

# HP Network Node Manager i Software

Windows<sup>®</sup>、HP-UX、Linux、および Solaris オペレーティングシステム用

ソフトウェア バージョン : NNMi 9.20

---

## デプロイメントリファレンス

ドキュメントリリース日 : 2012 年 5 月  
ソフトウェアリリース日 : 2012 年 5 月



## ご注意

### 保証について

HP 製品とサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載で追加保証を意図するものは一切ありません。HP では、ここに記載されている技術的、または編集上の不正確さや脱漏については責任を負いません。

ここに記載されている情報は、予告なく変更されることがあります。

### 権利制限について

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HP が提供する有効なライセンスが必要です。FAR 12.211 および 12.212 に準拠し、商用コンピューターソフトウェア、コンピューターソフトウェアドキュメント、および商用アイテムの技術データは、ベンダーの標準商用ライセンスの下、米国政府にライセンスされています。

### 著作権について

© Copyright 2008–2012 Hewlett-Packard Development Company, L.P.

### 商標に関する通知

Acrobat® は Adobe Systems Incorporated の登録商標です。

HP 9000 コンピューター上の 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。

Microsoft® および Windows® は、Microsoft Corporation の米国登録商標です。

Oracle および Java は Oracle およびその関連会社の登録商標です。

UNIX® は The Open Group の登録商標です。

### Oracle テクノロジーの制限された権限に関する通知

国防省連邦調達規則補足 (DOD FAR Supplement) に従って提供されるプログラムは、「商用コンピューターソフトウェア」であり、ドキュメントを含む同プログラムの使用、複製および開示は、該当する Oracle 社のライセンス契約に規定された制約を受けるものとします。それ以外の場合は、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ーソフトウェア」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、『商用コンピューターソフトウェア - 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 America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Oracle ライセンスの全文は、NNMi の製品 DVD 上にある license-agreements のディレクトリを参照してください。

### 謝辞

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>)

## 使用可能な製品ドキュメント

このガイドに加え、次のドキュメントが NNMi について利用できます。

- **HP Network Node Manager i Software** ドキュメント一覧 — HP マニュアル Web サイト上にあります。このファイルを使用して、このバージョンの NNMi の NNMi ドキュメントセットにある追加や改訂を調べることができます。リンクをクリックして、HP マニュアル Web サイト上のドキュメントにアクセスします。
- **HP Network Node Manager i Software** インタラクティブインストールガイド — これは対話型ドキュメントで、NNMI 9.20 製品メディアで入手できます。  
詳細については、製品メディアの `nnmi_interactive_installation_ja_README.txt` ファイルを参照してください。
- **HP Network Node Manager i Software** アップグレードリファレンス — HP マニュアル Web サイトから入手できます。このドキュメントには、旧バージョンの NNM および NNMi からのアップグレードに役立つ情報が含まれています。
- 『HP Network Node Manager i Software リリースノート』— 製品メディアおよび NNMi 管理サーバーで入手できます。
- 『HP Network Node Manager i Software システムとデバイス対応マトリックス』— 製品メディアおよび NNMi 管理サーバーから入手できます。
- 『HP Network Node Manager iSPI Network Engineering Toolset 計画とインストールガイド』(HP Network Node Manager iSPI Network Engineering Toolset Planning and Installation Guide) — NNM iSPI NET 診断サーバー製品メディアにあります。

最近の更新を確認する場合、または最新のドキュメントを使用しているか確認する場合は、以下をご覧ください。

**<http://support.openview.hp.com/selfsolve/manuals>**

このサイトを利用するには、HP Passport への登録とサインインが必要です。HP Passport ID の取得登録は、次の Web サイトから行なうことができます。

**<http://h20229.www2.hp.com/passport-registration.html>** (英語サイト)

または、HP Passport のログインページの [**New users - please register**] リンクをクリックします。

製品のサポートサービスに登録すると、最新版を入手できます。詳細は HP 販売員にお尋ねください。

## サポート

次の HP ソフトウェアサポートオンライン Web サイトを参照してください。

**[support.openview.hp.com](http://support.openview.hp.com)**

この Web サイトには、製品、サービス、および HP Software が提供するサポートの問い合わせ情報および詳細が記載されています。

HP ソフトウェアオンラインサポートには、お客様の自己解決機能が備わっています。ビジネスを管理するために必要な対話形式のテクニカルサポートツールにアクセスする迅速で効率的な方法が用意されています。お客様は、サポート Web サイトで以下の機能を利用できます。

- 関心のあるドキュメントの検索
- サポートケースおよび拡張リクエストの送信および追跡
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HP サポートの問合せ先の検索
- 利用可能なサービス情報の確認
- ソフトウェアを利用している他のユーザーとの情報交換
- ソフトウェアトレーニング情報の検索および参加登録

一部を除き、サポートのご利用には、HP Passport ユーザーとしてご登録の上、ログインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport ユーザー ID のご登録は、以下の URL で行ってください。

**<http://h20229.www2.hp.com/passport-registration.html>** (英語サイト)

アクセスレベルに関する詳細は、次の URL で確認してください。

**[http://support.openview.hp.com/access\\_level.jsp](http://support.openview.hp.com/access_level.jsp)**

# 目次

<b>このガイドについて</b> .....	<b>19</b>
ガイドの説明 .....	19
このドキュメントで使用するパス表記 .....	20
改訂履歴 .....	21
NNMiの詳細 .....	22
<b>準備</b> .....	<b>25</b>
<b>ハードウェアとソフトウェアの要件</b> .....	<b>27</b>
対応ハードウェアとソフトウェア .....	27
必要なパッチの確認 .....	28
システム設定 (UNIX) .....	29
NNMi および NNM iSPI のインストール .....	29
NNMi と HP Performance Insight の共存 .....	29
NNMi と HP Operations エージェントの共存 .....	30
NNMi 9.1x と NNM iSPI Performance for Metrics のバージョン要件 .....	30
<b>設定</b> .....	<b>31</b>
<b>設定の一般概念</b> .....	<b>33</b>
タスクフローモデル .....	33
ベストプラクティス: 既存の設定を保存 .....	34
ベストプラクティス: 作成者属性を使用する .....	34
ユーザーインターフェースモデル .....	34
順序 .....	35
ノードグループおよびインターフェースグループ .....	35
グループの重複 .....	36
ノードグループのメンバーシップ .....	37
階層 / コンテンメント .....	37
デバイスフィルター .....	38
追加フィルター .....	38
追加ノード .....	39
ノードグループのステータス .....	39
インターフェースグループ .....	39
ノード / インターフェース / アドレス階層 .....	40
すべてを停止して再度やり直す .....	40

<b>NNMi 通信</b> .....	<b>43</b>
通信の概念.....	44
通信の設定レベル.....	44
ネットワーク待ち時間とタイムアウト .....	45
SNMP アクセス制御 .....	45
SNMP バージョンの優先.....	46
管理アドレスの優先.....	47
ポーリングプロトコル .....	47
通信設定および <code>nnmsnmp*.ovpl</code> コマンド.....	48
通信の計画作成.....	48
デフォルトの通信設定 .....	49
通信設定領域 .....	49
特定のノードの設定.....	50
再試行とタイムアウトの値 .....	50
アクティブなプロトコル .....	50
複数のコミュニティ文字列または認証プロファイル.....	51
SNMPv1 と SNMPv2 のコミュニティ文字列 .....	51
SNMPv3 の認証プロファイル .....	51
通信の設定.....	52
SNMP プロキシの設定 .....	52
通信の評価.....	54
すべてのノードが SNMP 用に設定されましたか?.....	54
デバイスについて SNMP アクセスは現在利用できますか?.....	54
管理 IP アドレスは正しいですか? .....	54
NNMi は正しい通信設定を使っていますか? .....	55
State Poller 設定は通信設定と一致していますか?.....	55
通信の調整.....	55
<b>NNMi 検出</b> .....	<b>57</b>
検出の概念.....	57
NNMi はデバイスのプロファイルルから属性を導き出す.....	59
検出の計画.....	59
基本的な検出方法を選択する.....	59
リストに基づいた検出 .....	60
ルールベースの検出 .....	60
自動検出ルール .....	61
自動検出ルールの順序 .....	61
デバイスを検出から除外.....	61
Ping スweep .....	62
SNMP トラップからの検出ヒント.....	62
自動検出ルールの検出シード.....	62
自動検出ルールのベストプラクティス.....	63
例.....	63
ノード名の解決 .....	63

サブネット接続ルール	64
検出シード	64
再検出の間隔	65
オブジェクトを検出しない	65
インタフェースの検出範囲	66
NNMiによる仮想IPアドレスの監視	66
検出の設定	67
自動検出ルールを設定する場合のヒント	68
シードを設定する場合のヒント	68
検出の評価	69
初期検出の進行状況をたどる	69
すべてのシードが検出されているか?	69
すべてのノードには有効なデバイスのプロファイルがあるか?	70
すべてのノードが正しく検出されたか?	70
自動検出ルール	70
IPアドレス範囲	71
システムオブジェクトIDの範囲	71
すべての接続とVLANは正しいか?	71
レイヤー2接続の評価	71
NNMi検出と重複MACアドレス	72
デバイスを再検出する	72
検出の調整	72
検出ログファイル	73
無番号インタフェース	73
無番号インタフェース機能の有効化	73
無番号インタフェース機能の無効化	75
非応答オブジェクトの削除の制御	76
<b>NNMi 状態ポーリング</b>	<b>77</b>
状態ポーリングの概念	77
状態ポーリングの計画を作成	78
ポーリングチェックリスト	78
NNMiで何を監視できますか?	79
監視されないノードへのインタフェース	80
監視の停止	81
グループの計画作成	81
インタフェースグループ	82
ノードグループ	82
ポーリング間隔の計画作成	83
どのデータを収集するか	84
状態ポーリングの設定	85
インタフェースグループとノードグループの設定	85
インタフェースのモニタリングの設定	85
ノードのモニタリングの設定	86

デフォルト設定の確認	86
状態ポーリングの評価	87
ネットワークモニタリングの設定を確認します。	87
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？	87
どの設定が適用されていますか？	87
どのデータが収集されていますか？	88
ステータスのポーリングのパフォーマンスの評価	88
<b>State Poller</b> は最新の状態に付いていっていますか？	88
状態ポーリングの調整	90

## NNMi インシ

## デント..... 91

インシデントの概念	91
インシデントライフサイクル	92
トラップおよびインシデント転送	94
比較: サードパーティ <b>SNMP</b> トラップを別のアプリケーションに転送する	95
<b>MIB</b>	96
カスタムインシデント属性	97
解決済み管理イベントインシデントに追加される <b>CIA</b>	98
インシデント数の削減	99
インシデントの抑制、強化、およびダンプニング	99
ライフサイクル移行アクション	100
インシデントの計画	101
<b>NNMi</b> が処理するデバイストラップ	101
<b>NNMi</b> で表示するインシデント	101
インシデントに対する <b>NNMi</b> の対応方法	101
<b>NNMi</b> による <b>NNM</b> 管理ステーションからのトラップ受信の可否	101
<b>NNMi</b> による別のイベントレシーバーへのトラップ転送の可否	101
インシデントの設定	102
インシデントの抑制、強化、およびダンプニングの設定	102
ライフサイクル移行アクションの設定	102
トラップログの設定	103
インシデントログの設定	103
トラップサーバープロパティの設定	103
インシデント設定のバッチロード	105
<b>nnmincidentcfgdump.ovpl</b> によるインシデント設定ファイルの生成	105
<b>nnmincidentcfgload.ovpl</b> によるインシデント設定のロード	105
インシデントの評価	106
インシデントの調整	106
未定義トラップのインシデントの有効化および設定	107

## NNMi コンソール.....109

ノードグループの実例的使用例	109
----------------	-----



ノードグループの作成	110
ステップ 1: <b>My Network</b> ノードグループを作成する	110
ステップ 2: <b>USA</b> ノードグループを作成する	111
ステップ 3: フィルターを使用して <b>Colorado</b> ノードグループを作成する	111
ステップ 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ー結果を確認する	112
ステップ 5: <b>My Network</b> ノードグループのノードグループ階層を設定する	112
ステップ 6: <b>USA</b> ノードグループのノードグループ階層を作成する	112
ノードグループマップの設定	113
ステップ 1: ノードグループマップを作成する	113
ステップ 2: ノードグループマップを表示する	113
ステップ 3: ノードグループのステータスを設定する	113
ステップ 4: ノードグループマップの順序を設定する	114
ステップ 5: ノードグループマップに背景イメージを追加する	114
ノードグループの削除	115
ステップ 1: ノードグループに移動する	115
ステップ 2: ノードグループを削除する	115
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の削減	115
ノードグループマップの表示ノード数の削減	116
[分析] ペインのゲージの設定	117
表示されるゲージ数の制限	117
[分析] ペインにあるゲージの更新間隔の設定	117
ゲージの非表示	117
表示されるノードゲージの順序の制御	118
表示されるインタフェースゲージの順序の制御	118
表示されるカスタムポラーゲージの順序の制御	118
ゲージプロパティの適用方法の理解	118
ゲージ名の判別	119
ゲージに関する問題のトラブルシューティング	119
ゲージが多すぎる	119
ゲージが表示されない	119
デバイスのプロファイルルアイコンのカスタマイズ	120

## 詳細設定

121

### NNMi のライ

#### センス

恒久ライセンスキーのインストール準備	123
ライセンスの種類および管理対象ノードの数の確認	123
恒久ライセンスキーの取得およびインストール	124
<b>Autopass</b> および <b>HP</b> 注文番号の使用 (ファイアウォール使用時は不可)	124
コマンド行で、シードを追加する	124
追加のライセンスキーを取得する	125

<b>NNMi での証明書の使用</b> .....	<b>127</b>
すべてをまとめる .....	128
認証機関証明書を作成する .....	129
自己署名証明書を使用するようにアプリケーションフェイルオーバーを設定する .....	132
認証機関を使用するようにアプリケーションフェイルオーバーを設定する .....	134
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を設定する .....	136
自己署名証明書を使用するように高可用性を設定する .....	136
新規証明書を使用するように高可用性を設定する .....	136
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 .....	137
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 .....	138
自己署名証明書を使用するようにアプリケーションフェイルオーバーが有効なグローバルネットワーク管理 を設定する .....	139
ディレクトリサービスへの SSL 接続を設定する .....	140
<b>NNMi とシングルサインオンの 使用</b> .....	<b>143</b>
NNMi への SSO アクセス .....	144
1 つのドメインに対する SSO の有効化 .....	145
異なるドメインに配置されている NNMi 管理サーバーに対する SSO の有効化 .....	145
NNMi と NNM iSPI の SSO アクセス .....	146
SSO の無効化 .....	148
SSO セキュリティに関する注意 .....	148
<b>NNMi で使用する Telnet および SSH プロトコルを設定する</b> .....	<b>151</b>
Telnet または SSH メニュー項目の無効化 .....	151
Windows 上のブラウザーへの Telnet または SSH クライアントの設定 .....	152
Windows オペレーティングシステム提供の Telnet クライアント .....	154
サードパーティ Telnet クライアント (標準 Windows) .....	155
サードパーティ Telnet クライアント (Windows 上のウィンドウ) .....	156
サードパーティ SSH クライアント (標準 Windows および Windows 上のウィンドウ) .....	157
Linux で Telnet または SSH を使用する Firefox の設定 .....	159
Linux 上の Telnet .....	159
Linux 上のセキュアシェル .....	160
Windows レジストリを変更するファイル例 .....	160
nntelnet.reg の例 .....	161
nnmputtytelnet.reg の例 .....	161
nntelnet32on64.reg の例 .....	161
nntelnet.reg の例 .....	161
<b>NNMi と LDAP によるディレクトリサービスの 統合</b> .....	<b>163</b>
NNMi ユーザーのアクセス情報と設定オプション .....	163
オプション 1: NNMi データベースにすべての NNMi ユーザー情報を保存 .....	164
オプション 2: 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 .....	165

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保存 .....	166
ディレクトリサービスにアクセスする NNMi の設定 .....	167
ディレクトリサービスのアクセス設定を変更し、NNMi のセキュリティモデルをサポートする .....	175
ディレクトリサービスのクエリー .....	178
ディレクトリサービスアクセス .....	178
ディレクトリサービスの情報 .....	178
ディレクトリサービス管理者が所有する情報 .....	182
ユーザー識別 .....	183
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) .....	184
ユーザーグループの識別 .....	186
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) .....	187
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 .....	188
ディレクトリサービス統合のトラブルシューティング .....	189
ldap.properties 設定ファイルリファレンス .....	190
例 .....	195

## NAT 環境の重複 IP アドレスの

管理 .....	197
NAT とは .....	197
NAT の利点 .....	197
サポートされる NAT タイプ .....	198
NNMi に NAT を実装する方法 .....	198
静的 NAT の考慮事項 .....	198
静的 NAT のハードウェアとソフトウェアの要件 .....	199
静的 NAT での通信 .....	200
静的 NAT 環境における管理アドレスの ICMP ポーリングの管理 .....	200
NAT 環境における管理アドレスの ICMP ポーリングの有効化 .....	200
NNMi に対する変更点 .....	200
検出と静的 NAT .....	201
トラップと静的 NAT .....	201
SNMPv2c トラップ .....	202
SNMPv1 トラップ .....	203
サブネットと静的 NAT .....	205
グローバルネットワーク管理と静的 NAT .....	205
動的 NAT および動的 PAT の考慮事項 .....	205
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 .....	207
検出と動的 NAT および動的 PAT .....	207
サブネットと動的 NAT および動的 PAT .....	207
グローバルネットワーク管理と動的 NAT および動的 PAT .....	208
重複する IP アドレスマッピング .....	208
プライベート IP アドレスの範囲 .....	208

## NNMi セキュリティおよびマルチテナント .....

オブジェクトのアクセス制限による影響 .....	212
--------------------------	-----

NNMi セキュリティモデル	213
セキュリティグループ	214
セキュリティグループ構造の例	215
NNMi テナントモデル	218
テナント	218
テナント構造の例	219
NNMi のセキュリティおよびマルチテナント設定	221
設定ツール	222
テナントの設定	224
セキュリティグループの設定	225
設定の確認	227
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	229
NNMi セキュリティ、マルチテナント、およびグローバルネットワーク管理	230
初期 GNM 設定	231
GNM のメンテナンス	232
NPS レポートへの選択インタフェースの追加	232
<b>グローバルネットワーク管理</b>	<b>235</b>
グローバルネットワーク管理の利点	235
グローバルネットワーク管理が自分のネットワークの管理に適しているかどうかを判断するには	236
マルチサイトネットワークを継続的に監視する必要がありますか?	236
重要デバイスを表示できるか?	236
ライセンスの考慮事項	237
実践的なグローバルネットワーク管理の例	237
要件のレビュー	238
リージョナルマネージャーとグローバルマネージャーの接続	239
初期準備	240
ポート可用性: ファイアウォールの設定	240
自己署名証明書の設定	240
グローバルネットワーク管理でアプリケーションフェイルオーバーの設定を行う	240
NNMi 管理サーバー規模の考慮事項	241
システムクロックの同期化	241
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方法	241
グローバルネットワーク管理における自己署名証明書の使用方法	241
グローバルネットワーク管理における認証機関の使用方法	241
監視する重要な機器の一覧作成	242
グローバルマネージャーとリージョナルマネージャーの管理ドメインの検討	242
NNMi ヘルプトピックの確認	243
SSO およびアクションメニュー	243
グローバルネットワーク管理用にシングルサインオンを設定する	243
リージョナルマネージャーでの転送フィルターの設定	245
転送されるノードを制限する転送フィルターの設定	245
グローバルマネージャーとリージョナルマネージャーの接続	252
global1 から regional1 と regional2 への接続ステータスの判定	256

global1 インベントリの確認	257
global1 と regional1 との通信の切断	259
追加情報	261
検出とデータの同期化	261
デバイスのステータスのポーリングまたは設定ポーリング	264
グローバルマネージャーを使ったデバイスステータスの判定と NNMi インシデント生成	265
グローバルネットワーク管理でアプリケーションフェイルオーバーの設定を行う	266
グローバルマネージャーでのアプリケーションフェイルオーバーの設定	266
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	268
NNMi ヘルプのトラブルシューティング情報	268
クロック同期	268
グローバルネットワーク管理システム情報	268
グローバルマネージャーからのリージョナルマネージャー検出の同期化	269
破損した global1 上のデータベースの修復	269
NNMi 9.0x/9.1x からの NNMi 9.20 へのグローバルマネージャーとリージョナルマネージャー のアップグレード	270
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	270
グローバルネットワーク管理のアップグレード手順	270
グローバルネットワーク管理と NNM iSPI または第三者の統合	271
HP Network Node Manager iSPI Performance for Metrics Software	271
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	271
<b>IPv6 用 NNMi Advanced の設定</b>	<b>273</b>
機能説明	273
前提条件	275
ライセンス	275
サポートされる設定	276
管理サーバー	276
IPv6 をサポートする SNMP MIB	277
NNMi のインストール	277
IPv6 機能のアクティブ化	278
IPv6 機能の非アクティブ化	280
非アクティブ化後の IPv6 監視	280
非アクティブ化後の IPv6 インベントリ	280
IPv6 インベントリクリーンアップ時の既知の問題点	282
<b>Solaris ゾーン環境での NNMi の実行</b>	<b>283</b>
Solaris ゾーンでの NNMi のインストール	283
Solaris ゾーンでのトラップ転送	283
Solaris ゾーン環境での NNMi アプリケーションフェイルオーバーの実行	284
Solaris ゾーン環境の HA での NNMi の実行	284

