバー機能による HA 構成は利用できません。
- UCMDB から NNMi に SSL で接続しないでください。

NNMi と UCMDB の統合については，UCMDB 提供の取扱説明書を参照してください。

付録

付録 A　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 A.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¥
```

> **参考**
>
> 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。

- UNIX：

```
-$NnmInstallDir: /opt/OV
-$NnmDataDir: /var/opt/OV
```

> **参考**
>
> UNIX 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。

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 A.2　ほかの使用可能な環境変数」を参照してください。

付録 A.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：”C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥bin¥nnm.envvars.bat”
- UNIX：. /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 A-1（Windows）または表 A-2（UNIX）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 A-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¥nnm¥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:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥nmsas¥server¥nms¥log
%NNM_JBOSS_SERVERCONF%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥nmsas¥server¥nms
%NNM_JRE%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥nonOV¥jdk¥nnm
%NNM_LOG%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥log
%NNM_LRF%	C:¥ProgramData¥Hitachi¥Cm2NNMi¥shared¥nnm¥lrf
%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_SUPPORT%	C:¥Program Files (x86)¥Hitachi¥Cm2NNMi¥support
%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 A-2　UNIX OS での環境変数のデフォルトの場所

変数	HP-UX（例）
$NNM_BIN	/opt/OV/bin
$NNM_CONF	/var/opt/OV/conf

変数	HP-UX（例）
$NNM_DATA	/var/opt/OV
$NNM_DB	/var/opt/OV/shared/nnm/databases
$NNM_JAVA	/opt/OV/nonOV/jdk/nnm/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	/opt/OV/nmsas/server/nms/log
$NNM_JBOSS_SERVERCONF	/opt/OV/nmsas/server/nms
$NNM_JRE	/opt/OV/nonOV/jdk/nnm
$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_SUPPORT	/opt/OV/support
$NNM_USER_SNMP_MIBS	/var/opt/OV/shared/nnm/user-snmp-mibs
$NNM_TMP	/var/opt/OV/tmp
$NNM_WWW	/var/opt/OV/shared/nnm/www

付録 B　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 B.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 B.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 B.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 B.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 B.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：UNIX 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 B.6　失敗のシナリオは何ですか？