<b>アプリケーションフェイルオーバー構成の NNMi の設定</b> .....	<b>289</b>
アプリケーションフェイルオーバーの概要 .....	290
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 .....	290
アプリケーションフェイルオーバー構成の NNMi の設定 .....	292
NNMi クラスタセットアップウィザードを使用したクラスタの設定 (組み込みデータベースユーザーのみ) .....	293
クラスタ通信の設定 (オプション) .....	294
アプリケーションフェイルオーバー機能の使用 .....	294
組み込みデータベースを使用したアプリケーションフェイルオーバーの動作 .....	295
Oracle データベースを使用したアプリケーションフェイルオーバーの動作 .....	297
アプリケーションフェイルオーバーの例 .....	298
その他の <b>ovstart</b> および <b>ovstop</b> オプション .....	298
アプリケーションフェイルオーバーのインシデント .....	299
フェイルオーバー後、元の設定に戻る .....	299
NNM iSPI およびアプリケーションフェイルオーバー .....	300
NNM iSPI のインストールに関する情報 .....	300
統合アプリケーション .....	301
アプリケーションフェイルオーバーの無効化 .....	302
管理タスクおよびアプリケーションフェイルオーバー .....	304
アプリケーションフェイルオーバーおよび NNMi 9.20 へのアップグレード .....	304
組み込みデータベース .....	304
Oracle データベース .....	307
アプリケーションフェイルオーバーおよび NNMi パッチ .....	309
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) .....	309
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) .....	311
アプリケーションフェイルオーバーおよび NNMi 管理サーバーの再起動 .....	313
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 .....	313
アプリケーションフェイルオーバーおよび以前のデータベースバックアップから復旧 (組み込みデータベースのみ) .....	313
ネットワークレイテンシ/帯域に関する考慮 .....	315
アプリケーションフェイルオーバーと NNMi 組み込みデータベース .....	315
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 .....	316
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 .....	317
<b>高可用性クラスタに NNMi を設定する</b> .....	<b>319</b>
HA の概念 .....	320
HA 用語集 .....	322
NNMi HA クラスタのシナリオ .....	322
マンページ .....	326
HA 用 NNMi を設定するための前提条件の検証 .....	326
高可用性の設定 .....	328

HA 用の NNMi 証明書の設定 .....	328
HA 用の NNMi の設定 .....	329
NNMiHA 設定情報 .....	330
プライマリクラスターノードでの NNMi の設定 .....	333
セカンダリクラスターノードでの NNMi の設定 .....	335
HA 用の NNM iSPIs の設定 .....	336
NNM iSPI Performance for Metrics、NNM iSPI Performance for QA、および NNM iSPI Performance for Traffic .....	336
NNM iSPI for MPLS、NNM iSPI for IP Multicast、および NNM iSPI for IP Telephony .....	337
HA 下で実行中の NNM iSPI ネットワークエンジニアリングツールセットソフトウェアと NNMi	337
Oracle 環境での HA 用の NNMi の設定 .....	338
Oracle での NNMi の依存関係 .....	338
Oracle 環境での HA 用の NNMi の設定 .....	338
共有 NNMi データ .....	340
NNMi の共有ディスク内のデータ .....	340
設定ファイルの複製 .....	341
データレプリケーションの無効化 .....	341
共有ディスクの手動準備 .....	341
SAN または物理的に接続されたディスクの設定 .....	342
ov.conf ファイルへの HA 変数の設定 .....	342
NNMi HA リソースグループへの共有ディスクの移動 .....	343
Windows Server での共有ディスク設定についての注記 .....	343
HA クラスター内の NNMi のライセンス契約 .....	344
HA 設定のメンテナンス .....	345
メンテナンスモード .....	345
HA リソースグループをメンテナンスモードにする .....	345
HA リソースグループのメンテナンスモードを解除する .....	345
HA クラスター内の NNMi のメンテナンス .....	346
NNMi の起動と停止 .....	346
クラスター環境で NNMi のホスト名や IP アドレスを変更する .....	346
フェイルオーバーを行わせないように NNMi を停止する .....	348
メンテナンス後に NNMi を再起動する .....	349
NNMi HA クラスター内のアドオン NNM iSPI のメンテナンス .....	349
HA クラスター内の NNMi の設定を解除する .....	349
既存データベースを使用した HA 外での NNMi 実行 .....	352
HA 下の NNMi のパッチ .....	353
HA 下の NNMi を NNMi 9.0x/9.1x から NNMi 9.20 にアップグレードする .....	354
サポートされるすべてのオペレーティングシステムでの組み込みデータベースを使用した NNMi のア ップグレード .....	354
サポートされるすべてのオペレーティングシステムでの Oracle を使用した NNMi のアップグレード	358
HA 設定のトラブルシューティング .....	359
一般的な設定の誤り .....	359
RHCS 6 での設定の問題 .....	360
HA リソーステスト .....	360

一般的な HA のトラブルシューティング .....	361
エラー: 引数の数が正しくない.....	362
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) .....	362
製品の起動タイムアウト (Solaris) .....	363
製品の起動タイムアウト (Windows MSCS 2008) .....	363
アクティブなクラスターノードのログファイルが更新されない .....	363
NNMi HA リソースグループを特定のクラスターノードで起動できない.....	364
NNMi 固有の HA のトラブルシューティング.....	365
すべてのクラスターノードを設定解除した後の HA 用 NNMi の再有効化.....	365
NNMi を HA 下で正常に起動できない .....	366
NNMi データへの変更がフェイルオーバーの後に表示されない.....	366
HA の設定後、nmsdbmgr を起動できない .....	367
HA の設定後、pmd を起動できない .....	367
NNMi が 1 つの HA クラスターノードでのみ正常に実行される (Windows) .....	368
ディスクフェイルオーバーが行われない.....	368
共有ディスクにアクセスできない (Windows) .....	368
共有ディスクに最新データが含まれない.....	368
フェイルオーバー後にセカンダリノードが共有ディスクファイルを見つけられない.....	368
NNM iSPI 固有の HA のトラブルシューティング .....	369
HA 設定リファレンス .....	370
NNMi HA 設定ファイル .....	370
NNMi に付属している HA 設定スクリプト .....	370
NNMi HA 設定のログファイル.....	372
<b>NNMi Northbound インタフェース .....</b>	<b>375</b>
NNMi Northbound インタフェース.....	376
値 .....	376
サポートされるバージョン .....	376
用語 .....	376
ドキュメント .....	377
NNMi ノースバウンドインタフェースの有効化.....	377
NNMi ノースバウンドインタフェースの使用法.....	378
インシデント転送.....	378
インシデントライフサイクル状態変化通知.....	379
インシデント関連処理通知 .....	380
インシデント削除通知 .....	381
イベント転送フィルター .....	381
NNMi ノースバウンドインタフェースの変更 .....	381
NNMi ノースバウンドインタフェースの無効化.....	382
NNMi ノースバウンドインタフェースのトラブルシューティング .....	382
アプリケーションフェイルオーバーと NNMi ノースバウンドインタフェース.....	383
ローカル Northbound アプリケーション .....	384
リモート Northbound アプリケーション .....	384



[NNMi Northbound Interface デスティネーション] フォームのリファレンス .....	384
Northbound アプリケーションの接続パラメーター .....	385
NNMi Northbound インタフェース統合の内容 .....	386
NNMi Northbound インタフェース転送先のステータス情報 .....	388
NNMi Northbound インタフェースで使用される MIB 情報 .....	389
<b>NNMi のメンテナンス</b> .....	<b>391</b>
<b>NNMi のバックアップおよびリストアツール</b> .....	<b>393</b>
バックアップコマンドとリストアコマンド .....	394
NNMi データのバックアップ .....	394
バックアップタイプ .....	395
バックアップ領域 .....	395
NNMi データのリストア .....	398
同じシステムでのリストア .....	399
異なるシステムでのリストア .....	399
バックアップとリストアの方針 .....	400
すべてのデータを定期的にバックアップする .....	400
設定変更前のデータのバックアップ .....	400
NNMi またはオペレーティングシステムのアップグレード前のバックアップ .....	401
ファイルシステムのファイルのみのリストア .....	401
組み込みデータベースのみをバックアップおよびリストアする .....	401
HA 環境におけるバックアップおよび復元ツールの使用 .....	402
バックアップ .....	402
復元 .....	402
<b>NNMi の保守</b> .....	<b>403</b>
NNMi フォルダのアクセス制御リストの管理 .....	404
カスタムポーラー収集エクスポートの管理 .....	405
カスタムポーラー収集のエクスポートディレクトリの変更 .....	405
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の変更 .....	405
カスタムポーラーメトリックスの累積周期の変更 .....	406
インシデントアクションの管理 .....	407
同時アクション数の設定 .....	407
Jython アクションのスレッド数の設定 .....	407
アクションサーバー名のパラメーターの設定 .....	408
アクションサーバーのキューサイズを変更する .....	409
インシデントアクションログ .....	409
hosted-on-trapstorm.conf ファイルによるトラップストームのブロック .....	410
trapFilter.conf ファイルによるインシデントのブロック .....	410
NNMi の文字セットエンコードの設定 .....	411
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 .....	411
最も古い SNMP トラップインシデントの自動トリム機能の設定 .....	412
最も古い SNMP トラップインシデントの自動トリム機能の有効化 (インシデントアーカイブなし) ..	412
最も古い SNMP トラップインシデントの自動トリム機能の有効化 (インシデントアーカイブ有効) ..	413

保存する SNMP トラップインシデント数の削減	414
最も古い SNMP トラップインシデントの自動トリム機能の監視	414
最も古い SNMP トラップインシデントの自動トリム機能の無効化	414
SNMP MIB 変数名を表示するための NNMi ゲージタイトルの変更	415
NNMi 正規化プロパティの変更	415
初期検出後の正規化プロパティの変更	416
同時 SNMP 要求の変更	417
組み込みデータベースポートの変更	417
MIB ブラウザーパラメーターの変更	418
NNMi 自己監視	419
特定ノードの検出プロトコルの使用を抑える	419
検出プロトコル収集の使用の抑制	420
大規模スイッチの VLAN インデックス付けの使用を抑制する	421
VLAN インデックス付けの使用を抑制する	421
ノードコンポーネントステータスの設定	422
ノードへのノードコンポーネントステータスの伝達	422
ステータスをノードに伝達しないようにノードコンポーネントを設定する	423
ノードコンポーネントのステータス値の上書き	424
<b>NNMi ロギング</b>	<b>425</b>
NNMi ログファイル	425
ロギングファイルのプロパティの変更	426
ロギングのサインインおよびサインアウト	426
<b>NNMi セキュリティ</b>	<b>427</b>
Web アクセスおよび RMI 通信に SSL 通信を設定する	427
非ルート UNIX ユーザーに NNMi の開始と停止を許可する	427
組み込みデータベースツールのパスワードの入力	427
<b>追加情報</b>	<b>429</b>
<b>アプリケーションフェイルオーバー構成の NNMi の手動設定</b>	<b>431</b>
<b>NNMi 環境変数</b>	<b>435</b>
このドキュメントで使用する環境変数	435
他の使用可能な環境変数	435
<b>NNMi 9.20 およびウェルノウンポート</b>	<b>439</b>
<b>NNMi 9.20 iSPI のウェルノウンポート</b>	<b>445</b>
<b>設定変更の提案</b>	<b>455</b>
問題および解決策	455
<b>用語集</b>	<b>461</b>
<b>フィードバックをお待ちしております。</b>	<b>469</b>

# このガイドについて



(1) 最初のインストール  
またはテスト ベッド

NNMi インストール  
ガイドの手順に従っ  
てください



(2) 製品の導入および前バージョンからの移行

NNMi 導入リファレンス  
をご覧ください (このマ  
ニュアル)



この章には、以下のトピックがあります。

- ガイドの説明
- このドキュメントで使用するパス表記
- 改訂履歴
- NNMi の詳細

## ガイドの説明

このガイドには、NNMiやNNMi Advancedなど、HP Network Node Manager i Softwareを導入するための情報およびベストプラクティスが記載されています。対象読者は、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デプロイメントおよび管理に経験のある HP サポートエンジニアです。

このガイドでは、制限のある環境（またはテスト環境）に NNMi をインストール済みであること、クイックスタート設定ウィザードを使用したコミュニティ文字列の設定、ネットワークノードの制限範囲の検出設定、初期管理者アカウントの作成のような、設定作業の開始に慣れていることを仮定しています。これらの作業の詳細は、『HP Network Node Manager i Software インタラクティブインストールガイド』を参照してください（「[使用可能な製品ドキュメント](#)」（3 ページ）を参照）。

新しい情報が入手可能になると、製品リリースの間に、HP はこのガイドを更新します。ドキュメントの更新バージョン取得の詳細は、「[使用可能な製品ドキュメント](#)」（3 ページ）を参照してください。

## このドキュメントで使用するパス表記

このドキュメントには、NNMi bin ディレクトリに配置されているコマンドのコマンドパスは記載されていません。NNMi bin ディレクトリは以下の場所にあります。

- **Windows Server 2008:** <drive>%Program Files (x86)%HP%HP BTO Software%bin
- **UNIX®:** /opt/OV/bin

このドキュメントでは、主に以下の 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。以下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 Server 2008:**
  - %NnmInstallDir%: <drive>%Program Files (x86)%HP%HP BTO Software
  - %NnmDataDir%: <drive>%ProgramData%HP%HP BTO Software



**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



**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、「[他の使用可能な環境変数](#)」(435 ページ)を参照してください。

## 改訂履歴

次の表に、このドキュメントの新規リリースごとの主要な変更をリストします。

ドキュメントリリース日	主要な変更の説明
2011年3月(9.10)	<p>完全に更新。</p> <ul style="list-style-type: none"> <li>英語第4版。</li> <li>日本語第3版。</li> </ul>
2012年5月(9.20)	<ul style="list-style-type: none"> <li>統合情報を抽出し、別のドキュメントを作成しました。</li> <li>NAT環境の重複IPアドレスの管理を追加しました。</li> <li>NNMi 9.0x または 9.1x からのアップグレード情報を、『NNMi アップグレードリファレンス』に移動しました。</li> <li>NNMi セキュリティを追加しました。</li> <li>NNMi クラスタセットアップウィザードを使用したクラスタの設定(組み込みデータベースユーザーのみ)を追加しました。</li> <li>NNMi 9.20 およびウェルノウンポートを更新しました。</li> <li>NNMi 9.20 iSPI のウェルノウンポートを追加しました。</li> </ul>

## NNMi の詳細

NNMi 製品の完全な情報を入手するには、このガイドと他の NNMi ドキュメントと一緒に使用してください。次の表に、現在までのすべての NNMi ドキュメントを示します。ガイドとホワイトペーパーの両方を含みます。



情報はすべて <http://support.openview.hp.com/selfsolve/manuals> からダウンロードできます。詳細については、「使用可能な製品ドキュメント」(3 ページ)を参照してください。

目的	詳しい情報の参照先
このバージョンの NNMi で入手可能な文章の一覧を表示する。	「NNMi ドキュメント一覧」をダウンロードします。このファイルを使用して、このバージョンの NNMi の NNMi ドキュメントセットにある追加や改訂を調べることができます。リンクをクリックして、HP マニュアル Web サイト上のドキュメントにアクセスします。
NNMi または NNMi Advanced をインストール (初回)。	『HP Network Node Manager i Software インタラクティブインストールガイド』をダウンロードします。このガイドには、製品をインストールおよびアンインストールする基本手順、および NNMi クイックスタート設定ウィザードを使用して初期設定を行う方法が記載してあります。 <ul style="list-style-type: none"> <li>『HP Network Node Manager i Software インストールガイド』(Windows オペレーティングシステム用)</li> <li>『HP Network Node Manager i Software インストールガイド』(HP-UX オペレーティングシステム用)</li> <li>『HP Network Node Manager i Software インストールガイド』(Linux オペレーティングシステム用)</li> <li>『HP Network Node Manager i Software インストールガイド』(Solaris オペレーティングシステム用)</li> </ul>
ネットワーク導入の計画 (システム要件へのリンクを含む)。	このガイドの「準備」(25 ページ)を参照してください。
製品環境向けに NNMi を設定する。	このガイドの「設定」(31 ページ)を参照してください。
NNMi の高度設定を行う。	このガイドの「詳細設定」(121 ページ)を参照してください。
NNMi の設定を維持管理する。	このガイドの「NNMi のメンテナンス」(391 ページ)を参照してください。
Network Node Manager i Software の前バージョンから NNMi にアップグレードする。	『HP Network Node Manager i Software アップグレードリファレンス』を参照してください。これは、HP マニュアル Web サイトから入手できます。
NNMi 環境変数、ポート、メッセージのリファレンスを参照する。	このガイドの「追加情報」(429 ページ)を参照してください。
特定のトピックに関する詳細情報を取得する。	サンプルドキュメントやホワイトペーパーからダウンロードします。

目的	詳しい情報の参照先
NNMi ヘルプを印刷する。	ヘルプコンテンツの PDF をダウンロードします。
HP NNM iSPI NET (NNM iSPI NET) 診断サーバーをインストールし、NNM iSPI NET の機能について学ぶ。	Network Node Manager SPI for NET 製品カテゴリから、Windows オペレーティングシステム用の『HP NNM iSPI Network Engineering Toolset Planning and Installation Guide』をダウンロードします。
NNMi 開発者ツールキット (SDK) のドキュメントを入手する。	NNMi のライセンスを参照して、SDK の関連情報、SDK ライセンスの取得およびインストール、SDK のドキュメントおよびサンプルを確認してください。





# 準備

この項では以下の章について説明します。

- ハードウェアとソフトウェアの要件



# ハードウェア とソフトウェア の要件

この章には、以下のトピックがあります。

- 対応ハードウェアとソフトウェア
- 必要なパッチの確認
- システム設定 (UNIX)
- NNMi および NNM iSPI のインストール
- NNMi と HP Performance Insight の共存
- NNMi と HP Operations エージェントの共存
- NNMi 9.1x と NNM iSPI Performance for Metrics のバージョン要件

---

## 対応ハードウェアとソフトウェア

NNMi をインストールする前に、表 1 で説明する NNMi のハードウェアとソフトウェアの要件に関する情報を読んでください。



上記の最新版のドキュメントは、以下から入手してください。

**<http://support.openview.hp.com/selfsolve/manuals>**

表1 ソフトウェアおよびハードウェアのプレインストールのチェックリスト

チェック欄 (はい/いいえ)	確認していただくドキュメント
	HP Network Node Manager i Software インタラクティブインストールガイド <ul style="list-style-type: none"> <li>• ファイル名 = nnmi_interactive_installation_ja.zip または nnmi_interactive_installation_ja.jar</li> <li>• 指示ファイル名 : nnmi_interactive_installation_ja_README.txt</li> <li>• <b>Windows</b> メディア = DVD メインドライブ (root)</li> <li>• <b>UNIX</b> メディア = ルートディレクトリ</li> </ul>
	NNMi リリースノート <ul style="list-style-type: none"> <li>• ファイル名 = releasenotes_ja.html</li> <li>• <b>Windows</b> メディア = DVD メインドライブ (root)</li> <li>• <b>UNIX</b> メディア = ルートディレクトリ</li> <li>• <b>NNMi</b> コンソール = [ヘルプ] &gt; [NNMi ドキュメントライブラリ] &gt; [リリースノート]</li> </ul>
	NNMi システムおよびデバイス対応マトリックス <ul style="list-style-type: none"> <li>• ファイル名 = supportmatrix_ja.html</li> <li>• <b>Windows</b> メディア = DVD メインドライブ (root)</li> <li>• <b>UNIX</b> メディア = ルートディレクトリ</li> <li>• <b>NNMi</b> コンソール = リリースノートからリンクしている</li> </ul>

- ▶ 新しい情報が入手可能になると、HP は『NNMi システムおよびデバイス対応マトリックス』を更新します。NNMi を導入する前に、以下の Web サイトで、お持ちのバージョンのソフトウェアに関する最新の NNMi 対応マトリックスをチェックしてください。

[http://www.hp.com/go/hpsupportsupport/support\\_matrices](http://www.hp.com/go/hpsupportsupport/support_matrices)

(この Web サイトにアクセスするには、HP Passport の ID が必要です。)

- ▶ NNM スマートプラグイン (NNM iSPI) をインストールする場合は、NNMi 導入時に、これらの製品のシステム要件を組み入れてください。

## 必要なパッチの確認

NNMi では、組み込み Java 仮想マシンおよび JDK バージョン 1.6 が出荷されます。Java が正常に動作するには、特定のオペレーティングシステムパッチが必要です。HP-UX オペレーティングシステムを実行しているサーバーに NNMi をインストールする場合、**HPjconfig** コマンドを実行して、必要なパッチがサーバーにインストールされているかどうかを確認できます。**HPJconfig** を実行する場合、JDK バージョン 1.6 に適した選択を行ってください。HP-UX に **HPjconfig** をインストールして実行する方法の詳細については、以下の URL を参照してください。

<https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPJCONFIG>

HP-UX 以外でサポートされているオペレーティングシステムを実行しているサーバーに NNMi をインストールする場合は、そのオペレーティングシステムのリリースノートを参照してください。

---

## システム設定 (UNIX)

NNMi 管理サーバーに NNMi のマン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。

- ▶ NNMi は /etc/opt/OV ディレクトリにある設定ファイルを使用します。このディレクトリは削除しないでください。

---

## NNMi および NNM iSPI のインストール

いずれかの HP NNM iSPI を NNMi とともに使用する場合、HP NNM iSPI をインストールする前に、NNMi をインストールする必要があります。

---

## NNMi と HP Performance Insight の共存

HP Performance Insight と同じサーバーに NNMi をインストールする場合は、次の手順に従って、インストールシーケンスとポート矛盾の問題を回避してください。

- 1 HP Performance Insight を最初にインストールします。

- ▶ 手順 1 と手順 2 を完了してから、NNMi をインストールします。

- 2 HP Performance Insight の全プロセスを停止します。
- 3 NNMi をインストールします。特定の指示については、『HP Network Node Manager i Software インタラクティブインストールガイド』を参照してください。
- 4 次のコマンドで NNMi の全プロセスを停止します。

```
ovstop -c
```

- 5 ポート矛盾を解決するには nms-local.properties ファイルを変更します。このファイルは次のディレクトリにあります。
  - Windows: %NNM\_CONF%\nnm\props
  - UNIX: \$NNM\_CONF/nnm/props
- 6 HP Performance Insight プロセスを開始します。

7 次のコマンドで NNMi の全プロセスを開始します。

```
ovstart -c
```



HP Performance Insight と同じサーバーに NNMi がインストールされている場合に NNMi をアンインストールすると、HP PI MIB ブラウザーを実行したときに例外が発生します。この例外を回避するには、以下の手順を実行します。

1 NNMi をアンインストールします。

2 snmpmib MIB データベースを再作成します。

```
a mkdir -p /var/opt/OV/shared/nnm/conf/
```

```
b /opt/OV/lbin/nnmloadmib -load /usr/OVPI/mibs/GENMIB2IF.mib
```

3 nnmloadmib.ovpl コマンドを使用して、追加の MIB をロードします。

---

## NNMi と HP Operations エージェントの共存

(HP Operations Manager (HPOM) と通信するために ) NNMi 管理サーバーに HP Operations エージェントをインストールする場合は、HP Operations エージェントをインストールする前に NNMi をインストールします。

---

## NNMi 9.1x と NNM iSPI Performance for Metrics のバージョン要件

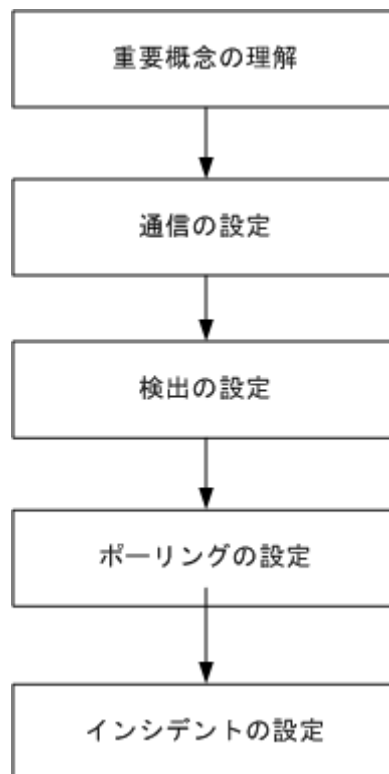
NNMi 9.1x と NNM iSPI Performance for Metrics は、同等のバージョンである必要があります。

- NNM iSPI Performance for Metrics バージョン 9.10 は、NNMi 9.10 でのみサポートされています。
- NNM iSPI Performance for Metrics バージョン 9.11 は、NNMi 9.1x パッチ 1 (9.11) でのみサポートされています。

# 設定

この項では以下の章について説明します。

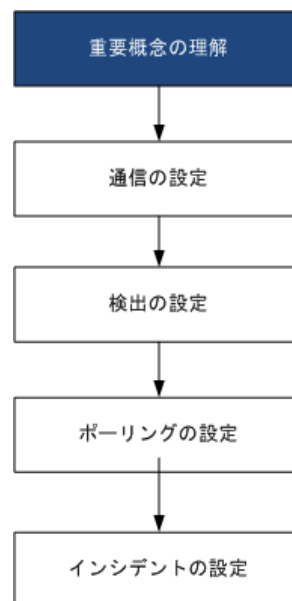
- 設定の一般概念
- NNMi 通信
- NNMi 検出
- NNMi 状態ポーリング
- NNMi インシデント
- NNMi コンソール







# 設定の一般概念



この章では概念の概論を説明しています。詳細については、このガイドの後のほうで説明しています。この章では、すべての HP Network Node Manager i Software (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。

この章には、以下のトピックがあります。

- タスクフローモデル
- ベストプラクティス : 既存の設定を保存
- ベストプラクティス : 作成者属性を使用する
- ユーザーインターフェースモデル
- 順序
- ノードグループおよびインターフェースグループ
- ノード/インターフェース/アドレス階層
- すべてを停止して再度やり直す

## タスクフローモデル

このガイドの設定の各章では、以下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。このガイドの情報は、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マニュアル化を開始または更新するよい機会です。
- 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ーフェースの組み合わせを使用して、設定を NNMi に入力します。具体的な手順については、NNMi ヘルプを参照してください。



コマンドラインインターフェース (PSQL コマンドなど) や外部ユーティリティを使用して、組み込みデータベースの設定を作成、修正、または変更することはできません。これを行おうとすると、データベースに取り返しのつかない損傷を与える可能性があります。

- 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、必要に応じて調節します。
- 5 **調整** — オプション。設定を調整して、NNMi のパフォーマンスを向上します。

---

## ベストプラクティス：既存の設定を保存

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。設定を変更した結果が気に入らなくても、保存した設定に簡単に戻すことができます。

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

これらのコマンドの使用の詳細については、該当するリファレンスページ、または UNIX のマン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` リファレンスページ、または UNIX のマンページを参照してください。

『HP Network Node Manager i Software ステップバイステップガイド (NNMi インポートおよびエクスポートツールの使用に関するホワイトペーパー)』(HP Network Node Manager i Software Step-by-Step Guide to Using NNMi Import and Export Tools White Paper) も参照してください。

---

## 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や変更する際は、作成者の組織がわかる値に [ **作成者** ] 属性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のみを引き出すことができます。

NNMi をアップグレードする際、作成者値が **HP** ではない設定は上書きされません。

---

## 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 コンソールま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 (フォーム上または含まれるフォーム上) 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、終了を取り消すよう求める警告が表示されます。



[ **検出シード** ] フォームは、トランザクションアプローチの例外です。このフォームは便宜上 [ **検出の設定** ] フォーム上にありますが、他の検出設定からは切り離されています。このため、[ **検出の設定** ] フォームを保存して閉じて自動検出ルールを実装した後で、これらの自動検出ルールに検出シードを設定する必要があります。

## 順序

いくつかの 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 が一致を見つけるまで評価し続けます。一致が見つかった時点で、NNMi は一致する設定の情報を使用し、これ以上一致を探すのをやめます。（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他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に遭遇した場合は、その領域の設定の順序を確認してください。

順序はローカルコンテンツ内で適用されます。[メニュー] および [メニュー項目] テーブルには、ローカルコンテキストであるため同じ順序番号の複数のオブジェクトが含まれます。

順序番号は次の箇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] フォームの順序で、関連メニューのローカルコンテキスト内の項目の順序が設定されます。
- [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。

### ベストプラクティス

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最も低い設定に適用します。

### ベストプラクティス

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。たとえば、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。

## ノードグループおよびインタフェースグループ

NNMi の基本的なフィルタリング手法では、ノードまたはインタフェースをグループ化してから、設定をグループに適用または可視化がグループ別にフィルタリングされます。ノードグループは、以下のいずれかまたはすべての目的に使用できます。

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

インタフェースグループは、以下のいずれかまたは両方の目的に使用できます。

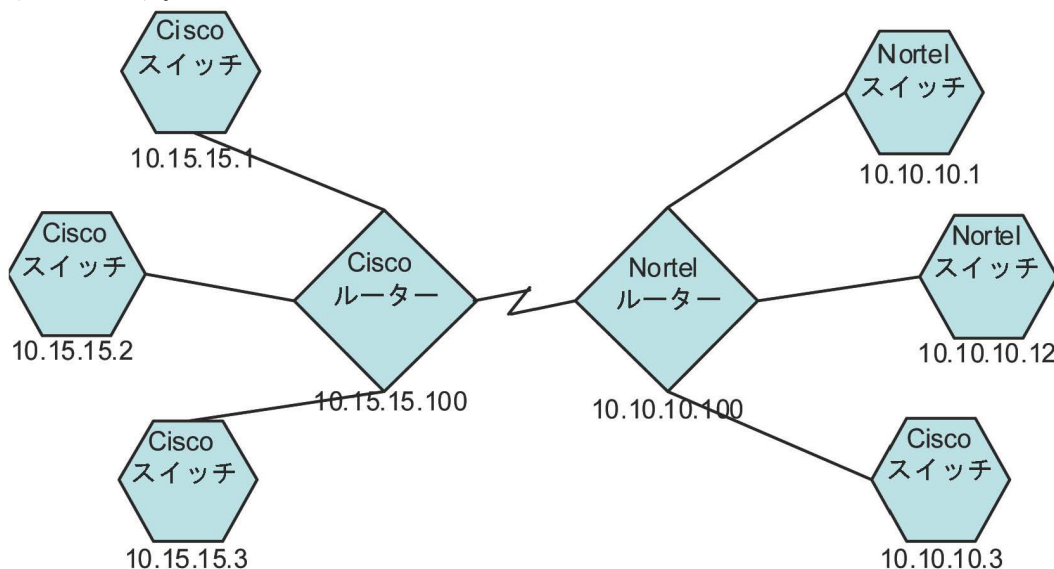
- 検出からのインタフェース除外
- 監視設定
- インシデント負荷量のフィルタリング

- 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。

## グループの重複

グループ定義をどのように使用するにかかわらず、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。様々な目的でグループが作成されるため、各々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ーを使用します。
- 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 アップグレードでは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、ベンダーフィルターを使用します。
- 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ーにグループ化するのがよいでしょう。この場合は、IP アドレスフィルターを使用します。

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ーを一覧に詰め込みすぎることのないよう、バランスをとってください。

## ノードグループのメンバーシップ

NNMi は、検出した各ノードを、設定された各ノードグループと比較することにより、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]** タブで指定したすべてのノードは、ノードグループのメンバーです。



NNMi 管理サーバーのリソースを過度に消費するため、**[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。

- **[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すべてのノードは、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ーに一致するすべてのノードは、そのノードグループのメンバーです。

## 階層 / コンテインメント

単純で再利用可能な原子グループを作成し、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ことにより、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ので、マップビューが大きく向上します。NNMi により、グループの定義とそのドリルダウン順序の徹底管理が可能になります。

単純で再利用可能な原子グループを最初に作成し、その後これらを増築するときの子グループとして指定します。また、最初に一番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。

たとえば、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指定されるので、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、以下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
- まとめてサービス停止にするノードのタイプ
- オペレーターの職務別のノードのグループ

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。



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分することで、設定の継承概念を真似ることができます。

設定インターフェースでは、循環階層の定義が自動的に防御されます。

## 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、そこから他の情報を、デバイスのプロファイルを通じて導き出します。( 詳細については、「[NNMi はデバイスのプロファイルから属性を導き出す](#)」(59 ページ) 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り、NNMi は正しいデバイスのプロファイルを通じて索引化して、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

導出されたこれらの値は、デバイスのプロファイルそのものとともに、フィルターとして使用できます。

たとえば、特定の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(たとえばルーター)をすべて、ベンダーを問わずにまとめることができます。

## 追加フィルター

追加のフィルターエディターを使用すると、以下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` (機能の一意キー)
- `customAttrName` (カスタム属性名)
- `customAttrValue` (カスタム属性値)

フィルターには、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。

機能は、本来は NNMi と統合される他のプログラムを目的としていました。たとえば、ルーター冗長性とコンポーネント稼働状態は、機能(フィールド)を NNMi データベースに追加します。これらの機能は、すでに検出されてデバイスからノード詳細を調べることにより、見ることができます。

iSPI によりカスタム属性を追加したり、独自のカスタム属性を作成できます。Web Services SDK を購入していない方は、各ノードのフィールドに手動で値を入れる必要があります。たとえば資産番号やシリアル番号は属性となりえますが、機能ではありません。

## 追加ノード

ノードグループに対してノードを限定するには、[追加フィルター]を使用することをお勧めします。フィルターを使用して制限することが困難である重要なデバイスがネットワークに含まれている場合、それらのデバイスをホスト名ごとに1つのグループに追加します。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、他に手段がない場合のみにしてください。



NNMi 管理サーバーのリソースを過度に消費するため、[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。

## ノードグループのステータス

そのように設定すると、以下のいずれ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深刻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選択します。
- 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。たとえば、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。(NNMi 8.x からのアップグレードでは、それ以前のステータス計算設定が保持されます。)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することができます。

## 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 アドレスなど他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コンテインメントも継承しませんが、インタフェースをホスト管理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することができます。

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機能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。



## ノード / インタフェース / アドレス階層

NNMi は監視設定を、以下の方式で割り当てます。

- 1 **インタフェース設定** —NNMi は、ノードのインタフェースおよび IP アドレスの各々を、最初に一致する**インタフェース設定**定義に基づいてモニターします。最初に一致するのは、順序番号が最も小さい**インタフェース設定**定義です。
  - 2 **ノード設定** —NNMi によって、各ノードと前回一致しなかった各インタフェースまたは IP アドレスが、最初に一致する**ノード設定**定義に基づき監視されます。最初に一致するのは、順序番号が最も小さい**ノードの設定**定義です。
- ▶ 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（たとえば、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（たとえば、親 =20、子 =10）。
- 3 **デフォルト設定** — **手順 1** または **手順 2** のノード、インタフェース、IP アドレスに一致が見つからない場合、NNMi ではデフォルトの監視設定が適用されます。

## すべてを停止して再度やり直す

検出を完全に再スタートして NNMi 設定のすべてのやり直したい場合、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。このプロセスにより、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。

この手順で説明しているコマンドの詳細は、該当する参照ページか UNIX のマンページを参照してください。

以下の手順に従ってください。

- 1 NNMi サービスを、次のコマンドを使用して停止します。

```
ovstop -c
```

- 2 オプション。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でしょう。

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

- 3 オプション。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 ファイルに出力します。



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、または UNIX のマンページを参照してください。

- 4 オプション。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。インシデントは、nnmtrimincidents.ovpl のリファレンスページ、または UNIX のマンページに記載されているように CSV 形式でアーカイブされます。



- 5 NNMi データベースを削除して再作成します。
  - 組み込みデータベースの場合は、次のコマンドを実行します。

```
nnmresetembdb.ovpl -nostart
```
  - Oracle データベースの場合は、Oracle データベース管理者に NNMi データベースの削除と再作成を依頼してください。データベースインスタンス名は、削除せずに保持してください。
- 6 iSPI または NNMi と統合されるスタンドアロン製品をインストールした場合は、これらの製品をリセットして古いトポロジ識別名を削除します。具体的な手順については、製品のマニュアルを参照してください。
- 7 NNMi サービスを、次のコマンドを使用して開始します。

```
ovstart -c
```

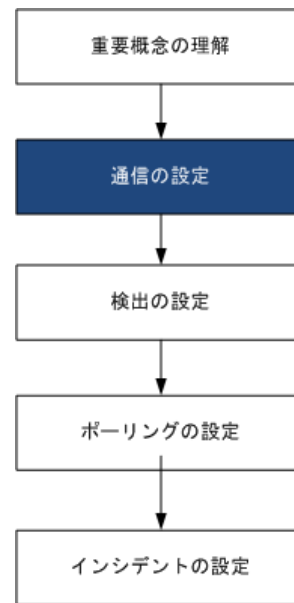
これで NNMi はデフォルト設定のみとなり、本製品を新しいシステムにインストールしたのと同じ状態です。
- 8 NNMi の設定を開始します。以下のいずれかを行います。
  - 「クイックスタート設定ウィザード」を使用します。
  - NNMi コンソールの [ 設定 ] ワークスペースに情報を入力します。
  - `nnmconfigimport.ovpl` コマンドを使用して、手順 3 で保存した NNMi 設定の一部またはすべてをインポートしてください。



`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 (3600 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` リファレンスページまたは UNIX のマンページを参照してください。



# NNMi 通信



HP Network Node Manager i Software (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ヘルスを監視します。各自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、アクセス資格認定、適切なタイムアウト、再試行値すべてで NNMi を設定します。ネットワークのいくつかのエリアでプロトコルを無効にし、トラフィックを削減またはファイアウォールを順守できます。

設定する通信の値は NNMi の検出および状態ポーリングの基礎を形成します。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。このように、ネットワークのいくつかの領域との 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。

この章には、以下のトピックがあります。

- 通信の概念
- 通信の計画作成
- 通信の設定
- 通信の評価
- 通信の調整

## 通信の概念

NNMi は、SNMP と ICMP を主に要求と応答の方式で使います。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。

以下の概念が 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
- 通信設定および `nnmsnmp*.ovpl` コマンド

### 通信の設定レベル

NNMi 通信設定には、以下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、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 は設定を以下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します。
- 2 どの設定もまだ定義されていない場合、NNMi はノードがいずれかの**領域**に属するか判断します。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、該当する特定のノードの空白の値（ある場合）に使用します。追加領域の設定は考慮されません。
- 3 まだ定義されない設定がある場合、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みます。

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。

## 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。たとえば、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かかります。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。設定により、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。たとえば、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、2 回目の要求への応答は 10 秒待ち、3 回目の要求の応答は 15 秒待つてからのポーリングサイクルに移ります。

## 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ーの制御に従う必要があります。

- 認証プロトコルは、メッセージダイジェストアルゴリズム 5 (MD5) またはセキュアハッシュアルゴリズム (SHA) のいずれかを選択した方を使って、ハッシュベースのメッセージ認証コード (HMAC) 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、またはデータ暗号化標準 - 暗号ブロック連鎖 (DES-CBC) 対称暗号化プロトコルを使用します。



DES-CBC は弱い暗号と考えられています。そのため、DES-CBC を使用する場合は、より強い暗号を選択することをお勧めします。暗号の選択を変更するには：

- 1 NNMi コンソールから、[ 設定 ] ワークスペースをクリックします。
- 2 [ インシデント ] フォルダーを展開します。
- 3 [ トラップサーバー ] フォルダーを展開します。
- 4 [ トラップ転送の設定 ] をクリックします。
- 5 [ プライバシプロトコル ] リストで、より強い暗号を選択します。