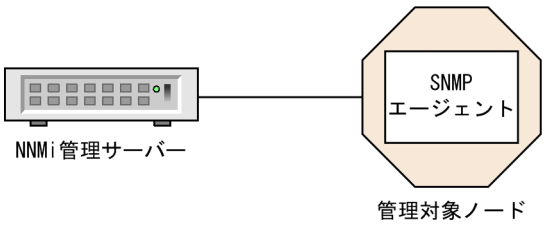
次のシナリオは，ネットワークでの問題の例と，Causal Engine がこれらの問題を診断するために行う作業を示しています。これらのシナリオが示すインシデントの例をほかの例とともに次の表に示します。

表 B-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。

インシデント名	説明
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，隣接ノードが 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。

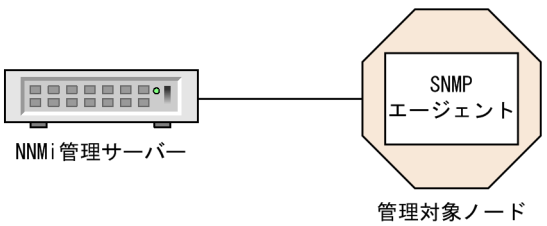
インシデント：SNMPAgentNotResponding インシデントが発生しました。

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，「付録 B.6(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。

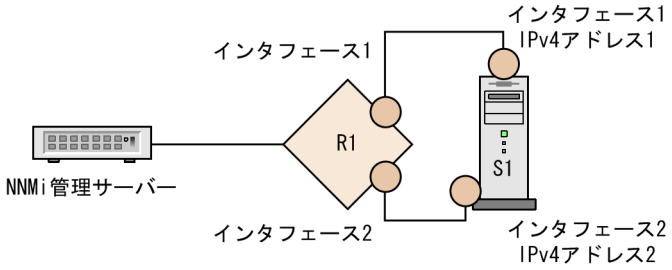
インシデント：発生なし。SNMPAgentNotResponding インシデントがクローズしました。

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。

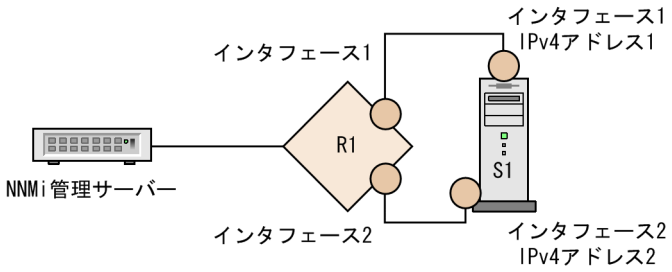
インシデント：AddressNotResponding インシデントが発生しました。

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，「付録 B.6(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。

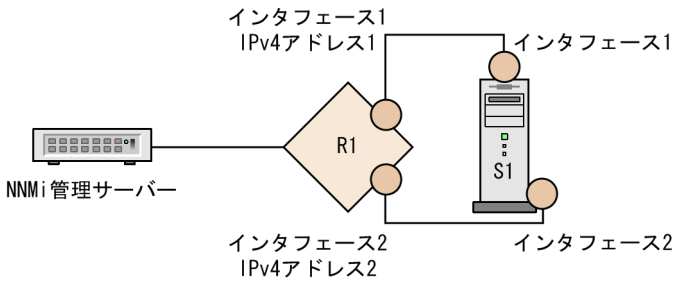
インシデント：発生なし。AddressNotResponding インシデントがクローズしました。

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。

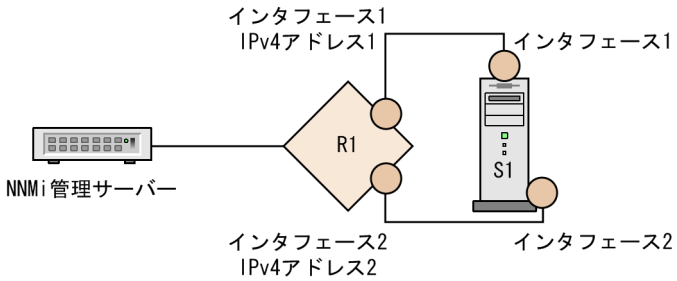
インシデント：InterfaceDown インシデントが発生しました。LinkDown インシデントがInterfaceDown インシデントの下に相関付けされています。

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，「付録 B.6(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。

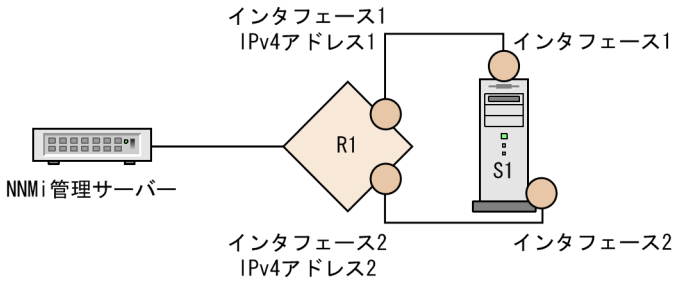
インシデント：発生なし。InterfaceDown インシデントがクローズしました。

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。

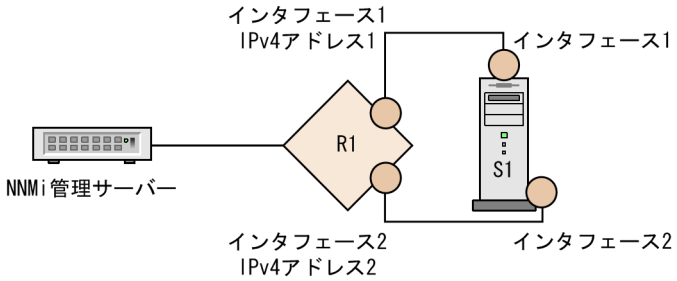
インシデント：発生なし。

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，「付録 B.6(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。

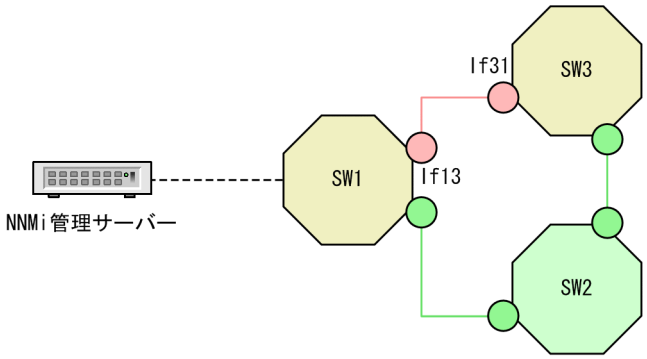
インシデント：発生なし。

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。

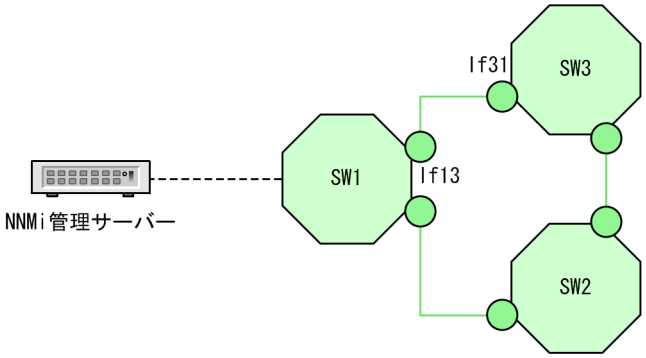
根本原因：IF13 と IF31 の間の接続が停止しています。

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，「付録 B.6(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。

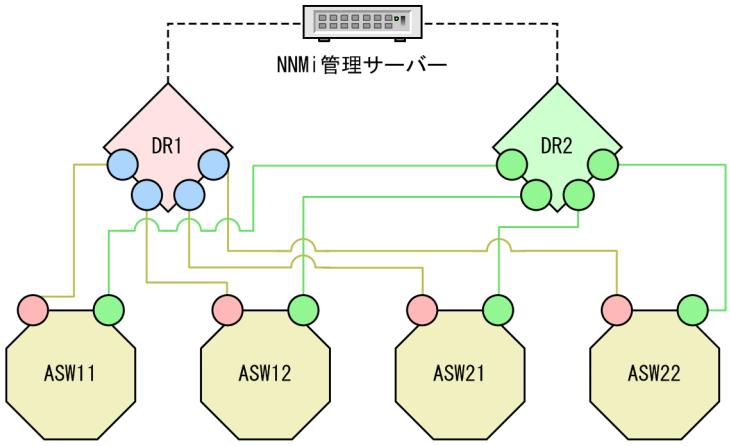
根本原因：IF13 と IF31 の間の接続が稼働しています。

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。

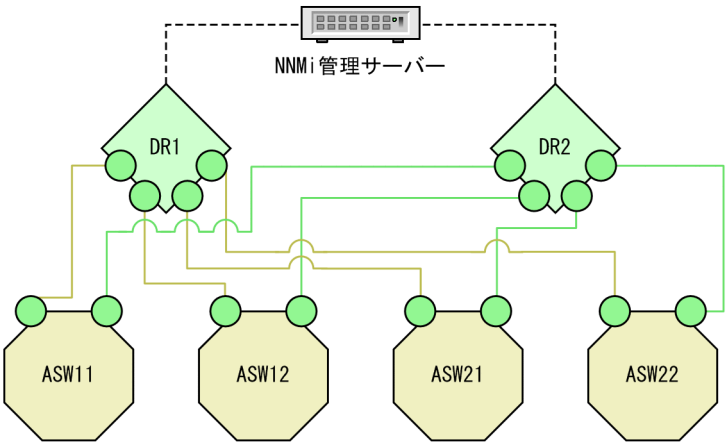
根本原因：ノード DR1 が隣接解析に従って停止しています。

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，「付録 B.6(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。