- ▶ NNMi が管理するノードで **SNMPv3** 通信を設定する場合は、**DES-CBC** を使用しないでください。

NNMi は、(IP アドレスフィルターやホスト名フィルター経由で定義された) 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ています。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。

## 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、各自のネットワーク環境でどのバージョンでも処理できますし、全バージョンの混合したものも処理できます。

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** バージョンにより、NNMi でのノードからのトラップの受け入れが、以下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 **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、以下のとおりです。

- [ **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、プライバシーなし ]**— 認証もプライバシー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ーがないユーザー、次に、認証とプライバシーがあるユーザーを試みます。
- **[ 認証、プライバシーなし ]**— 認証はあるがプライバシー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ーのあるユーザーを試みます。
- **[ 認証、プライバシー ]**— 認証もプライバシー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。

## 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り、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** 検出サイクルの後、以前検出した **SNMP** エージェントが応答しない場合（たとえば、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り **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」とであると報告し、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]** は、NNMi の **SNMP** 使用方法に影響します。詳細については、NNMi ヘルプにある「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。

## 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することができます（たとえば、インフラストラクチャー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）。

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 では以下のような結果が生じます。

- オプションの自動検出ルール ping スイープ機能は、ネットワークの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、シードされるか、または隣接 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、すべてシードしないと失われる可能性があります。
- StatePoller 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。(ただし、デバイスが SNMP に応答すると、StatePoller は ICMP を使用しません。)
- オペレーターはトラブルシューティングの間は、[アクション]>[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 では以下のような結果になります。

- 検出では、存在しないデバイスの情報は収集できません。すべてのデバイスで No SNMP デバイスのプロファイルを受信します。
- 検出では、クエリーによって追加の隣接デバイスを見つけることができません。デバイスはすべて直接にシードされる必要があります。
- 検出では、データベー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 デバイスのプロファイルを持つデバイスについては、StatePoller は ICMP (Ping) のみ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。
- StatePoller は、コンポーネントの稼動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、デバイスに接触して近隣分析を実行し、インシデントの根本分析を見つけることはできません。

## 通信設定および nnmsnmp\*.ovpl コマンド

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 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は、ユーザー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。ユーザーとパスワードを指定しないと、コマンドはエラーになります。

## 通信の計画作成

以下の領域で決定します。

- 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

## デフォルトの通信設定

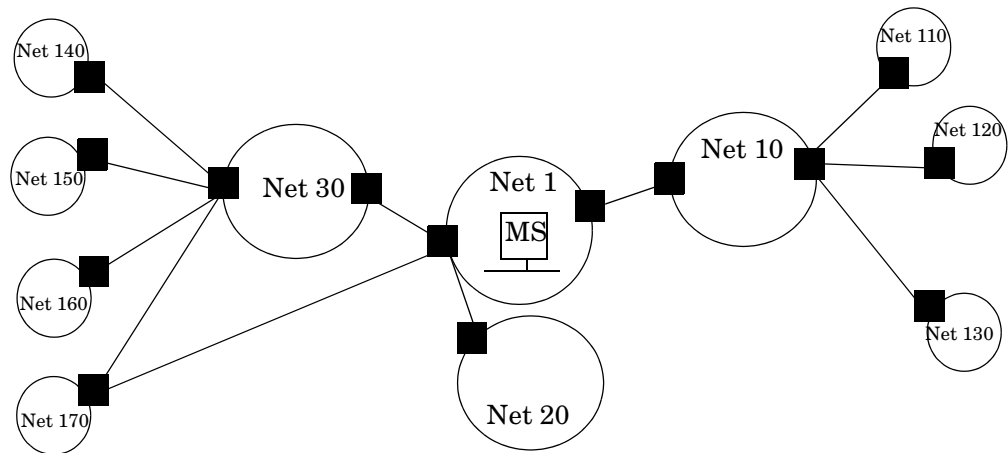
NNMi は、該当する領域や特定のノードで指定しなかった設定をデフォルト値を使用して完成させるため、大半のネットワークで妥当なものになるようデフォルトを設定します。

- 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？

## 通信設定領域

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。たとえば、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、以下のように領域を設定することができます。

- 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 ホップのどちらのパス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どうかに従って、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（または可能なコミュニティ文字列の一部）が使用されていて、命名規約（`rtrnmm.yourdomain.com` など）でルーターを識別できる場合は、全ルーターを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い（かつ、他に一致する領域がない）領域から設定を適用します。

## 特定のノードの設定

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、以下の例があります。

- SNMPv2c/SNMPv3 GetBulk 要求に適切に応答しないノード
- 他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」を参照してください。

## 再試行とタイムアウトの値

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、ビジー状態にあるか、または離れたところにあるデバイスからより多くの応答を集められます。このように応答率が高まると、偽の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注意が必要なことを知るのに時間がかかるようになります。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、このために各自の環境で値のテストと調整の期間が必要になる可能性があります。

各ホップの現在のラグタイムに関するヒントを得るには、以下を実行します。

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

## アクティブなプロトコル

通信の設定とモニタリングの設定を使用して、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することができます。インフラストラクチャー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、モニタリングの設定を使用してプロトコルの使用を微調整します。通信またはモニタリング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。



SNMP 通信を無効にすると、ネットワークの NNMi のステータスと稼動状態の監視機能がかなり危険な状態になります。

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はずであるかに注意してください。

アクセス資格認定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 デバイスのプロファイルに割り当て、**ICMP** のみを使ってデバイスを監視します。

## 複数のコミュニティ文字列または認証プロファイル

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。



有望なコミュニティ文字列を試す間に、**NNMi** クエリーにより、デバイスで資格認定不合格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、資格認定不合格は安全に無視できる可能性があることを業務部に知らせてください。代わりに、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）が可能な限り厳しく設定して、資格認定不合格の数を最小に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。

### SNMPv1 と SNMPv2 のコミュニティ文字列

**SNMPv1** または **v2c** 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。

### 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（ある場合）を決定します。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。

**SNMPv3** 通信の場合、**NNMi** では以下の認証プロトコルがサポートされます。

- HMAC-MD5-96
- HMAC-SHA-1

**SNMPv3** 通信の場合、**NNMi** では以下のプライバシプロトコルがサポートされます。

- DES-CBC
- TripleDES
- AES-128
- AES-192
- AES-256

固有ノードまたは領域設定ごとに、1つの認証プロトコルおよび1つのプライバシープロトコルを指定できますが、指定しないこともできます。

- ▶ TripleDES、AES-192、AES-256 のプライバシープロトコルを使用するには、**Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files** ライブラリが必要です。このライブラリは NNMi インストールプロセスの一部として自動的にインストールされます。ライブラリを誤って削除してしまった場合は、「[設定変更の提案](#)」(455 ページ) の手順に従って復元できます。

## 通信の設定

この項を読んだ後、特定の手順については、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。

- ▶ 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、「[ベストプラクティス: 既存の設定を保存](#)」(34 ページ) を参照してください。

通信の以下のエリアの設定

- 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によって、ノードの設定を入力できます。

- ▶ すべての [**通信の設定**] を [**保存して閉じる**] と、NNMi コンソールに戻り、変更が導入されます。

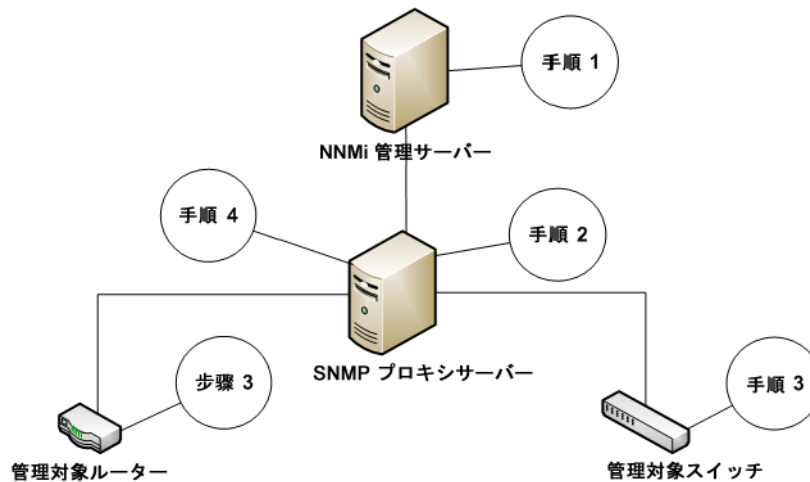
### ベストプラクティス

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。

## SNMP プロキシの設定

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 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します。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 要求を送信します。
- 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します。
- 4 SNMP プロキシサーバーが NNMi 管理サーバーに応答します (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 は以下の OID 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。

## 通信の評価

この項では、通信設定の進行と成功を評価する方法をリストします。これらの作業のほとんどを完了できるのは、検出が完了した後です。

以下について考えます。

- すべてのノードが **SNMP** 用に設定されましたか？
- デバイスについて **SNMP** アクセスは現在利用できますか？
- 管理 **IP** アドレスは正しいですか？
- **NNMi** は正しい通信設定を使っていますか？
- **State Poller** 設定は通信設定と一致していますか？

### すべてのノードが SNMP 用に設定されましたか？

- 1 **[ノード]** インベントリビューを開きます。
- 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します。
  - 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、アクセス資格認定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)。

### デバイスについて SNMP アクセスは現在利用できますか？

- 1 インベントリビューでノードを選択します。
- 2 **[アクション]>[ステータスのポーリング]** または **[アクション]>[設定のポーリング]** を選択します。

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` リファレンスページ、または **UNIX** のマンページを参照してください。

### 管理 IP アドレスは正しいですか？

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、以下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します。
- 2 **[アクション]>[通信の設定]** を選択します。
- 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します。

## NNMi は正しい通信設定を使っていますか？

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ます、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、または以下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します。
- 2 **[アクション]>[通信の設定]** を選択します。
- 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します。

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。

## State Poller 設定は通信設定と一致していますか？

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します。
- 2 **[アクション]>[モニタリングの設定]** を選択します。

監視設定または通信設定のどちらかによってデバイスへのある種類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。

## 通信の調整

### 認証不合格の削減

検出の間に NNMi があまりにも多くの認証トラップを生成している場合は、NNMi が試行するアクセス資格認定の、より小さいグループで小さい領域または特定のノードを設定します。

### タイムアウトと再試行の調整

NNMi が検出中に SNMP を使ってデバイスに接触を試みる時、通信設定は NNMi が必要なデバイス情報を収集できるかどうかを調べます。通信設定に正しい 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 デバイスを検出している場合、NNMi は SNMP タイムアウトと再試行用に設定済みの構成を使います。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分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のみ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。

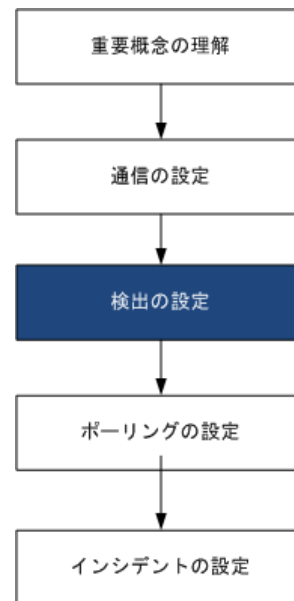
### デフォルトコミュニティ文字列の削減

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。





# 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。HP Network Node Manager i Software (NNMi) 検出により、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、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 6.x/7.x の使用経験があり、NNMi 9.20 で検出がどのように変わったのかを知りたい方は、相違点の概要について NNM 6x/7x からの移行の「ネットワーク検出」を参照してください。

この章には、以下のトピックがあります。

- 検出の概念
- 検出の計画
- 検出の設定
- 検出の評価
- 検出の調整

## 検出の概念

ルーターとスイッチのみを検出する NNMi のデフォルト動作により、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（たとえばパソコンやプリンター）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限り避けるべきでしょう。たとえば、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、いく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、検出構成を非常に単純にしたり、極めて複雑にしたり、その間の適当なレベルにできます。

▶ NNMi はデフォルトの検出を何も実行しません。各種のデバイスが NNMi トポロジに存在する前に、検出を設定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 がそのノードを積極的に管理しているかどうかに関係なく、ライセンスの限度までカウントします。所有している NNMi ライセンスの内容は、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。ネットワークのいくつかの領域ではこのデフォルト設定を変更できるため、職責の範囲を超えたデバイスの検出が可能になります。（StatePoller の詳細については、「NNMi 状態ポーリング」（77 ページ）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伝え、NNMi に各領域の開始アドレスを指定して、NNMi に定義された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、スパイラル検出では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) 領域内に存在する可能性があります。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（これはシード済み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。

▶ 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。

## NNMi はデバイスのプロファイルから属性を導き出す

NNMi はデバイスを検出する際に、SNMP を使用していく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 システムの性能を収集して、データベースのトポロジ部分に格納します。システム性能は、ノードフォームに表示されます。ただし、これらの性能は NNMi の他の部分（つまり、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 のデバイスのプロファイルにより）を使用して、デバイスをノードグループに分類します。ノードビュー表では、「**デバイスカテゴリ**」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、リリース時に入手できた数千のシステムオブジェクト ID のデバイスのプロファイルが付属しています。ご使用の環境内にしかないデバイス用にデバイスのプロファイルのカスタム設定して、これらのデバイスをカテゴリ、ベンダーなどに対応付けることができます。

---

## 検出の計画

以下の領域で決定します。

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

### 基本的な検出方法を選択する

完全なリストベース検出を行うのか、完全なルールベース検出を行うのか、それともこの 2 つの方法を組み合わせるのかを決定します。

## リストに基づいた検出

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) 領域内に存在する可能性があります。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（これはシード済み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。



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して検出することをお勧めします。

リストベース検出のみを使用することの利点を以下に示します。

- 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仕様をサポートします。
- 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。

リストベース検出のみを使用することのデメリットを以下に示します。

- NNMi は、ネットワークに新規ノードが追加されても検出しません。
- 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。

## 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。各々のルールに対して、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weep を有効にすることにより）指定する必要があります。それにより 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利点を以下に示します。

- 大規模なネットワークに適しています。NNMiは大量の数のデバイスを、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)。
- 新規デバイスをタイミングよく管理するためのサービス内容合意書や、許可されていない新規デバイスがあれば注意を与えるためのセキュリティガイドラインを順守するために、新しいデバイスがネットワークに追加されると検出されます。

ルールベース検出を使用することのデメリットを以下に示します。

- 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よりも多くのデバイスをNNMiが検出する場合は、不要なデバイスをNNMiトポロジから削除できます。ノードの削除には時間がかかる可能性があります。
- すべての非シードノードは、検出時にデフォルトのテナントを受信します。NNMiマルチテナント方式を使用する場合は、検出後にテナント割り当てを更新する必要があります。

ルールベース検出  
のみ

## 自動検出ルール

### 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、検出範囲に次のように影響します。

- **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。たとえばある自動検出ルールによりIPアドレスの一氏が除外されると、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限り検出されません。
- **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すべて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のみ適用されます。

### 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、順序番号を小さくして作成します。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、このルールに一致するオブジェクトを検出プロセスはすぐにとばします。

- IPアドレス範囲またはシステムオブジェクト ID 範囲のルールにより無視された設定は、その自動検出ルールのみに影響します。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可能です。



一部のネットワークでは、Hot Standby Router Protocol (HSRP) や Virtual Router Redundancy Protocol (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。HSRP を使用するときのように、ルーターがルーター冗長グループ (RRG) で設定されている場合、RRG 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 RRG の検出および管理をサポートしません。各 RRG には固有の保護された IP アドレスが必要です。

## Ping スweep

ping sweep 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することができます。初期検出では、すべてのルールで ping sweep を有効にするとよいでしょう。そうすることで十分な情報が 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。



ping sweep は、16 ビット以下のサブネット (たとえば 10.10.\*.\*) で機能します。

ping sweep 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で便利です。



ファイアウォールは ping sweep をネットワークに対する攻撃としてみなすことがよくあり、その場合、ファイアウォールは ping sweep 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。

### ベストプラクティス

ping sweep は、小さな検出範囲にのみ有効にしてください。

## SNMP トラップからの検出ヒント

NNMi 9.01 の時点で、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するようになりました。この機能は、WAN 内でデバイスを検出する場合に特に役立ちます。

## 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ためのオプションを以下に示します。

- [設定] ワークスペースの [検出] にある [シード] をクリックして [検出シード] フォームのシードを入力します。
- `nmloadseeds.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 sweep 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。

## 自動検出ルールのベストプラクティス

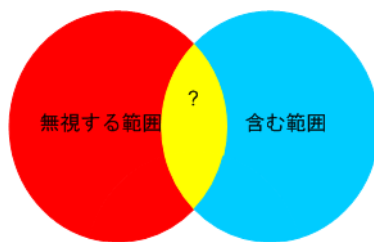
- NNMi はすべての検出対象デバイスを自動的に管理するため、管理したいネットワークの範囲に厳密に一致する IP アドレス範囲を使用してください。
  - 複数の IP アドレス範囲を 1 つの自動検出ルール内で使用して、検出を限定することができます。
  - 自動検出ルールに大きな IP アドレス範囲を追加した後に、そのルール内の検出からいくつかの IP アドレスを除外することができます。
- システムオブジェクト ID 範囲の指定は接頭部分であり、絶対値ではありません。たとえば、範囲 1.3.6.1.4.1.11 は 1.3.6.1.4.1.11.\* と同じです。

## 例

### 検出ルールの重複

図 2 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 検出で検出されて含まれる 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 2 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ー以外のすべての 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 プリンター (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。

## ノード名の解決

デフォルトでは、NNMi はノードを次の順序で識別しようとします。

- 1 短い DNS 名
- 2 短い sysName
- 3 IP アドレス



ノードのホスト名を変更した場合、NNMi データで名前変更が反映されるまでに時間がかかります。これは、パフォーマンスを向上させるために、NNMi が DNS 名をキャッシュするためです。



以下のシナリオでは、ノード名解決のデフォルト順序を変更したほうがよい場合を説明しています。

- 組織が DNS 設定の更新を外部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できます。この場合、**sysName** の選択をノード名解決の最初の選択肢として設定して、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** を設定も維持もしない場合、**sysName** をノード名解決の 3 番目の選択肢として選択します。

### ベストプラクティス

**DNS** 完全名または **DNS** 短縮名を基本的な命名方法として使用している場合、**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** 完全名が命名方法の場合、トポロジマップ上のラベルを長くできます。

### ベストプラクティス

**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。( **NNMi 8.0x** では、最大のループバックアドレスが管理アドレスとして選択されます。 )

## サブネット接続ルール

### リストベース検出のみ

リストベース検出では、**NNMi** 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。

### ルールベース検出のみ

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：

- 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 一致が見つかり、**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。

### ベストプラクティス

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

## 検出シード

検出シードとして使用するデバイスをリストします。

### ベストプラクティス

優先管理 **IP** アドレスを選択する **NNMi** のルールの 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り、**NNMi** に影響を与えることができます。

### ベストプラクティス

**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、デバイス上の他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明するように正しく設定されていることを確認します。



### リストベース検出のみ

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します。このリストを、資産管理ソフトウェアから、または他のツールからエクスポートすることが可能です。

NNMiはこのリストにデバイスを自動的に追加することがないため、責任を負っているデバイスだけがリストに追加含まれるようにするか、監視/ステータス計算に影響を及ぼすデバイスだけがリストに含まれるようにしてください。

### ルールベース検出のみ

ルールベース検出の場合、検出シードはオプションです。

- ping スweepが自動検出ルールに対して有効の場合、そのルールのシードを指定する必要はありません。
- ping スweepが無効な各自動検出ルールで、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リンク全体のARPエントリを維持しないため、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。

### ベストプラクティス

ルールベース検出を最も完璧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。

## 再検出の間隔

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は行われません。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(たとえば、ファームウェアアップグレードまたは接点システム)。
- ネットワークに追加された新しいノード。

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。非常に動的なネットワークでは、最低24時間の間隔を使用するとよいでしょう。これより安定したネットワークでは、その期間を広げることができます。

## オブジェクトを検出しない

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3つの方法があります。

- [通信の設定]フォームで、ICMP通信またはSNMP通信あるいはその両方を、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いずれ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、「ポーリングプロトコル」(47ページ)を参照してください。