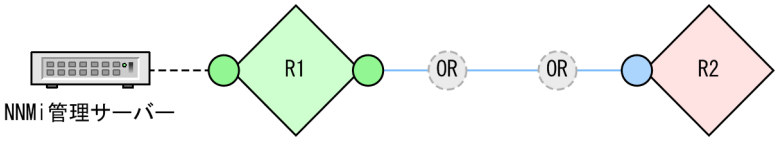
根本原因：ノード DR1 は稼働しています。

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

> **参考**
>
> 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。

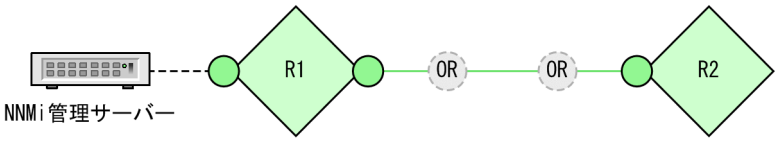
根本原因：ルーター R2 が隣接解析に従って停止しています。

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

> **参考**
>
> 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，「付録 B.6(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。

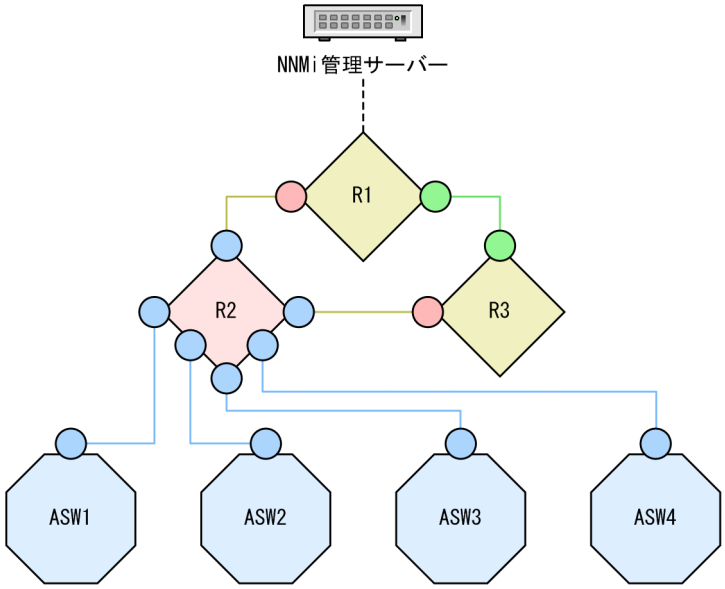
根本原因：R1 と R2 の間の接続が稼働しています。

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。

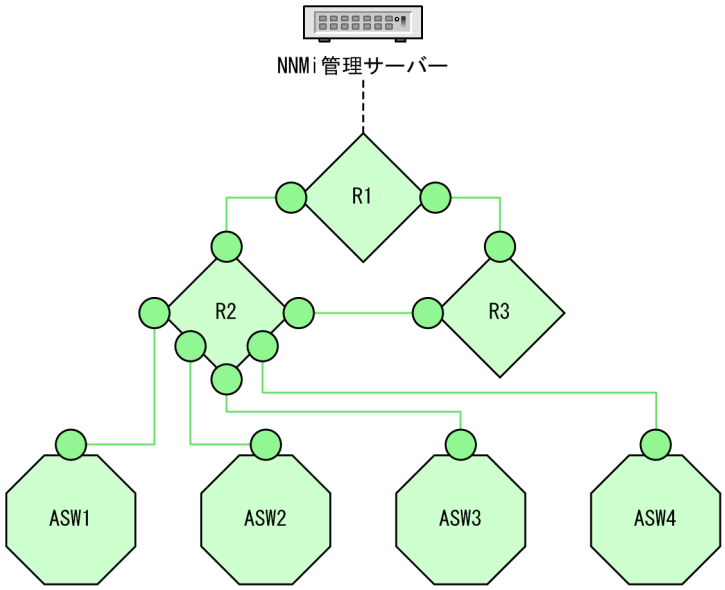
インシデント：NodeDown インシデントが発生しました。1 ホップネイバーからのInterfaceDown インシデントがNodeDown インシデントの下に相関付けされます。

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，「付録 B.6(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。

参考

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，「付録 B.6(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。

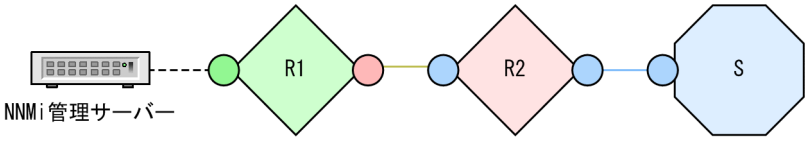
根本原因：ノードは稼働しています。

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。

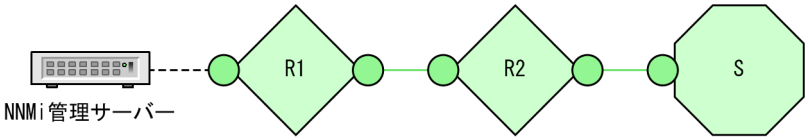
根本原因：ノードまたは接続は停止しています。

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，「付録 B.6(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。

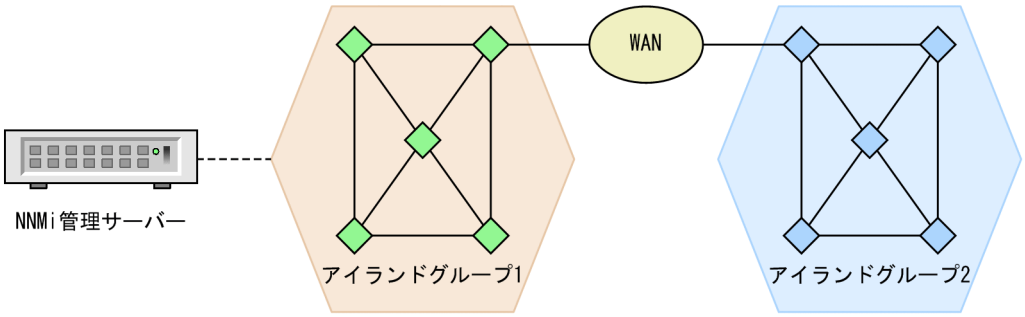
根本原因：NodeUp

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



> **参考**
>
> 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。

参考

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。

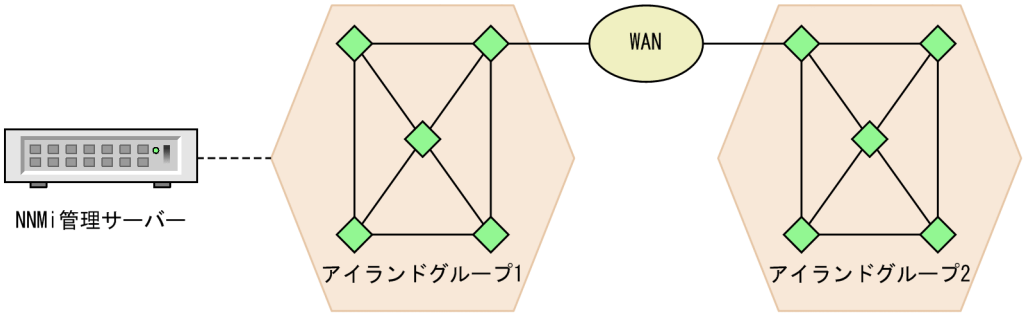
根本原因：アイランドグループ 2 が隣接解析に従って停止しています。

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



参考

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，「付録 B.6(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）

(a)　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。

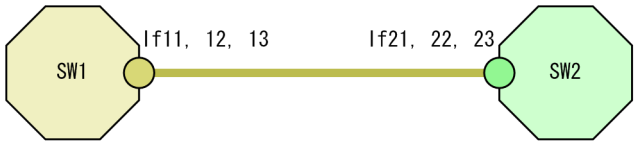
根本原因：すべての操作および管理の状態が動作中です。

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

(b)　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

(c)　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）

(a) 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

(b) 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

(c)　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。

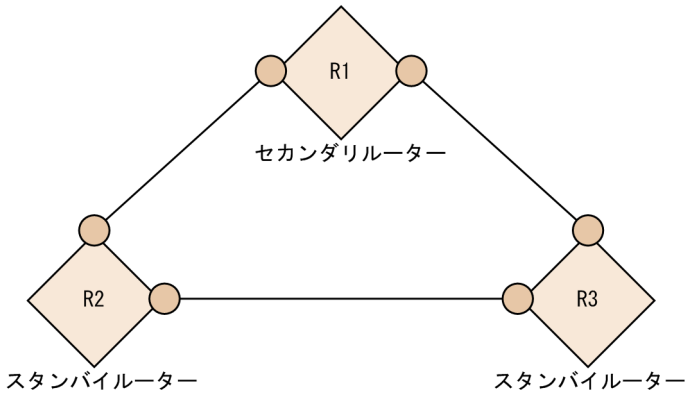