- **[ 検出の設定 ]** フォームで、NNMi に特定の 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ように指示する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まで行われません。
- **[ 検出の設定 ]** フォームで、特定の IP アドレス範囲または特定の IP アドレス、あるいはその両方をデータベースから除外するよう NNMi に指示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 アドレスフィルターを設定して、IP アドレス範囲を検出から除外することができます。



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、重複アドレスドメインを異なるテナントに配置します (これはシード済み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。

- **[ 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することができます。詳細については、NNMi ヘルプを参照してください。

## インタフェースの検出範囲

NNMi では、フィルター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のみ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後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、大きいデバイスの検出パフォーマンスを向上させることができます (特にそのようなデバイスのすべてのインタフェースを管理しない場合)。

[ 検出の設定 ] フォームの **[ 含まれるインタフェース範囲 ]** タブで定義する含まれるインタフェース範囲のフィルターでは、システムオブジェクト ID プレフィックス値および 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。

## NNMi による仮想 IP アドレスの監視

NNMi は、仮想 IP アドレスを共有するクラスター化されたサーバーなどのデバイスを検出および監視します。クラスターが新しいアクティブノードにフェイルオーバーすると、NNMi はその仮想 IP アドレスを新しいアクティブノードに関連付けます。フェイルオーバーしてから NNMi が変更を検出するまでにしばらく時間がかかるため、この関連付けはすぐには行われません。

特定の状況に合わせて NNMi を設定するため、いくつかのアクションを実行できます。

NNMi で仮想 IP アドレスを監視する場合は、以下のオプションのいずれか 1 つだけを使用してください。

- オプション 1: このオプションの場合、NNMi は  $N+1$  個の非 SNMP デバイスを管理します。ここで  $N$  は、非仮想 IP アドレスによって検出されたクラスターに属するメンバーの数です。NNMi は、さらにもう 1 つの (+1) 非 SNMP ノードを検出し、仮想 IP アドレスを使用して設定します。

NNMi が仮想 IP アドレスを検出する動作を停止しないでください。この方法を使用することにより NNMi は、仮想 IP アドレスと、この仮想 IP アドレスを使用するように設定されたデバイスのネットワークインタフェースカード (NIC) に関連付けられている物理 IP アドレスを検出します。NNMi は、各デバイスを別々の非 SNMP ノードとして検出および監視します。

- オプション 2: デバイスの物理 IP アドレスをクラスター化されたサーバーの優先される管理アドレスとして使用するよう NNMi を設定します。この方法の詳細については、NNMi のヘルプの「特定ノードの設定フォーム (通信設定)」のトピックを参照してください。



NNMi は、アクティブノードから新しいアクティブノードへの仮想 IP アドレスの移行をすぐには認識しない場合があります。NNMi は、クラスター内の現在のアクティブノードとは別のノードを使用して仮想 IP アドレスのステータスを表示することがあります。

NNMi で仮想 IP アドレスを監視しない場合は、NNMi コンソールを使用して以下の手順を実行します。

- 1 [設定] ワークスペースの [検出の設定] をクリックします。
- 2 [除外対象 IP アドレス] タブをクリックします。
- 3 仮想 IP アドレスまたはアドレス範囲を、検出対象から除外するアドレスの一覧に追加します。
- 4 作業内容を保存します。

## 検出の設定

ここでは、設定のヒントを一覧にし、いくつかの設定例について説明します。このセクションの情報を read した後で、特定の手順の NNMi ヘルプの「検出の設定」を参照してください。



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 (ある場合) を完了する。
- サブネット接続ルールを設定する。
- 名前解決設定を設定する。
- コンソールまでさかのぼって [保存して閉じる] を行う。



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、「[ベストプラクティス: 既存の設定を保存](#)」(34 ページ)を参照してください。

## 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。

## 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` コマンドを使用してそのノードリストを NNMi にインポートします。
- シードファイルで、管理アドレスとして NNMi が選択する IP アドレスに影響を与える手段として IP アドレスを指定します。(ホスト名を使用すると、DNS は IP アドレスを各ノードに提供します。)
- シードファイルのエントリーとして適切な書式を以下に示します。

```
IP_address # node name
```

```
IP_address2, <tenant_UUID_or_tenant_name> # node name
```

以下の書式は、NNMi と人間の両方が容易に理解できます。

- 保守目的のため、使用するシードファイルは1つだけにするをお勧めします。ノードを必要に応じて追加して、`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。



シードファイルをロードできない場合、`nmsproc` (644 パーミッション) でファイルを読み取れるようにします。

- 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。ノードは直接 NNMi コンソールで削除してください。
- 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から、そして NNMi コンソールの [設定] ワークスペースの [検出] 領域にある [シード] フォームから削除してから、そのノードを NNMi コンソールで再入力するか、`nnmloadseeds.ovpl` コマンドを実行します。
- 検出ルールを、そのルールのシードを指定する前に、完全に設定します。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] フォームは、データベースモデルの [検出の設定] フォームに含まれていない個別のフォームです。結果として、[検出シード] フォームについての情報を保存すると、NNMi によってシード設定は直ちに更新されます。)

ルールベース検出  
のみ

## 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記載しています。

### 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、多少の時間がかかります。初期検出の進行状況を測定する方法を以下に示します。

- **[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]** ワークスペースの **[検出]** で、**[シード]** ページを確認します。このページを、すべてのシードに「ノードが作成されました」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」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 (**[全般]** タブにあります) が Discovery Completed 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、ネットワークの様々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 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 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。

### すべてのシードが検出されているか？

- 1 **[設定]** ワークスペースの **[検出]** で、**[シード]** をクリックします。
- 2 **[シード]** ページで、ノードのリストを **[検出シードの結果]** 列でソートします。ノードがエラー状態の場合は、以下について検討してください。
  - ノードに到達できなかったか DNS 名または IP アドレス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 hostnlookup.conf ファイルに加えます。ホスト名に解決されるべきではない IP アドレスが原因で発生する問題に対処するには、該当する IP アドレスを ipnlookup.conf ファイルに含めます。詳細については、hostnlookup.conf および ipnlookup.conf のリファレンスページ、または UNIX マン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、すでに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。検出したノードをいくつか削除するか、ノードパックライセンスを追加購入します。
- ノードが検出されたが SNMP 応答がない — SNMP 通信の問題は、シードされたデバイスだけでなく自動検出によって検出されたデバイスにも発生します。詳細については、「通信の評価」(54 ページ)を参照してください。

## すべてのノードには有効なデバイスのプロファイルがあるか？

- 1 [ノード] インベントリビューを開きます。
- 2 [デバイスのプロファイル] 列を、「デバイスのプロファイルなし」文字列が含まれるようにフィルタリングします。
- 3 ノードが検出されてもデバイスの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]で新規デバイスのプロファイルを追加してから、ノード上で設定ポリシーを実行してそのデータを更新します。

## すべてのノードが正しく検出されたか？

検出の問題を回避するには、管理ドメイン内の他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のみを NNMi で管理するようにします。たとえば、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、そのノードがルーター冗長グループ (RRG) の一部になっている場合には、特別な要件があります。RRG 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固有の IP アドレス (保護されたアドレス以外) を使用する必要があります、そのアドレスで SNMP を有効にする必要があります。NNMi は、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、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「すべてのノードが SNMP 用に設定されましたか？」(54 ページ)の説明にしたがって確認します。

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、以下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 (たとえば 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 スニープを有効にします。

リストベース検出  
のみ

## 自動検出ルール

予想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、以下の設定を確認してください。

- [含まれているノードの検出] を有効にし、自動検出がルールに実行されるようにする必要があります。



- 以下の設定が、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のみが検出されて、SNMP 以外のノードは検出されないことを忘れないでください。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、NNMi が予期した以上のノードを検出してしまう可能性があります。

## 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** ] エントリーに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ある場合、すべてのアドレスヒントが一致とみなされます。(この場合は、「**自動検出ルールを設定する場合のヒント**」(68 ページ)を参照してください。)さらに、ヒントは「**ルールにより無視された**」とマークされたエントリー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、このルールの設定がヒントの処理に使用されます。

- 予想したデバイスのいくつかを検出されない場合、設定した IP 範囲を確認してそのデバイスの IP アドレスが範囲の中に含まれていて小さい順序番号のルールで無視されないようにしてください。
- 必要以上のデバイスが検出されている場合は、含む範囲を変更するか、検出したくないデバイスの IP アドレスの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。

## 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** ] エントリー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ある場合、すべてのオブジェクト ID が一致とみなされます。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ー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、このルールの設定が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、自動検出を拡大して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。
- 各ノードは、検出されて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。

## すべての接続と VLAN は正しいか？

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後の別個のステップとして作成します。NNMi に接続と 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 接続の評価

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** ] > [ **ノードグループマップ** ] をクリックします。)このマップで他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、各々の [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。

## NNMi 検出と重複 MAC アドレス

NNMi は、検出の実行中、ネットワークデバイス間の通信パスを判断するため、ネットワーク内の Ethernet スイッチから転送データベース (FDB) テーブルを読み取ります。NNMi は、これらの FDB テーブルで、検出されたノードに関する情報を検索します。NNMi 管理サーバーは、重複するメディアアクセス制御 (MAC) アドレスへの FDB 参照を検出すると、以下の処理を行います。

- 検出された 2 つ以上のノード (同一テナント内のノード、またはデフォルトテナントのノードとそれ以外のテナントのノード) に同じメディアアクセス制御 (MAC) アドレスに関連付けられたインタフェースが含まれる場合、NNMi は、FDB にあるそれらの重複 MAC アドレスについてレポートされている通信パスを無視します。これにより、それらの重複 MAC アドレスを含むネットワーク領域の NNMi マップで、接続が失われる場合があります。

**NNMi Advanced - グローバルネットワーク管理機能:** 2 つの NNMi 管理サーバーが、同じメディアアクセス制御 (MAC) アドレスに関連付けられている 1 つのインタフェースを含むノードを検出すると、リージョナル NNMi 管理サーバーのマップで認識される接続がグローバル NNMi 管理サーバーのマップでは失われる可能性があります。

- 1 つのノードに同じ MAC アドレスを持つ複数のインタフェースが含まれる場合、NNMi は、それらのインタフェースについてのすべての通信パス情報を収集し、NNMi マップにその情報を表示します。

データベース (FDB) 情報を転送すると、以下の場合に NNMi が誤った L2 接続を確立する可能性があります。

- FDB がキャッシュとして設定されており、使用されていないデータが含まれている。
- それぞれ異なる (場合によっては競合する) FDB データを生成するさまざまなベンダーのハードウェアがネットワーク環境に含まれている。

オプション: NNMi 管理者は、特定のノードグループでこの FDB データを無視するように検出を設定できます。

## 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します。
- 2 デバイスを削除します。

そのデバイスがシードの場合、シードを削除し、それからシードを再度追加します。

---

## 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のみが検出されるようにしてください。

- IP アドレス範囲またはシステムオブジェクト ID、あるい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 1 つ以上のノードを削除するには、`nnmnodedelete.ovpl` コマンドを使用します。このコマンドにより、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 1 つ以上の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または VLAN のインデックス付けを無効にすることによって修復できる、特別な検出状況もあります。詳細については、「[特定ノードの検出プロトコルの使用を抑える](#)」(419 ページ) または「[大規模スイッチの VLAN インデックス付けの使用を抑制する](#)」(421 ページ) を参照してください。

## 検出ログファイル

`nnm?.0.log` ファイル内で、文字列 `com.hp.ov.nms.disco` で始まるクラスの **Exception** というキーワードを含むメッセージを探します。ログファイルの詳細については、「[NNMi ロギング](#)」(425 ページ) を参照してください。

## 無番号インタフェース

NNMi 9.10 パッチ 2 より前の NNMi は、xDP を有効にしない限り、無番号インタフェースのレイヤー 2 接続を検出しませんでした。NNMi 9.20 パッチ 2 では、無番号インタフェース検出と監視ソリューションを提供しており、デフォルトの MIB-II `ipRoutingTable` と `ipCidrRoutingTable` を使用するデバイスをサポートします。

NNMi 9.20 は、このセクションで説明するソリューションを使用して、IPv4 無番号インタフェースとそれに関連付けられたレイヤー 2 接続を検出および監視します。

このセクションで説明するソリューションは、グローバルネットワーク管理設定で以下のように機能します。

- リモート NNMi 管理サーバーでは通常どおり動作します。
- グローバル NNMi 管理サーバーの場合は、そのサーバーで管理されるノードの場合にのみ動作します。
- リモート NNMi 管理サーバーによって管理されるノードのグローバル NNMi 管理サーバーでは動作しません。

## 無番号インタフェース機能の有効化

- 1 無番号インタフェースを含むデバイスを含むノードグループを作成します。デバイス ID を含む 1 つのノードグループを作成するか、デバイス ID を含む複数の子ノードグループを代表する親ノードグループを作成します。
- 2 以下のファイルを作成します。

Windows: `%NNM_DATA%\shared\%nnm%\conf\%disco%\UnnumberedNodeGroup.conf`

UNIX: `$(NNM_DATA)/shared/nnm/conf/disco/UnnumberedNodeGroup.conf`

- 3 単一のノードグループ名をこのファイルに追加します。繰り返しますが、このファイルには、デバイス ID を含む 1 つのノードグループの名前を含める必要があります。あるいは、デバイス ID を含む複数の子ノードグループを代表する親ノードグループの名前にすることもできます。

```
# This is the name of an node group containing devices with
unnumbered interfaces.
Unnumbered Node Group
```

上の例では、**Unnumbered Node Group** という名前のノードグループが NNMi 内に存在しています。# 文字を先頭に付けて、コメント情報を別の行で追加します。

- 4 オプションステップ: 以下のファイルを作成します。

```
Windows: %NNM_DATA%\shared\%nm%\conf\disco\UnnumberedSubnets.conf
UNIX: $NNM_DATA/shared/nm/conf/disco/UnnumberedSubnets.conf
```

- 5 オプションステップ: このファイルに情報を追加して、NNMi で検出する必要がある特定のルーティングアドレス範囲を示します。複数行の IPv4 CIDR サブネットエントリをランダムな順序でこのファイルに追加できます。



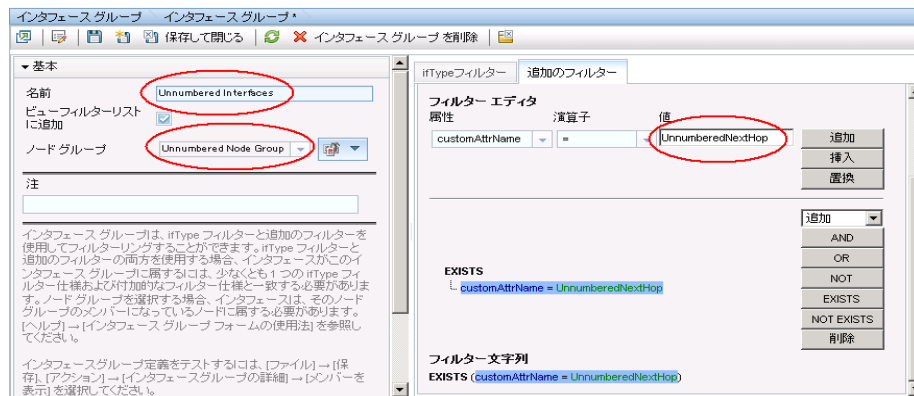
このファイルを作成および設定しない場合、NNMi は、MIB-II ルーティングテーブル全部が、設定されたノードグループのノードに向かって進むようにします。NNMi は、UnnumberedSubnets.conf ファイルを使用して、指定されたサブネット宛先の範囲に収まるルートのみから MIB データを要求します。このファイルを使用し、デバイスでの検出トラフィックの量とパフォーマンスへの影響を軽減することは、優れた手法の 1 つです。

以下に、UnnumberedSubnets.conf ファイルに追加するエントリーの例をいくつか示します。

```
10.1.5.0/18 #This entry filters the following routes: 10.1.0-63.
15.2.126.0/16 #This entry filters the following routes: 15.2.*.*
192.168.1.0/24 #This entry filters the following routes:
192.168.1.0-255
```

- 6 NNMi 管理サーバーを再起動します。
- 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。
- 7 NNMi が次の検出サイクルを完了するまで待ちます。

- 8 無番号インタフェースをすべて検出するには、UnnumberedNextHop という名前のカスタム属性を持つインタフェースを含める新しいインタフェースグループを設定します。



- 9 このソリューションによって作成されたレイヤー2接続を表示するには、[レイヤー2の接続]ビューに移動し、ROUTESからソースを探します。

ステータス	名前	トポロジソース
✓	ntc-g350[NO NAME],ntc-g430[NO NAME]	FDB
✓	ROADM[pdcc0],Site3[pdcc1]	ROUTES

## 無番号インタフェース機能の無効化

無番号インタフェース機能を無効にする場合は、以下の手順を実行します。

- 以下のファイルを削除します。
  - Windows: %NNM\_DATA%\shared\nnm\conf\disco\UnnumberedNodeGroup.conf
  - UNIX: \$NNM\_DATA/shared/nnm/conf/disco/UnnumberedNodeGroup.conf
- 以下のファイルが存在する場合は削除します。
  - Windows: %NNM\_DATA%\shared\nnm\conf\disco\UnnumberedSubnets.conf
  - UNIX: \$NNM\_DATA/shared/nnm/conf/disco/UnnumberedSubnets.conf
- NNMi 管理サーバーを再起動します。
  - NNMi 管理サーバーで **ovstop** コマンドを実行します。
  - NNMi 管理サーバーで **ovstart** コマンドを実行します。
- NNMi が次の検出サイクルを完了するまで待ちます。

詳細については、UnnumberedNodeGroup.conf および UnnumberedNodeGroup.conf のリファレンスページ、または UNIX マニュアルページを参照してください。

## 非応答オブジェクトの削除の制御

オブジェクトが応答しなくなっからの待機日数を指定して、以下の非応答オブジェクトの削除を制御できます。

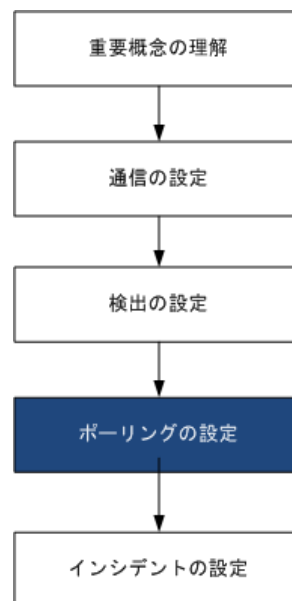
- 非応答ノード
- 停止している接続

非応答オブジェクトの削除を制御するには、以下の手順を実行します。

- 1 **[設定]** ワークスペースで、**[検出の設定]** をクリックします。
- 2 **[非応答オブジェクト制御の削除]** 領域で、該当のオブジェクトを削除するまでにシステムが待機する日数を入力します。ゼロ (0) の値は、ノードが削除されないことを示します。

指定した待機期間が経過すると、非応答オブジェクトがデータベースから削除されます。

# NNMi 状態ポーリング



この章では、HP Network Node Manager i Software (NNMi) StatePoller 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のに役立つ情報を示します。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。

NNM 6.x/7.x で作業した経験があり、NNMi 9.20 で監視がどのように変更されたかを知りたい場合は、『NNM 6x/7x からの移行』の「ステータス監視」に記載されている相違点の概要を参照してください。

この章には、以下のトピックがあります。

- 状態ポーリングの概念
- 状態ポーリングの計画を作成
- 状態ポーリングの設定
- 状態ポーリングの評価
- 状態ポーリングの調整

## 状態ポーリングの概念

この項では、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。この項を読んだ後、さらに詳細な情報については「[状態ポーリングの計画を作成](#)」(78 ページ)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であるか、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 では、トポロジデータベースでのみデバイスをポーリングできます。NNMi が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の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することができます。

State Poller のデータ収集が ICMP (ping) 応答を基礎にするように、または SNMP データを基礎にするように設定できます。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、設定を大幅に簡単にします。

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を注意深く考える必要があります。グループという概念が初めての場合は、その概要について「[ノードグループおよびインタフェースグループ](#)」(35 ページ)と「[ノード/インタフェース/アドレス階層](#)」(40 ページ)を参照してください。

### 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、StatePoller は、明確に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：

- 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します。グループは最も小さい順序番号から最も大きい順序番号へという順序で評価されます。最初に一致するグループが使われ、その時点で評価は停止します。
- 2 オブジェクトを把握したインタフェースグループがない場合、グループは最も小さい順序番号から最も大きい順序番号への順序で評価されます。最初に一致するグループが使われ、その時点で評価は停止します。含まれている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の基準を満たしていないものは、ホストであ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ます。

## 状態ポーリングの計画を作成

この項では、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。

### 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、その他の基準に基づいて、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？以下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

- 1 つ以上の **NNM Performance iSPI** に対応するライセンスが 1 つある場合は、**SNMP パフォーマンスデータ**
- 追加の **SNMP** コンポーネント稼働状態データ

## ポーリング設定の例

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

### 1 NNMi で何を監視できますか？

監視できるのは検出されたもののみであるため、自動検出ルールを設定して、**NNMi** のデータベースに自分の **ProximiT** プロキシサーバーがあることを確認します。検出の設定の詳細は、「**NNMi 検出**」(57 ページ)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 **ProximiT** プロキシサーバーを 1 つのグループにまとめ、同じ監視設定を適用することは理にかなっていません。デバイスのインタフェース (**SNMP**) 監視を行っているのですから、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、ビューをフィルターし、プロキシサーバーのステータスをグループとしてチェックし、グループをサービス停止中にしてファームウェアを更新することもできます。

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

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

### 4 どのデータを収集する必要があるでしょうか？

監視設定が他のグループと異なるのは次の点です。**ProximiT proxy** サーバーの例として、**ICMP** 障害の監視を有効にし、**SNMP** 障害および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、以下のトピックを参照してください。

- 「**NNMi で何を監視できますか?**」(79 ページ)
- 「**グループの計画作成**」(81 ページ)
- 「**ポーリング間隔の計画作成**」(83 ページ)
- 「**どのデータを収集するか**の決定」(84 ページ)

## NNMi で何を監視できますか？

デフォルトで、**NNMi State Poller** は **SNMP** ポーリングを使って以下を監視します。

- **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。



ほとんどの場合、インタフェースに接続されたポーリングによってのみ、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。

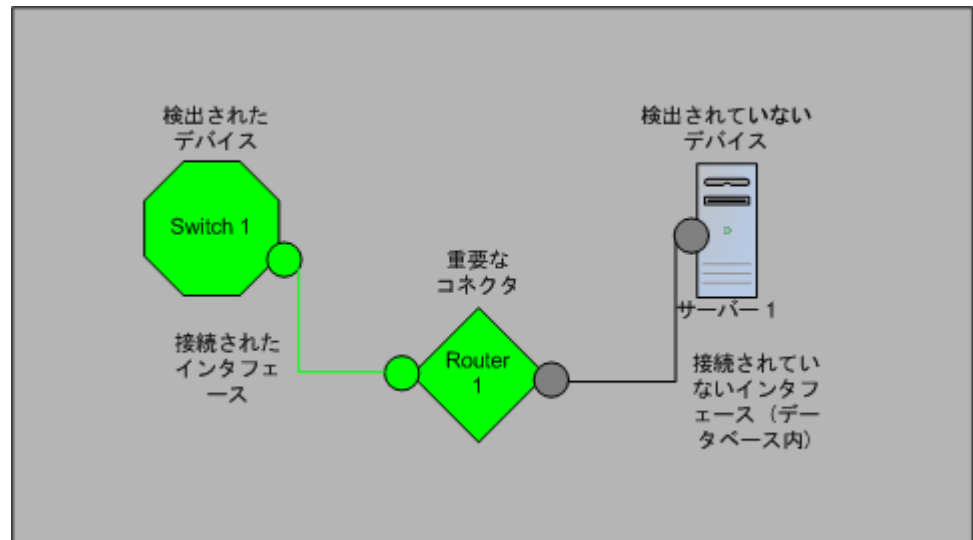
## 監視の拡張

監視を拡張して、以下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のみであり、**ルーターノードグループ**に含まれます。



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。

## 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。たとえば、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 はそのサーバーに面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、NNMi は、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、管理されていないノード。



検出された各ノードは、NNMi が積極的にそのノードを管理しているかどうかに関係なく、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備えたネットワークデバイスを含む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、多数のインタフェースのあるデバイスに大量のトラフィックを追加できる、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します。

## 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管理対象外またはサービス停止中に設定できます。[ 管理対象外 ] は恒久的な状況と見なされます。オブジェクトのステータスを知る心配をする必要はありません。[ サービス停止中 ] は一時的な状況と見なされます。1 つ以上のオブジェクトがオフラインになり、停止中のインシデントが過剰になります。

すべてのグループ設定全体のオーバーレイとして、管理モードを考えてください。グループ、ポーリング間隔、種類に無関係に、オブジェクトのステータスが管理対象外またはサービス停止中に設定されている場合、**State Poller** はそのオブジェクトと通信しません。

### ベストプラクティス

検出を行い、データベースに配置することを選択したデバイスやインタフェース（またはその両方）の中には、ポーリングの必要がないものもあります。管理対象外に恒久的に設定するオブジェクトに注意してください。1 つ以上のノードグループを作成し、管理モードを簡単に設定することもできます。

## グループの計画作成

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する必要があります。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のチェックをあまり頻繁でないようにできます（そもそもチェックを行う場合です）。

### ベストプラクティス

ネットワーク監視を行う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。マップにより、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。

これらグループは、[ 設定 ] > [ ノードグループ ] または [ 設定 ] > [ 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ーするのに使うのと同じグループです。監視設定用にノードフィルターまたはインタフェースフィルターの別個のセットを作成するには、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 ] をクリックします。

[ モニタリングの設定 ] フォームの [ ノードの設定 ] タブと [ インタフェースの設定 ] タブにあるノードグループまたはインタフェースグループのレベルで、ポーリングの種類とポーリングの間隔を設定します。

類似のポーリングのニーズごとに、インタフェースやデバイス（またはその両方）をグループにまとめる基準を決定します。計画作成に際して考慮すべきいくつかの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るか？
- デバイスの種類ごとに収集したポーリング間隔またはデータを差別化しますか？ インタフェースの種類ごとにか？
- NNMi には使用できる事前設定されたグループがあるか？

## ベストプラクティス

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、場所ごとであれ、他の何らかの基準ごとであれ、作成することができます。たとえば、IOS アップグレードを適用しながら、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。

## 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(「状態ポーリングの概念」(77 ページ)を参照)。インタフェースグループはノードグループメンバーシップを参照できるので、計画を実現するインタフェースグループの前に、ノードグループの設定を完了できます。

## 事前設定されたインタフェースグループ

NNMi には、使用できるようにすでに設定済みの便利なインタフェースグループがいくつかあります。たとえば、次のとおりです。

- ISDN 接続に関連付けられた **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
- リンク集合プロトコルに関与するインタフェース

HP は、時間をかけてさらに多くのデフォルトのグループを追加し、設定作業を簡単にしていきます。既存のグループを使用または変更するか、または自分専用のグループを作成できます。

インタフェースには次の 2 つの種類の修飾子があります。つまり、ホストであるノードと **IFType** のノードグループメンバーシップ、またはインタフェース用の他の属性です。これらは次のように組み合わせできます。

- ノードグループ内のノードの全インタフェースを **IFType** と無関係にグループにまとめます。**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、またはその他の **IFType** 属性など)は選択しません。
- 特定の **IFType** または属性のセットのインタフェースは、それらインタフェースが存在するノードに無関係にすべてグループにまとめられます。
- 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のみがグループにまとめられます。

## 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全ノードグループがフィルタービューに意味があるとは限らないので、ノードグループは独立に設定できます。

## 事前設定されたノードグループ

HP は、ノードグループのデフォルト集合を用意して、設定作業を簡単にしています。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以下が含まれます。

- ルーター
- ネットワーキングインフラストラクチャーデバイス (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を持っていないデバイス
- 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。

HP は、時間をかけてさらに多くのデフォルトのグループを追加し、設定作業を簡単にしていきます。既存のグループを使用または変更するか、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
- ホスト名ワイルドカード規約
- デバイスのプロファイルル派生物。たとえば、カテゴリ、ベンダー、ファミリー
- MIB II sysName、sysContact、sysLocation

## ベストプラクティス

簡単に再使用可能な極小のグループを作成し、監視または視覚化のためにこれらを結合して階層クラスターにすることができます。グループ定義は重なることがあります。たとえば、「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」です。ノードは複数のグループに属することができると考えられます。

バランスを取るためには、使われない余分なエントリーのリストで負担を大きくしないように、設定および表示用に豊富なグループのセットを作成します。

## デバイスの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のプロファイルのリストにインデックスを作成します。デバイスプロパティは、ベンダー、製品、ファミリー、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。たとえば、ベンダーを問わず、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「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、検出されたデバイスはすべて、ノードグループについての設定を受け取ります。

## ポーリング間隔の計画作成

オブジェクトグループごとに、NNMi がデータを収集するのに使うポーリング間隔を選択します。サービスレベル契約条項に最も適切に一致するように、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。

## ベストプラクティス

間隔が短いと、可能な限り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。各自の環境について、リソース利用と間隔の間で最良のバランスを見つけてください。



**Causal Engine** は 24 時間ごとに各ノードのステータスのポーリングを実行し、必要に応じてステータス、結果、およびインシデント情報を更新します。ステータスのポーリングは、デバイスに設定されたポーリング間隔のタイミングには影響しません。

## どのデータを収集するか の 決定

**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) を使用して実行できます。

### ICMP (ping)

**ICMP** アドレス 監視は、**ping** 要求を使って、管理対象の各 IP アドレスの使用可能性を確認します。

### 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 **SNMP** 情報を収集するよう、高度に最適化されています。設定の変更を保存すると、**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、収集する設定済み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**、およびベンダー特有の **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。**NNM iSPI Performance for Metrics** をインストールしてある場合は、パフォーマンス測定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、ポーリングを実行する時刻への影響はありません。しかし、各オブジェクトについて保存された追加データによって、**State Poller** 用にメモリー要求が増加する可能性があります。



パフォーマンス 監視設定は **NNM iSPI Performance for Metrics** でのみ使用されます。パフォーマンスに関するコンポーネント稼働監視は、デバイスのパフォーマンスポーリング間隔設定に従います。

## ベストプラクティス

監視設定変更をバッチ処理すると、**State Poller** の進行中の操作が混乱することは少なくなります。

## 状態ポーリングの設定

この項では、設定のヒントを示し、設定例をいくつか挙げます。この項を読んだ後、特定の手順については、NNMi ヘルプの「モニタリング動作の設定」を参照してください。

- ▶ 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、「ベストプラクティス: 既存の設定を保存」(34 ページ)を参照してください。

### インタフェースグループとノードグループの設定

**[設定]** ワークスペースでインタフェースグループとノードグループを作成します。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を作成する」を参照してください。

例 たとえば、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]** > **[ノードグループ]** を開き、**[新規作成]** をクリックします。
- 2 グループ **Proxy Servers** という名前を挙げ、**[ビューフィルターリストに追加]** をオンにします。
- 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 **=** のままにします。
- 4 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** (デバイスのプロファイル) 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 **[デバイスカテゴリ]** セレクターにアクセスし、作成した **Proxy Server** カテゴリをグループの基礎にすることができます。

- 5 グループ定義で **[保存して閉じる]** をクリックします。

- ▶ ノードグループを設定してから、インタフェースグループ設定でノードグループを参照する必要があります。

### インタフェースのモニタリングの設定

**State Poller** は、ノードグループの前に、インタフェースグループメンバーシップを分析します。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ごとに、**[モニタリングの設定]** ダイアログと **[インタフェースの設定]** 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指示のカスタムセットを作成します。指示には以下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
- **NNM iSPI Performance for Metrics** がある場合、パフォーマンスポーリングの有効化または無効化
- **NNM iSPI Performance for Metrics** がある場合、パフォーマンスポーリング間隔の設定

- NNM iSPI Performance for Metricsがある場合、パフォーマンス管理しきい値の設定
- 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大きい順序番号へとリストを評価することを覚えておいてください。

### ベストプラクティス

複数のグループにあてはまるオブジェクトは最も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。

## ノードのモニタリングの設定

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。最も小さい順序番号から最も高い順序番号へと、設定は最初の合致するノードグループに適用されます。

ノードグループごとに、[ **モニタリングの設定** ] フォームを開いてから [ **ノードの設定** ] タブを開きます。State Poller がグループを処理する方法に関する指示のカスタムセットを作成します。指示には以下のものを入れられます。

- 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
- NNM iSPI Performance for Metrics がある場合、パフォーマンスポーリングの有効化または無効化
- NNM iSPI Performance for Metrics がある場合、パフォーマンスポーリング間隔の設定
- NNM iSPI Performance for Metrics がある場合、パフォーマンス管理しきい値の設定
- 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。

### ベストプラクティス

複数のグループにあてはまるオブジェクトは最も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。

## デフォルト設定の確認

State Poller は、定義済みのインタフェース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査し、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。たとえば、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ありません。



変更を実現するためには、コンソールに戻り、すべての [ **設定の監視** ] ダイアログボックスを必ず [ **保存して閉じる** ] ようにしてください。



## 状態ポーリングの評価

この項では、監視設定の進行と成功を評価する方法をリストします。

### ネットワークモニタリングの設定を確認します。

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決定すると、ステータスのポーリングをいつでも開始できます。

### 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、他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念頭に置いてください。

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、グループ名ごとにアルファベット順であって、どの設定が適用されるかを決定する順序番号を反映していません。

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のデバイスのプロファイルを取得します。
- 2 **[設定]** > **[デバイスのプロファイル]** 下にあるデバイスのプロファイルに関する属性マップを確認します。
- 3 ノードグループ定義の属性要件を確認します。

不一致がある場合は、**[デバイスのプロファイル]** に由来するカテゴリを調整して、その種類のデバイスが自分のノードグループに当てはまるようにできます。**[アクション]** > **[設定のポーリング]** を実行して、ノードが当てはまるようにノードの属性を更新する必要があります。

### 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]** > **[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。

**[有効化された障害ポーリング]** と **[障害ポーリング間隔]** の値を調査します。これらの値が予想どおりでない場合は、**[ノードグループ]** または **[インタフェースグループ]** の値を見て、どの順序付けられたグループ一致が適用されるか調べます。

オブジェクト用にトラフィック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]** > **[通信の設定]** をチェックする必要があります。

## 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の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]>[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は出力に表示されます。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。

## ステータスの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のポーリングのパフォーマンスを評価するには、State Poller 稼働状態チェックの情報を使って、State Poller サービスの動作を数値で表し、評価します。

### State Poller は最新の状態に付いていていますか？

表2に説明されているように、[システム情報] ウィンドウの [StatePoller] タブでStatePoller サービスの現在の稼働状態統計をいつでもチェックできます。

表2 StatePoller 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ー	<ul style="list-style-type: none"> <li>最後の 1 分に要求された収集</li> <li>最後の 1 分に完了された収集</li> <li>進行中の収集</li> </ul>
最後の 1 分にスキップを実行する時刻	<p>設定済みのポーリング間隔内で完了しなかった、定期的なスケジュールされたポーリングの数。値がゼロでない場合は、ポーリングエンジンが最新の状態に付いていないか、またはターゲットが応答より速くポーリングされています。</p> <ul style="list-style-type: none"> <li>監視すべきもの：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。</li> <li>実行すべきアクション：nmm?.0.log ファイルで文字列 com.hp.ov.nms.statepoller で始まるクラス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。 <ul style="list-style-type: none"> <li>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、タイムアウトを増やします。</li> <li>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 の使用可能メモリー) を確認します。</li> </ul> </li> </ul>
過去 1 分以内の古い収集	<p>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。</p> <ul style="list-style-type: none"> <li>監視すべきもの：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。</li> <li>実行すべきアクション：nmm?.0.log ファイルで文字列 com.hp.ov.nms.statepoller で始まるクラス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。 <ul style="list-style-type: none"> <li>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外します。</li> <li>古い収集のターゲットが異なる場合、NNMi システムと NNMi 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。</li> </ul> </li> </ul>



表2 StatePoller 稼働状態情報 ( 続き )

情報	説明
ポーリング結果のキューの長さ	<ul style="list-style-type: none"> <li>監視すべきもの: この値はほとんどの時間 0 に近いはずです。</li> <li>実行すべきアクション: キューのサイズがきわめて大きい場合、ovjboss はメモリーを超えて実行されている可能性があります。</li> </ul>
状態マッパー入力キューの長さ	<ul style="list-style-type: none"> <li>監視すべきもの: この値はほとんどの時間 0 に近いはずです。</li> <li>実行すべきアクション: 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。</li> </ul>
状態アップデーターキュー期間	<ul style="list-style-type: none"> <li>監視すべきもの: この値はほとんどの時間 0 に近いはずです。</li> <li>実行すべきアクション: 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。</li> </ul>

## 状態ポーリングの調整

状態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促進されます。ステータスの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、次の設定を考慮してください。

- 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のみをポーリングしていることを確認します。**[ノードの設定]** フォームまたは **[インタフェースの設定]** フォーム (**[モニタリングの設定]** フォームでグローバルにはなく) でこれらのポーリングを有効にし、最も特定の制御を維持し、ポーリングするインタフェースの最も小さいサブセットを選択します。
- 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のみを監視するには、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、ステータスのポーリングは、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のあるステータスのポーリングは多くのネットワーク負荷をかけませんが、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、ステータスの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、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、これらの設定変更でできるのはそれだけです。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、メモリー) 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、StatePoller が使用するメモリーの量が減少する可能性があります。

# NNMi インシ デント



HP Network Node Manager i Software (NNMi) には、NNMi コンソールに作業可能インシデント数を提供する受信 SNMP トラップをフィルタリングする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。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。

NNM 6.x/7.x で作業した経験があり、NNMi 9.20 で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、『NNM 6x/7x からの移行』に記載されている相違点の概要を参照してください。

この章には、以下のトピックがあります。

- インシデントの概念
- インシデントの計画
- インシデントの設定
- 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

## インシデントの概念

NNMi では、以下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動状態が分析され、継続的に各デバイスの稼動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。
- 1つ以上の NNM 6.x/7.x 管理ステーションから転送される NNM 6.x/7.x イベント。
- HP ArcSight Logger 統合からの syslog メッセージ。

NNMi は、この情報をネットワーク管理に有用な情報を提供するこの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ー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関連処理が用意されています。デフォルトのインシデント関連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関連処理を作成することができます。

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、NNM 6.x/7.x イベント、または syslog メッセージと一致しない場合、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 [管理対象外] または [サービス対象外] に設定されている場合、またはデバイスの障害ポーリングが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、以下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」
- <http://support.openview.hp.com/selfsolve/manuals> から入手できる『HP Network Node Manager i-series Software 因果関係分析ホワイトペーパー』

## インシデントライフサイクル

表 3 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 3 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
抑止済み	インシデントは保管場所にあり、別のインシデントとの関連処理待ちです。インシデントビューアー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、「インシデントの抑制、強化、およびダンプニング」(99 ページ)を参照してください。	NNMi	• NNMi

表3 NNMi インシデントライフサイクル(続き)

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
登録済み	インシデントは、インシデントビューで見ることができます。 インシデントは任意の設定済み宛先へ転送されます(近隣またはグローバルマネージャー)。	NNMi ユーザーはインシデントビューでこの状態を設定することもできます。	<ul style="list-style-type: none"> <li>• ユーザー</li> <li>• ライフサイクル移行アクション</li> <li>• インシデントを転送する統合</li> </ul>
進行中	インシデントは問題を調査するいずれかの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	<ul style="list-style-type: none"> <li>• ユーザー</li> <li>• ライフサイクル移行アクション</li> <li>• インシデントを転送する統合</li> </ul>
完了	インシデントによって指定された問題の統合は完了し、ソリューションが配置されています。 インシデントが識別する問題 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	<ul style="list-style-type: none"> <li>• ユーザー</li> <li>• ライフサイクル移行アクション</li> <li>• インシデントを転送する統合</li> </ul>
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。たとえば、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	<ul style="list-style-type: none"> <li>• ユーザー</li> <li>• ライフサイクル移行アクション</li> <li>• インシデントを転送する統合</li> </ul>

## トラップおよびインシデント転送

表 4 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。テーブルの補足テキストによって、NNMi の SNMP トラップ転送メカニズムと NNMi のノースバウンドインタフェース SNMP トラップ転送メカニズムが比較できます。