インシデント：AggregatorLinkDown インシデントが生成されます。

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）

(a)　ルーター冗長グループにプライマリがない



（説明）
R1：ルーター1（セカンダリルーターとして動作中）
R2：ルーター2（スタンバイ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。

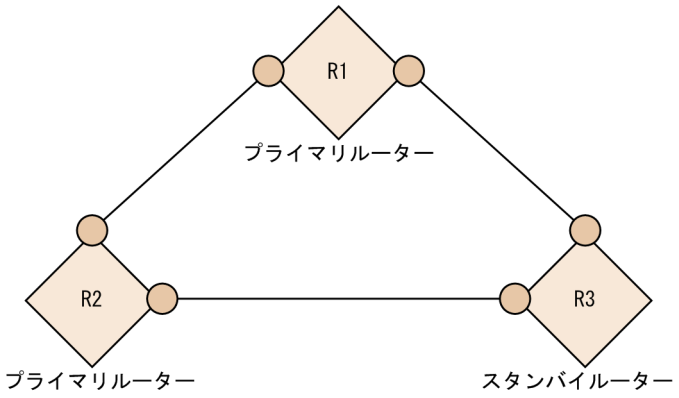
根本原因：このシナリオは，セカンダリルーターがアクティブでない場合にプライマリルーターのインタフェースに障害が発生していたか，ルーター冗長グループの設定に誤りがあったことが原因である可能性があります。

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

(b) 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。

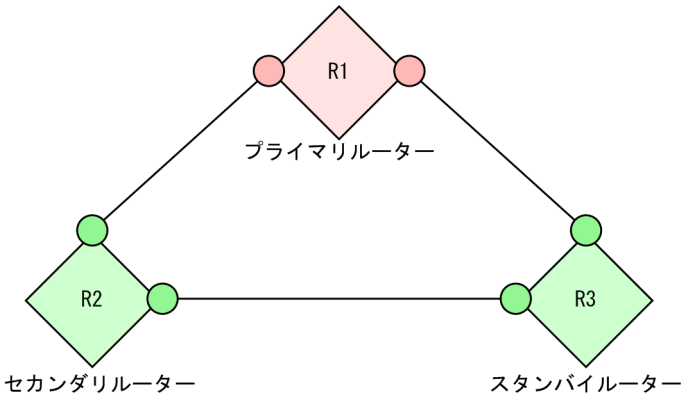
根本原因：このシナリオは，ルーター冗長グループの設定の誤りが原因である可能性があります。

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

(c) 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。

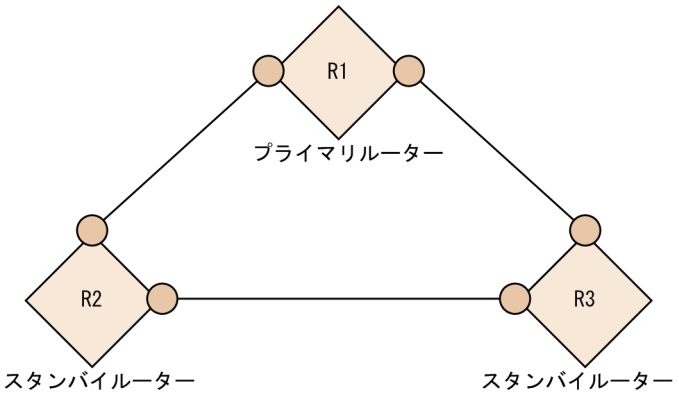
根本原因：このシナリオはプライマリルーターの障害が原因である可能性が最も高いです。

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

(d) 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。

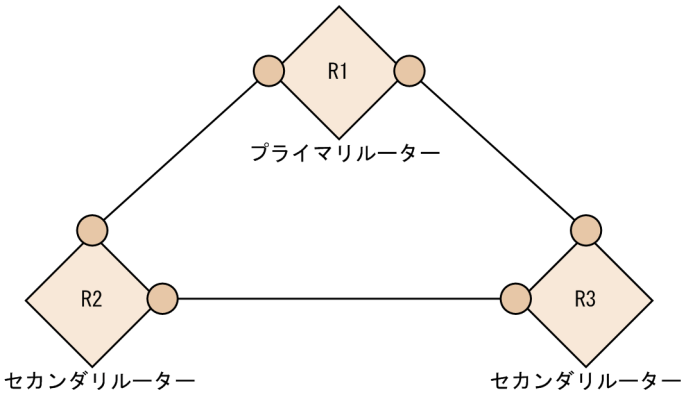
根本原因：このシナリオは，ルーターのインタフェースの障害か，ルーターグループの何らかの設定ミスが原因である可能性があります。

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

(e) 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。

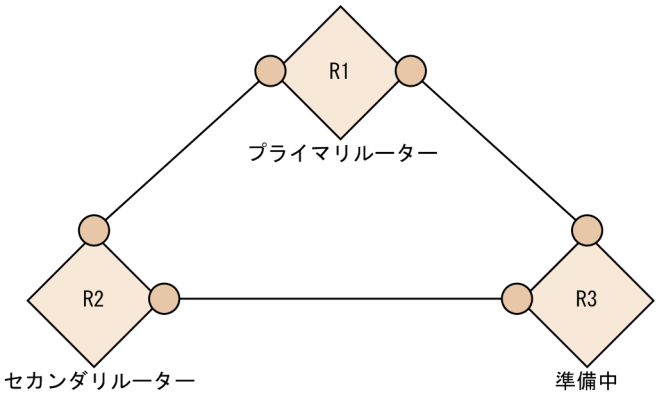
根本原因：このシナリオは，ルーター冗長グループの設定ミスが原因である可能性があります。

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

(f) 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

(a) 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

(b) 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

(c) 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

(d) 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 B.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。

> **参考**
>
> 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 B.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 C　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。

注意事項

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 C-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
80	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 • UNIX $NNM_CONF/nnm/props/nms-local.properties
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。
443	TCP	nmsas.server.port.web.https	デフォルトのセキュア 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。 • Windows %NNM_CONF%¥nnm¥props¥nms-local.properties • UNIX $NNM_CONF/nnm/props/nms-local.properties

ポート	タイプ	名称	目的	設定の変更
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 • UNIX $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 • UNIX $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 • UNIX $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 • UNIX $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 • UNIX $NNM_CONF/nnm/props/nms-local.properties

ポート	タイプ	名称	目的	設定の変更
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 • UNIX $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 • UNIX $NNM_CONF/nnm/props/nms-local.properties
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 • UNIX $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 • UNIX $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 • UNIX $NNM_CONF/nnm/props/nms-local.properties

ポート	タイプ	名称	目的	設定の変更
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 • UNIX $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 • UNIX $NNM_CONF/nnm/props/nms-local.properties
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 • UNIX $NNM_CONF/nnm/props/nms-cluster.properties
8886	TCP	OVsPMD_MGMT	NNMi ovspmd（プロセスマネージャ）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 • UNIX /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）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。

ポート	タイプ	名称	目的	設定の変更
				2. services ファイルを開きます。 • Windows %Windir%¥system32¥drivers¥etc¥services • UNIX /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。

（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 C-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
NNMi コンソール	80／tcp	• NNMi←Web ブラウザ • NNMi（グローバルマネージャ）→NNMi（リージョナルマネージャ）
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
SNMP トラップ転送	162／udp	NNMi→SNMP マネージャ 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 NNMi コンソール	443／tcp	• NNMi←Web ブラウザ • NNMi（グローバルマネージャ）→NNMi（リージョナルマネージャ）
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）→NNMi（リージョナルマネージャ）

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
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）→NNMi（リージョナルマネージャ）
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）

(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，「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」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 C-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 C-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	TCP ポート