表 4 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

	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	<ul style="list-style-type: none"> <li>ネットワークデバイスからの SNMP トラップ</li> <li>NNM 管理ステーションからの NNM 6.x/7.x イベント</li> <li>HP ArcSight Logger からの syslog メッセージ</li> </ul>	<ul style="list-style-type: none"> <li>ネットワークデバイスからの SNMP トラップ</li> <li>NNMi 管理イベント</li> <li>HP ArcSight Logger からの syslog メッセージ</li> </ul>	<ul style="list-style-type: none"> <li>ネットワークデバイスからの SNMP トラップ</li> <li>NNM 管理ステーションからの NNM 6.x/7.x イベント</li> <li>HP ArcSight Logger からの syslog メッセージ</li> </ul>
転送フォーマット	受信したままの SNMPv1、v2c、または v3 トラップ (SNMPv3 トラップは SNMPv2c トラップへ変換可能)	NNMi インシデントから作成された SNMPv2c トラップ	NNMi インシデント
追加情報	ほとんどの場合、NNMi は varbind を追加して元のソースオブジェクトを識別します。NNMi が SNMPv1 トラップを変更することはありません。	NNMi は varbind を追加して元のソースオブジェクトを識別します。	リージョナルマネージャープロセスによってインシデントに追加された情報はすべて、転送済みインシデントに保持されます。
設定先	[設定] ワークスペースの [トラップ転送の設定]	[統合モジュールの設定] ワークスペースの [HPOM]、[Northbound インタフェース]、または [Netcool]	[SNMP トラップの設定] フォーム、[リモート NNM 6.x/7.x のイベント設定] フォーム、または syslog 設定の [グローバルマネージャーへの転送] タブ

表4 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(続き)

	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
注		NNMi には、NNMi Northbound インタフェース上にいくつかの統合が構築されています。『NNMi 統合リファレンス』の「Netcool ソフトウェア用 NNMi 統合モジュール」および「HP NNMi-HPOM 統合」の章を参照してください。	グローバルマネージャーのインシデントビューに表示されるリモートインシデントを転送します。転送済みインシデントはグローバルマネージャー上での相関処理に参加します。
詳細情報	NNMi ヘルプにトラップ転送を設定する	『NNMi 統合リファレンス』の「NNMi Northbound インタフェース」の章を参照してください。	<ul style="list-style-type: none"> <li>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設定への転送設定</li> <li>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設定への転送設定</li> </ul>

## 比較: サードパーティ SNMP トラップを別のアプリケーションに転送する

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、以下のいずれかの方法を使用します。

- NNMi SNMP トラップ転送メカニズムを使用。NNMi SNMP トラップ転送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 ノースバウンドインタフェース SNMP トラップ転送メカニズムを使用。受信した SNMP トラップを転送する NNMi Northbound インタフェースの設定の詳細については、『NNMi 統合リファレンス』の「NNMi Northbound インタフェース」の章を参照してください。

受信側アプリケーションがトラップを識別する方法は、SNMP トラップ転送メカニズムでは以下のように異なります。

- Windows (すべて) および UNIX (元のトラップ転送なし)

この説明は、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り、トラップ転送先へ転送する前に各 SNMP トラップが収集されます。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。)

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付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。 **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があります。

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、RFC 2851、および以下のファイルの「NNMi が提供するトラップ **varbind**」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搭載された 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り、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、受信側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する必要があります。

詳細については、NNMi ヘルプの「トラップ転送先フォーム」の元のトラップ転送オプションを参照してください。

- NNMi ノースバウンドインタフェース (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付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。 **IncidentNodeHostname (1.3.6.1.4.1.11.2.17.19.2.2.21)** および **IncidentNodeMgmtAddr (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。

## MIB

NNMi では、以下の管理情報ベース (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。

- Custom Poller 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 が稼動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)
- (NNM iSPI Performance for Metrics) しきい値監視で使用するすべての MIB 変数

NNMi では、以下の管理情報ベース (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。



- ノースバウンド宛先に転送するすべての **SNMP** トラップ
- (NNM iSPI NET) トラップ分析レポートからアクセスするすべての **MIB** 変数



NNMi には、現在サポートされていない **MIB** がリストされた README.txt ファイルがあります。README.txt ファイルは以下のディレクトリに保存されています。

- **Windows:** %NnmInstallDir%\misc\%nnm%\snmp-mibs
- **UNIX:** \$NnmInstallDir/misc/nnm/snmp-mibs

## 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、重複解除、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。

## 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、表 5 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 [ 関連処理の注 ] 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 5 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
<code>cia.reasonClosed</code>	<p><b>NNMi</b> がインシデントをキャンセルしたか解決済みにした理由。この理由は、<b>NodeUp</b> や <b>InterfaceUp</b> など、結果の名前にもなります。</p> <p>このフィールドが設定されていない場合は、<b>NNMi</b> 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。</p> <p><code>cia.reasonClosed</code> CIA の <b>NNMi</b> の期待値を判断するには、<b>NNMi</b> ヘルプの「<b>NNMi</b> によるインシデントの解決方法」を参照してください。</p>
<code>cia.incidentDurationMs</code>	<p>機能停止の時間 (ミリ秒単位)。ステータスが停止中になってから動作中に戻るまで、<b>NNMi</b> が測定します。この値は、<code>cia.timeIncidentDetectedMs</code> と <code>cia.timeIncidentResolvedMs</code> 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。</p>
<code>cia.timeIncidentDetectedMs</code>	<p><b>NNMi Causal Engine</b> が最初に問題を検出したときのタイムスタンプ (ミリ秒単位)。</p>
<code>cia.timeIncidentResolvedMs</code>	<p>問題が解決したことを <b>NNMi Causal Engine</b> が検出したときのタイムスタンプ (ミリ秒単位)。</p>

**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 5 の示した CIA を追加します。たとえば **NodeDown** インシデントには、**InterfaceDown** インシデントと **AddressDown** 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、以下のデフォルト管理イベントインシデントタイプに、表 5 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回答して **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
- 以下のファミリのインシデント

- 相関処理
- ライセンス
- NNMi ヘルス
- トラップ分析

## インシデント数の削減

NNMiには、ネットワークオペレーターが NNMi コンソールで見るインシデント数を削減する以下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 翌 1 日以内のインシデントが別のインシデントによってキャンセルされます。
- **重複解除相関処理** 設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、重複解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** 設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。

## 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、以下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することができます。

- **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、他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**（負荷量）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によってそのインシデントのアクティビティが遅延されます。インシデントのダンプニングには、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あり、NNMi コンソール内のインシデント数を減らし、より意味のあるインシデントにする上で便利です。

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以下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、以下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のチェック：
  - 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、一致内で最下位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のチェック：
  - 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、一致内で最下位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します (ある場合)。

## ライフサイクル移行アクション

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、1つのインシデントタイプの1つのライフサイクル状態に固有です。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り、実行するコマンドが特定されます。コマンドには引数を含めることができ、これによって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いずれかにすることができます。アクションコードは1つの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はしたりできます。たとえば、**ConnectionDown**、**NodeDown**、**NodeOrConnectionDown** のいず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ー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することができます。たとえば、**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** に基づいて負荷量フィルターを組み込んで、アクションが実行されることを制限できます。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。たと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負荷量フィルターをこの属性値に基づくようにすることができます。

## インシデントの計画

以下の領域で決定します。

- NNMi が処理するデバイストラップ
- NNMi で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
- NNMi による NNM 管理ステーションからのトラップ受信の可否
- NNMi による別のイベントレシーバーへのトラップ転送の可否

### NNMi が処理するデバイストラップ

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。MIB に TRAP-TYPE または NOTIFICATION-TYPE マクロが含まれる場合は、MIB で定義されたトラップにスケルトンインシデント設定を作成できます。

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。

### NNMi で表示するインシデント

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。

重複解除、レート設定、ペア関連処理によって削減できるインシデントを計画します。

### インシデントに対する NNMi の対応方法

インシデントが発生した場合の NNMi のアクション ( ネットワークオペレーターへの電子メール送信など ) 各アクションを実行するライフサイクルの状態

### NNMiによるNNM管理ステーションからのトラップ受信の可否

NNMiと連動してネットワーク領域の管理を継続する1つ以上のNNM 6.x/7.x管理ステーションが環境に含まれる場合は、NNMi オペレーターのネットワーク管理をサポートする NNM 6.x/7.x イベントを識別します。NNMi コンソールで使用できる各 NNM 6.x/7.x イベントのインシデント設定を計画します。

### NNMiによる別のイベントレシーバーへのトラップ転送の可否

環境にサードパーティのトラップ統合が含まれる場合は、NNMi SNMP トラップ転送メカニズムを NNMi ノースバウンドインタフェース SNMP トラップ転送メカニズムと一緒に使用するかどうかを決定します。

NNMi ノースバウンドインタフェース SNMP トラップ転送メカニズムを選択する場合は、NNMi がイベントレシーバーに転送するすべてのトラップの MIB をロードします。

## インシデントの設定

ここでは、設定のヒントを一覧にし、いくつかの設定例について説明します。このセクションの情報を読んだ後で、具体的な手順の 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。



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、「[ベストプラクティス: 既存の設定を保存](#)」(34 ページ)を参照してください。

- 計画したインシデントタイプを設定します。可能な場合は、MIB で定義したトラップのスケルトンインシデント設定から開始します。
- トラップ転送に必要な MIB をすべてロードします。
- NNMi 管理サーバーにトラップを送信するデバイスが設定されていることを確認します。

### インシデントの抑制、強化、およびダンプニングの設定

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、以下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負荷量フィルターを指定できます。
- 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設定を設定します。
- 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設定を設定します。
- 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** ] タブにデフォルト設定を設定します。

### ライフサイクル移行アクションの設定

ライフサイクル移行アクションを設定するときは、以下に注意してください。

- デフォルトでは、NNMi は以下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がない場合は、[ **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。



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。

## トラップログの設定

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。トラップは以下の場所に記録されます。

- Windows: %NnmDataDir%\nnm¥log
- UNIX: \$NNM\_DATA/nnm/log

トラップログファイルは、nmtrapconfig.ovpl スクリプトを使用して設定します。以下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)。
- TXT: トラップは TXT 形式で記録されます (trap.log)。
- BOTH: トラップは CSV と TXT 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。

たとえば、BOTH モードでトラップを記録するように指定する場合は、以下のコマンドを使用します。

```
nmtrapconfig.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のみ有効です。

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 (nmtrapconfig.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、nmtrapconfig.ovpl リファレンスページまたは UNIX のマン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」も参照してください。

## インシデントログの設定

受信インシデント情報が incident.log 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。

## トラップサーバープロパティの設定

トラップサーバープロパティ (nmtrapserver.properties) を設定するには、nmtrapconfig.ovpl スクリプトを使用します。



nmtrapserver.properties ファイルが存在するファイルディレクトリは編集しないでください。nmtrapconfig.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以下のデフォルト値が設定されています。

**表 6**   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	1097
com.hp.ov.nms.trapd.trapInterface	すべてのインタフェース
com.hp.ov.nms.trapd.recvSocketBufSize	2048 キロバイト
com.hp.ov.nms.trapd.pipeline.qSize	50000 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
com.hp.ov.nms.trapd.database.fileCount	5 ファイル
com.hp.ov.nms.trapd.database.qSize	300000 トラップ
com.hp.ov.nms.trapd.discohint.cacheSize	5000 エントリー
com.hp.ov.nms.trapd.discohint.cacheEntryTimeout	3600 ミリ秒

詳細については、`nmstrapconfig.ovpl` リファレンスページまたは **UNIX** のマンページを参照してください。



## 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。

### nnmincidentcfgdump.ovpl によるインシデント設定ファイルの生成

NNMi nnmincidentcfgdump.ovpl スクリプトでは、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。

以下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、以下の構文の例を使用します。

```
nnmincidentcfgdump.ovpl -dump <file_name> -u <NNMiadminUsername>
-p <NNMiadminPassword>
```

詳細については、nnmincidentcfgdump.ovpl リファレンスページ、または UNIX のマンページを参照してください。

### nnmincidentcfgload.ovpl によるインシデント設定のロード

NNMi nnmincidentcfgload.ovpl スクリプトでは、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。



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。

必要な形式については、以下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、以下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、以下の構文の例を使用します。

```
nnmincidentcfgload.ovpl -load <file_name> -u <NNMiadminUsername>
-p <NNMiadminPassword>
```

以下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。



NNMi は、これらの設定に関連付けられたコード値 (インシデントファミリーなど) の上書きも行います。

- NNMi は、NNMi データベースに存在し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、`nmmincidentcfgload.ovpl` リファレンスページ、または UNIX のマンページを参照してください。

## 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。



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ており、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いずれかを使用しても、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。

## 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。以下のメソッドのいずれかを使用します。

- 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。
- [管理対象外] または [サービス停止中] を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は廃棄されます。
- 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は廃棄されます。

- 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。

## 未定義トラップのインシデントの有効化および設定

NNMiは、デフォルトで未定義トラップをサイレントにドロップします。NNMi 9.01以降、NNMiは、ドロップされる可能性がある未定義SNMPトラップを特定できるようになります。



NNM iSPI NET が NNMi 管理サーバーでライセンス供与されている場合は、Total Traps Received (by OID) レポートを使用して、ドロップされたSNMPトラップを調べます。詳細については、NNMi ヘルプの「トラップ情報を分析する (NNM iSPI NET)」を参照してください。

NNM iSPI NET が NNMi 管理サーバーでライセンス供与されておらず、インシデントとして欠落したトラップを確認する場合は、未定義SNMPトラップインシデントを以下のように設定します。

- 以下のファイルを編集します。
  - Windows:** %NNM\_PROPS%\nms-jboss.properties
  - UNIX:** \$NNM\_PROPS/nms-jboss.properties
- ファイルから、以下の行のようなセクションを特定します。  

```
#!com.hp.nnm.events.allowUndefinedTraps=false
```

 この行を以下のように変更します。  

```
com.hp.nnm.events.allowUndefinedTraps=true
```
- オプション。nms-jboss.properties ファイルで説明されている値を使用し、インシデントの重大度を指定します。ファイルから、以下の行のようなセクションを特定します。  

```
#!com.hp.nnm.events.undefinedTrapsSeverity=NORMAL
```

 この行を以下のように変更し、定義した重大度の値を **YourSpecifiedSeverity** の代わりに使用します。  

```
com.hp.nnm.events.undefinedTrapsSeverity=YourSpecifiedSeverity
```
- オプション。nms-jboss.properties ファイルで説明されている値を使用し、インシデントの特性を指定します。ファイルから、以下の行のようなセクションを特定します。  

```
#!com.hp.nnm.events.undefinedTrapsNature=INFO
```

 この行を以下のように変更し、定義した特性の値を **YourSpecifiedNature** の代わりに使用します。  

```
com.hp.nnm.events.undefinedTrapsNature=YourSpecifiedNature
```
- 以下のコマンドを実行して NNMi を再開します。
  - ovstop**
  - ovstart**

- 6 未定義トラップのリストを検討し、制御するトラップ用に新しいインシデント設定を作成します。NNMiで新しいインシデントを表示する場合はそれを有効にして、NNMiで新しいインシデントを無視する場合はそれを無効にします。詳細については、NNMi ヘルプの「SNMP トラップインシデントを設定する」を参照してください。

# NNMi コンソール

本章の情報を読み、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理解してください。

本章には、以下のトピックがあります。

- ノードグループの実例的な使用例
- 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の削減
- ノードグループマップの表示ノード数の削減
- [分析]ペインのゲージの設定

---

## ノードグループの実例的な使用例

以下に、ノードグループを設定する場合の実例的な例を示します。

**My Network:** 他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:** 他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。

以下の点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ベストプラクティスです。
- 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つ設定するのがベストプラクティスです。マップにより、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。
- この例において、**Colorado** はノードが含まれている唯一のノードグループです。
- NNMi では、いく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。このドキュメントで説明するステップに精通すれば、後続のノードグループやノードグループマップをより効率よく作成する方法を見つけることもできます。

このドキュメントでは、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以下の手順について説明します。また、ノードグループを削除する手順についても説明します。

### ノードグループの作成

- ステップ 1: **My Network** ノードグループを作成する
- ステップ 2: **USA** ノードグループを作成する
- ステップ 3: フィルターを使用して **Colorado** ノードグループを作成する
- ステップ 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ー結果を確認する
- ステップ 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
- ステップ 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。その代わりに、定義に子ノードグループのみが含まれています。この例では、My Network および USA ノードグループが、子ノードグループのみ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

- ステップ 1: ノードグループマップを作成する
- ステップ 2: ノードグループマップを表示する
- ステップ 3: ノードグループのステータスを設定する
- ステップ 4: ノードグループマップの順序を設定する
- ステップ 5: ノードグループマップに背景イメージを追加する

### ノードグループの削除

- ステップ 1: ノードグループに移動する
- ステップ 2: ノードグループを削除する

## ノードグループの作成

ノードグループを作成してノードグループマップに含める作業から説明を始めます。

### ステップ 1: My Network ノードグループを作成する

**My Network** ノードグループを作成するには、以下の手順を実行します。

- 1 **[設定]** ワークスペースに移動します。
- 2 **[ノードグループ]** を選択します。
- 3 **[新規作成]** アイコンをクリックします。
- 4 **[名前]** 属性に、「**My Network**」と入力します。
- 5 **[注]** 属性に、「**最上位のノードグループです**」と入力します。
- 6 **[保存して閉じる]** をクリックしてこの設定を保存します。

## ステップ 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します。
- 2 [ノードグループ] を選択します。
- 3 [新規作成] アイコンをクリックします。
- 4 [名前] 属性に、「USA」と入力します。
- 5 [保存して閉じる] をクリックしてこの設定を保存します。

## ステップ 3: フィルターを使用して 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、フィルターエディターを使用してノードを選択するフィルターを設定します。



可能であれば、[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ーを使用すると、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します。
- 2 [ノードグループ] を選択します。
- 3 [新規作成] アイコンをクリックします。
- 4 [名前] 属性に、「Colorado」と入力します。
- 5 [追加のフィルター] タブを選択します。
- 6 ノードが入力したホスト名値のいず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] をクリックします。
- 7 フィルターエディターの [属性] フィールドで、[hostname] を選択します。  
[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 がホスト名値を照合するように指定されます。
- 8 [演算子] フィールドで、[like] を選択します。  
[like] を選択すると、検索でワイルドカード文字を使用できます。
- 9 [値] フィールドに、ノードグループに含めるデバイスを表す値を入力します。たとえば、`cisco*.ntc.example.com` は、`cisco`<このテキストで置き換える>.<network\_domain> という名前のデバイスを表します。
- 10 [追加] をクリックします。
- 11 [属性] フィールドで、[hostname] を選択します。
- 12 [演算子] フィールドで、[like] を選択します。
- 13 [値] フィールドに、Colorado ノードグループに追加する残りのデバイス名を表すワイルドカードを入力します。この例では、「`cisco?*`」を使用します。
- 14 [追加] をクリックします。
- 15 [保存] をクリックして、ウィンドウを閉じずにノードグループを保存します。

## ステップ 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ー結果を確認する

ノードグループフィルターをテストするため、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。



ノードグループフィルターが正しく機能すると確信できるまで、ノードグループフィルター定義の結果を調べてください。

## ステップ 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

最上位レベルの **My Network** ノードグループを初め、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 [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。
- 2 **My Network** ノードグループに移動して、[開く] をクリックします。
- 3 [子ノードグループ] タブをクリックします。
- 4 [新規作成] アイコンをクリックします。
- 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] を選択します。



[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します。
- 7 [OK] をクリックします。
- 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] フォームを閉じます。
- 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] フォームを閉じます。

## ステップ 6: USA ノードグループのノードグループ階層を作成する

次に、**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。ステップ 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説明にあるのと同じステップを繰り返して行い、**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。



## ノードグループマップの設定

### ステップ 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きます。
  - a [設定]ワークスペースの[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します。
- 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ます。
- 4 [レイアウトの保存]アイコンをクリックして、ノードグループマップを作成します。



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]により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]をクリックします。
- 6 作成した各ノードグループで、ステップ 1～5 までを繰り返します。

### ステップ 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します。
- 2 [ノードグループの概要]を選択します。
- 3 最上位レベルマップ [My Network] を選択します。
- 4 アイコンをダブルクリックして、子ノードグループマップに移動します。
- 5 ツールバーの上にある階層リンクを使用して前のマップに戻ります。

### ステップ 3: ノードグループのステータスを設定する

NNMiにより、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、以下の中から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。



[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します。
- 2 [ステータスの設定]を選択します。

- 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してください。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。

## ステップ 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します。
- 2 [ **ユーザーインターフェース** ] > [ **ノードグループマップの設定** ] を選択します。



以下の例に示すように、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** ] の値を他のどのマップの [ **トポロジマップ順序** ] の値よりも小さい数字 (たとえば 5) にします。

- 3 **My Network** ノードグループマップを開きます。
- 4 [ **トポロジマップ順序** ] 属性で、値を 5 に変更します。
- 5 [ **保存して閉じる** ] をクリックして変更を保存し、フォームを閉じます。

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。そのためには、[ **設定** ] ワークスペースで [ **ユーザーインターフェースの設定** ] オプションを使用します。

- 1 [ **設定** ] ワークスペースに移動します。
- 2 [ **ユーザーインターフェースの設定** ] をクリックします。
- 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** ] を選択します。
- 4 [ **保存して閉じる** ] をクリックして変更を保存し、フォームを閉じます。

これにより、**My Network** マップが初期ビューとして表示されます。

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。

## ステップ 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します。
- 2 [ **ユーザーインターフェース** ] > [ **ノードグループマップの設定** ] をクリックします。
- 3 **My Network** ノードグループマップを開きます。
- 4 [ **背景イメージ** ] タブに移動します。
- 5 [ <http://MACHINE:PORT/nnmdocs/images/> ] をクリックします。

NNMi に、HP が提供するグラフィックの一覧が表示されます。

- 6 **world.png** リンクを右クリックします。
- 7 **[リンクの場所をコピー]** を選択します。
- 8 ディレクトリのリストウィンドウを閉じます。  
コピーしたリンクを **[背景イメージ]** 属性に貼り付けます。



後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。

- 9 **[保存して閉じる]** をクリックして変更を保存します。
- 10 **[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します。

## ノードグループの削除

ノードグループを削除する場合を考えます。たとえば、この例で先ほど作成した **Colorado** ノードグループを削除します。

### ステップ 1: ノードグループに移動する

- 1 **[設定]** ワークスペースで、**[ノードグループ]** をクリックします。
- 2 リストで **Colorado** ノードグループを選択し、**[開く]** ボタンをクリックします。

### ステップ 2: ノードグループを削除する

- 1 **[ノードグループの削除]** ボタンをクリックします。
- 2 ダイアログボックスが表示されて、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。
- 3 **[OK]** をクリックしてノードグループを削除します。

---

## 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の削減

**[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すぎると、ノードを移動するときのマップの反応が遅くなったり、複雑すぎて実際の表示に適さなくなったりする可能性があります。

**[ネットワークの概要]** マップに表示する最大ノード数を増減させることが可能です。これを行うには、**[ユーザーインターフェースの設定]** フォームの **[デフォルトのマップ設定]** タブにある **[表示するノードの最大数]** 属性を編集します。

**[ネットワークの概要]** マップに表示する最大ノード数の増減は、次に示す例の手順を実行しても行うことができます。

**[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を 250 から 100 に変更するとします。これを行うには、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - **Windows:** %NNM\_PROPS%\nms-ui.properties
  - **UNIX:** \$NNM\_PROPS/nms-ui.properties
- 2 以下の行に似たテキストを特定します。
 