非 SSL	nmsas.server.port.web.http	80
	nmsas.server.port.hq	4457
SSL	nmsas.server.port.web.https	443
	nmsas.server.port.hq.ssl	4459

付録 D　各バージョンの変更内容

付録 D.1　10-50 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ました。
- 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ました。
- 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ました。
- NNMi Northbound インタフェースの説明を追加しました。
- 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ました。
- 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ました。
- 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ました。
- 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ました。
- 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ました。
- 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ました。
- HA クラスタのソフトウェアとして，Symantec Cluster Server（SCS）を追加しました。
- HA 設定の注意事項を追加しました。
- WSFC の各リソースの設定内容の例を追加しました。
- 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ました。
- 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ました。
- 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ました。
- 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ました。
- 次の NNMi セキュリティの説明を追加しました。
  - 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 D.2　10-10 の変更内容

- 作成者属性の使用方法の説明を変更しました。
- メニューおよびメニュー項目の設定の説明を変更しました。

- 監視設定をモニタリングの設定に変更しました。
- 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ました。
- 次の SNMP 通信の説明を追加しました。
  - 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ました。
- 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ました。
- 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ました。
- 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ました。
- 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ました。
- NNMi ステータスポーリングで次の項目を変更しました。
  - 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ました。
  - 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ました。
  - 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ました。
  - 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ました。
- 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ました。
  - 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ました。
- NAT 環境の設定方法を追加しました。
- 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ました。
- 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ました。
- 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ました。
- 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ました。
- 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ました。
  - 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ました。
- 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ました。
- NNMi IPv6 管理機能で次の説明を追加または変更しました。
  - 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ました。
- 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ました。
  - 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ました。
  - 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ました。
- NNMi の保守で次の説明を追加または変更しました。
  - 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ました。

- バージョン 9・10-00 の NNMi からの移行で次の説明を追加しました。
  - 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ました。
  - 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ました。
- NNMi が使用するポートの一覧を変更しました。

付録 E　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 E.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 10 JP1/Cm2/Network Node Manager i　インストールガイド（3021-3-241）
- JP1 Version 10 JP1/Cm2/Network Node Manager i　Developer's Toolkit ガイド（3021-3-243）

付録 E.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)
HP-UX	HP-UX (IPF)	HP-UX 11i V3 (IPF)
Linux	Linux 6	Red Hat Enterprise Linux(R) Server 6（64-bit x86_64）
NNMi	NNMi	JP1/Cm2/Network Node Manager i
	NNMi Advanced	JP1/Cm2/Network Node Manager i Advanced
Solaris		Solaris 10(SPARC)

HP-UX，Solaris，および Linux を総称して，UNIX と表記することがあります。

付録 E.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
CIA	Custom Incident Attribute

このマニュアルでの表記	正式名称
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
NOC	Network Operations Center
SCS	Symantec Cluster Server※
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
VCS	Veritas Cluster Server※
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 E.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。

付録 E.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 F　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 Services 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。

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/Cm2/Network Node Manager i および JP1/Cm2/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です。ネットワークノードの継続検出，イベントの監視，および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 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 コンソールを使って 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

「アクティブなサーバー」を参照してください。

アクティブなサーバー

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。

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（ユーザーが設定し，jboss クラスタリングサポートを利用）。

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。ICMP ping は，ステータスポーリング用の SNMP クエリーとともに NNMi が使います。

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，MIB データオブジェクトを識別する数字のシーケンス。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 階層のそのレベルでの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 OID1.3.6.1.2.1.15.0.1 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。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，管理対象ノード上のエージェントプロセッサ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，SNMPv2 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，管理対象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。管理情報ベース内のデータオブジェクトは管理対象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）を使って，管理対象ノードとの間で SNMP クエリーを出し，SNMP トラップを受け取ります。

く

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を持つ，NNMi の分散型の配備です。

グローバルマネージャ

分散 NNMi リージョンマネージャ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，またはそのほかの検出プロトコルクエリー，または Ping スイープを使用して NNMi が見つけた IP アドレス。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の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

ある範囲のユーザー定義 IP アドレスまたはシステムオブジェクト ID（OID），もしくはその両方などルールベースの検出プロセスを制限するのに使われるルールのことです。検出ルールは，NNMi コンソールの **[検出の設定]** の **[自動検出ルール]** 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を指します。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と比べてください。

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。ルールベースの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に基づいた検出に制限されます。

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

ユーザーまたはユーザーグループのために 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

「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

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。「ルールベースの検出」と読み比べてください。「検出プロセス」と「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い，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の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候補をチェックします。検出ルールは，NNMi コンソールの **[検出の設定]** の **[自動検出ルール]** 部分に設定します。「リストに基づいた検出」と読み比べてください。

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

記号

A

C

D

H

I

O

P

R

S

V

W

X

あ

い

え

お

か

き

く

け

こ

さ

し

す

せ

と

な

に

ね

の

は

ひ

ふ

へ