```
#!com.hp.nnm.ui.networkOverviewMaxNodes = 250
```

 行を以下のように変更します。
 

```
com.hp.nnm.ui.networkOverviewMaxNodes = 100
```

 行の始めにある **#!** 文字を必ず削除してください。
- 3 変更を保存します。



## ノードグループマップの表示ノード数の削減

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可能性があります。

解決方法は、以下の手順を実行して、表示されるノードまたは表示されるエンドポイント、あるいはその両方の数を制限することです。

- 1 NNMi コンソールで、[ **設定** ] をクリックします。
- 2 [ **ユーザーインターフェース** ] の下にある [ **ユーザーインターフェースの設定** ] をクリックします。
- 3 [ **デフォルトのマップ設定** ] タブを選択します。
- 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します。
- 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します。
- 6 [ **保存して閉じる** ] をクリックします。

詳細については、NNMi ヘルプの「デフォルトマップ設定を定義する」を参照してください。

## [ 分析 ] ペインのゲージの設定

[ 分析 ] ペインの [ ゲージ ] タブには、**StatePoller** とカスタムポーラーの **SNMP** データを示すために、リアルタイムの **SNMP** ゲージが表示されます。これらのゲージには、ノード、インタフェース、カスタムノード収集のデータや、**CPU**、**メモリー**、**バッファ**、**バックプレーン**タイプのノードコンポーネントのデータが表示されます。

以下のプロパティファイルを編集してゲージを設定できます。

- **Windows:** %NNM\_PROPS%\nms-ui.properties
- **UNIX:** \$NNM\_PROPS/nms-ui.properties

設定する各プロパティで、行の始めにコメント文字 (#!) が存在する場合は削除します。



後続の項で説明するプロパティはすべてのノードに適用されます ( 個別のノードグループにプロパティを適用することはできません )。



変更を行う前に nms-ui.properties ファイルのバックアップコピーを作成します。バックアップコピーは、編集するプロパティファイルが格納されているディレクトリに配置しないでください。

詳細については、nms-ui.properties ファイル内のコメントも参照してください。

### 表示されるゲージ数の制限

以下の行を編集して目的の値を入力し、表示するゲージの最大数を設定します。

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



ゲージ数が多いほど、分析ペインの表示時のパフォーマンスに影響します。ゲージ数が少ないほどゲージのサイズが大きくなります。

### [ 分析 ] ペインにあるゲージの更新間隔の設定

以下のプロパティ値を編集して、[ 分析 ] ペインに表示されるゲージの更新間隔 ( 秒 ) を設定します。

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



値を「0」に設定すると、ゲージが更新されなくなります。更新間隔を 10 秒より速くすると、一部の **SNMP** エージェントでは短時間で値がキャッシュされ、結果が同じになります。

### ゲージの非表示

以下の行を編集し、非表示にするゲージのリストを入力して、( すべてのゲージビューの ) 表示しないゲージを定義します。

```
com.hp.nnm.ui.analysisGaugeNoDisplayKeyPatterns =
```

関連するすべての行のコメントを解除してください。ゲージのリスト内にコメントを含めることはできません。また、空白行があるとその場所でエントリーが終了するため、ゲージのリスト内に空白行を含めないでください。

コメント内の設定がこのプロパティのデフォルト設定です。この設定を拡張または修正する場合、これらの設定を含める必要があります。含めないと、予期しない数のゲージが表示されます。

## 表示されるノードゲージの順序の制御

以下の行を編集して、ノードゲージが表示される順序を制御できます。

```
com.hp.nnm.ui.analysisGaugeNodeComponentKeys =
```

このプロパティ設定では、ワイルドカードはサポートされていません。リストにコメントまたは空白行が含まれていないことを確認してください。

コメント内の設定がこのプロパティのデフォルト設定です。この設定を拡張または修正する場合、これらの設定を含める必要があります。含めないと、意図した順序で表示されません。

## 表示されるインタフェースゲージの順序の制御

以下の行を編集して、インタフェースゲージが表示される順序を制御できます。

```
com.hp.nnm.ui.analysisGaugeInterfaceKeys =
```

このプロパティ設定では、ワイルドカードはサポートされていません。リストにコメントまたは空白行が含まれていないことを確認してください。

コメント内の設定がこのプロパティのデフォルト設定です。この設定を拡張または修正する場合、これらの設定を含める必要があります。含めないと、意図した順序で表示されません。

## 表示されるカスタムポーラーゲージの順序の制御

以下の行を編集して、カスタムポーラーゲージが表示される順序を制御できます。

```
com.hp.ov.nnm.ui.analysisGaugeCustomPolledInstanceKeys =
```

この属性にデフォルト設定はありません。

## ゲージプロパティの適用方法の理解

ゲージプロパティは以下の順序で適用されます。

- 1 すべてのゲージのリストが **StatePoller** から取得されます。
- 2 `analysisGaugeNoDisplayKeyPatterns` が最初に適用されて、指定のゲージがリストから削除されます。
- 3 `analysisGaugeNodeComponentKeys`、`analysisGaugeInterfaceKeys`、または `analysisGaugeCustomPolledInstanceKeys` が必要に応じて適用され、表示されるゲージのリストの順序が決まります。
- 4 最後に、`maxGaugePerAnalysisPanel` が適用されて、表示されるリストが切り捨てられます。

## ゲージ名の判別

ゲージを含めたり、抑制したり、順序を決めたり、トラブルシューティングしたりするには、ゲージ名を把握する必要があります。以下のようにゲージ名を判別します。

- 1 **http://<nnmiHost>/jmx-console** で **JMX** コンソールを起動します。
- 2 ページで以下のいずれかを検索します。
  - com.hp.ov.nms.statePoller ( ノードおよびインタフェースのゲージ )
  - com.hp.ov.nms.customPoller ( カスタムポーラーのゲージ )
- 3 **[コレクター] mbean** をクリックします。
- 4 関数 `dumpCollectionsMatchingTopologyObjectAndPolicy` を検索し、パラメータ値を入力せずにその下にある **[呼び出し]** をクリックします。これにより、NNMi システムの `tmp` ディレクトリにファイルが作成されます。
- 5 このファイルを開き、該当のノードを検索します。次に、ノードに関連付けられている収集情報を探します。次に例を示します。

columnsToCollect:

```
Type: SNMPInstrumentationVariable, Name: sysUpTime, Value:
.1.3.6.1.2.1.1.3
```

```
Type: SNMPInstrumentationVariable, Name: cpu5s, Value:
.1.3.6.1.4.1.9.9.109.1.1.1.3
```

```
Type: SNMPInstrumentationVariable, Name: cpu1m, Value:
.1.3.6.1.4.1.9.9.109.1.1.1.4
```

```
Type: SNMPInstrumentationVariable, Name: cpu5m, Value:
.1.3.6.1.4.1.9.9.109.1.1.1.5
```

このリストで収集の名前 (ゲージの名前) を参照できます。

## ゲージに関する問題のトラブルシューティング

### ゲージが多すぎる

`maxGaugePerAnalysisPanel` プロパティを使用して表示されるゲージの数を制限するか、`analysisGaugeNoDisplayKeyPatterns` プロパティを使用して不要なゲージを削除します。

### ゲージが表示されない

**JMX** コンソールを使用して、**StatePoller** の収集をダンプし、デバイスで実行されている収集とその名前を確認します。デバイスで収集がサポートされていない可能性があります。たとえば、`cpu1m` は特定のデバイスで使用できません。

---

## デバイスのプロファイルアイコンのカスタマイズ

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付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。

`nnmicons.ovpl` コマンドを使用して 1 つ以上のアイコンをカスタマイズできます。詳細については、`nnmrestore.ovpl` リファレンスページ、または UNIX のマンページを参照してください。NNMi ヘルプも参照してください。



# 詳細設定

この項では以下の章について説明します。

- NNMi のライセンス
- NNMi での証明書の使用
- NNMi とシングルサインオンの使用
- NNMi で使用する Telnet および SSH プロトコルを設定する
- NNMi と LDAP によるディレクトリサービスの統合
- NNMi セキュリティおよびマルチテナント
- グローバルネットワーク管理
- IPv6 用 NNMi Advanced の設定
- Solaris ゾーン環境での NNMi の実行
- NNMi Northbound インタフェース



# NNMiのライセンス

恒久ライセンスキーをインストールしていない場合は、NNMi 製品には、NNMi のインストール後 60 日間有効な一時試用ライセンスキーが含まれています。この一時試用ライセンスキーを使用すると、NNMi Advanced 機能を使用できるようになります。できるだけ早く、恒久ライセンスキーを入手し、インストールしてください。

NNMi Advanced ライセンスに含まれている機能のリストを表示するには、『HP NNMi Software リリースノート』の「ライセンス」のセクションを参照してください。

## 恒久ライセンスキーのインストール準備

試用ライセンスでは、250 ノードまでの制限が付けられています。試用ライセンスキーで NNMi を実行している場合、恒久ライセンスでサポートできる数以上のノードを管理できる場合があります。ただし、恒久ライセンスが有効になると、ライセンス制限を超えた分のノードは NNMi により自動的に管理対象外になります。

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

## ライセンスの種類および管理対象ノードの数の確認

現在、NNMi が使用しているライセンスの種類を確認するには、以下の手順を実行します。

- 1 NNMi コンソールで、[ヘルプ]>[Network Node Manager について]の順にクリックします。
- 2 [Network Node Manager について] ウィンドウで、[ライセンス情報の表示]をクリックします。  
([ライセンス情報の表示]は、[NNMi コンソールのサインイン]ページから入手することもできます。)
- 3 [消費量]フィールドに表示されている値を探します。この値が、現在 NNMi が管理しているノードの数です。

- 4 恒久ライセンスがサポートできるノード数が、現在 NNMi が管理しているノード数より少ない場合は、NNMi コンソールを使用して、あまり重要でないノードを削除します。詳細については、NNMi ヘルプの「ノードの削除」を参照してください。

## 恒久ライセンスキーの取得およびインストール

恒久ライセンスキーを申請するには、以下の情報が必要です。

- HP 製品番号や製造番号が明記されたエンタイトルメント証明書
- NNMi 管理サーバーの 1 つの IP アドレス
- HA で動作する NNMi のライセンスの場合は、NNMi HA リソースグループの仮想 IP アドレス
- お客様の企業情報もしくは団体情報

### Autopass および HP 注文番号の使用 (ファイアウォール使用時は不可)

恒久ライセンスキーを入手してインストールするには、以下の手順に従ってください。

- 1 コマンドプロンプトで、以下のコマンドを入力し、Autopass ユーザーインターフェースを開きます。  
`nnmlicense.ovpl NNM -gui`
- 2 [Autopass] ウィンドウの左側にある [ライセンス管理] をクリックします。
- 3 [ライセンスキーのインストール] をクリックします。
- 4 [ライセンスキーの取得/インストール] をクリックします。
- 5 HP 注文番号を入力し、Autopass プロンプトに従ってライセンスキーの取得プロセスを完了します。
- 6 NNMi により、インストールが自動的に完了します。

### コマンド行で、シードを追加する

自動プロセスが完了しない場合は (NNMi 管理サーバーがファイアウォールの背後にある場合など)、以下の手順を実行します。

- 1 ライセンスキーを取得するには、以下の HP パスワード配信サービスに移動します。  
`https://webware.hp.com/welcome.asp` (英語サイト)
- 2 NNMi 管理サーバーのコマンドプロンプトで以下のコマンドを入力し、システムを更新して、ライセンスデータファイルを保存します。  
`nnmlicense.ovpl NNM -f license_file`  
(製品ライセンス ID (NNM) では大文字と小文字が区別されます。)  
詳細については、`nnmlicense.ovpl` のリファレンスページまたは UNIX のマンページを参照してください。
- 3 NNMi により、インストールが自動的に完了します。

## 追加のライセンスキーを取得する

NNMi ライセンス構造に関する詳細について HP 営業担当または Hewlett-Packard 正規販売店に問い合わせ、企業向けインストールにライセンス層を追加する方法について調べます。

追加のライセンスキーを取得するには、HP ライセンスキー配信サービスに移動します。

**<https://webware.hp.com/welcome.asp> (英語サイト)**

詳細については、NNMi ヘルプの「ライセンス容量を拡張する」を参照してください。

**開発者の方へ:** NNMi 開発者ツールキットを使用すると、カスタム Web サービスクライアントを統合して NNMi の機能を拡張できます。NNMi 開発者ライセンスをインストールすると、NNMi により doc フォルダに sdk-dev-kit.jar ファイルが作成されます。sdk-dev-kit.jar ファイルを解凍すると、NNMi 開発者ツールキットドキュメントやサンプル集を表示できます。



# 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。

管理者は、ネットワークから NNMi への HTTP やその他の非暗号化アクセスを無効にできます。「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」(411 ページ) を参照してください。

この章には、以下のトピックがあります。

- [すべてをまとめる](#)
- [認証機関証明書を生成する](#)
- [自己署名証明書を使用するようにアプリケーションフェイルオーバーを設定する](#)
- [認証機関を使用するようにアプリケーションフェイルオーバーを設定する](#)
- [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を設定する](#)
- [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](#)
- [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](#)
- [自己署名証明書を使用するようにアプリケーションフェイルオーバーが有効なグローバルネットワーク管理を設定する](#)
- [ディレクトリサービスへの SSL 接続を設定する](#)

## すべてをまとめる

以下の情報に従い、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、「[認証機関証明書を生成する](#)」(129 ページ)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する場合は、追加の設定手順があります。「[自己署名証明書を使用するようにアプリケーションフェイルオーバーを設定する](#)」(132 ページ)の説明にあるグローバルネットワーク管理設定を完了する前に、クラスターごとに NNMi 管理サーバーの `nnm.keystore` 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る、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、「[認証機関証明書を生成する](#)」(129 ページ)の説明にある手順を実行し、次に「[認証機関を使用するようにアプリケーションフェイルオーバーを設定する](#)」(134 ページ)の説明にあるグローバルネットワーク管理設定を完了する前に、クラスターごとに NNMi 管理サーバーの `nnm.keystore` 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、「[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を設定する](#)」(136 ページ)の説明にあるグローバルネットワーク管理設定を完了する前に、`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ーを正しく設定したら、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、それからトラストストアをマージすることにより、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。「[自己署名証明書を使用するようにアプリケーションフェイルオーバーが有効なグローバルネットワーク管理を設定する](#)」(139 ページ)の情報を確認してください。NNMi 管理サーバーで「[認証機関証明書を生成する](#)」(129 ページ)の説明にある手順を使用して生成した 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証明書はそれらの CA 証明書のみです。
- 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、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ーに含めるよう変更する場合は、「[自己署名証明書を使用するようにアプリケーションフェイルオーバーを設定する](#)」(132 ページ)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、リージョナル、グローバル、またはその両方で HA を使用するように変更する場合は、「[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を設定する](#)」(136 ページ)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、「[ディレクトリサービスへの SSL 接続を設定する](#)」(140 ページ)の指示に従ってください。



## 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、以下の手順で CA 証明書を生成します。

▶ 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変更します。
  - 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します。
- 3 システムからプライベートキーを生成します。このプライベートキーを生成するには、keytool コマンドを使用します。
  - a 以下のコマンドをそのまま実行します。

— Windows: %NnmInstallDir%\nonOV\jdk\%nnm%\bin\keytool.exe  
-genkeypair -validity 3650 -keyalg rsa -keystore  
nnm.keystore -storepass nnmkeypass -alias  
myserver.mydomain

— UNIX: \$NnmInstallDir/nonOV/jdk/nnm/bin/keytool  
-genkeypair -validity 3650 -keyalg rsa -keystore  
nnm.keystore -storepass nnmkeypass -alias  
myserver.mydomain

▶ 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することができますが、HP では、*myserver.mydomain* 別名の変数として、ご使用のシステムの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。



**重要:** 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 以下のコマンドをそのまま実行して、CSR ( 証明書署名要求 ) ファイルを作成します。
  - 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



keytool コマンドの詳細については、<http://www.oracle.com/technetwork/java/index.html> で「鍵および証明書管理ツール」を検索してください。

- 5 CA 署名機関に CSR を送信します。以下のいずれかが発行されます。

- `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。
- この手順における、`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。

以下は、**CA** 署名機関から受け取るファイルの例です。

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/AVQQKExNQ0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLLEwdOZXR3b3Js
eGV5ZXZvY2F0aW9uTG1zdD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJldGJw
.....
TZImiZPyLQQBGRYDaW50MRIwEAYKCZImiZPyLQQBGRYCC2cxEzARBgNVBAMTCmNb
pSo6o/76yShtT7Vr1fz+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/VQQKExNQ0EgQ29ycG9yYXRpb24gTHRkMRAwDgYDVQQLLEwdOZXR3b3Js
eGV5ZXZvY2F0aW9uTG1zdD9iYXNlP29iamVjdENsYXNzPWNSTERpc3RyaWJldGJw
.....
TZImiZPyLQQBGRYDaW50MRIwEAYKCZImiZPyLQQBGRYCC2cxEzARBgNVBAMTCmNb
pSo6o/76yShtT7Vr1fz+mXjWyEHaIy/QLCpPebYhejHEg4dZgzWWT/lQt==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
Sample2/Gh0dHA6Ly9jb3JwMWRjc2cyLnNnLmludC5wc2FnbG9iYWwY29tL0Nlc
RaOCApwwggKYYMB0GA1UdDgQWBBSqaWZzCRcpvJWOFpZ/Be9b+QSPyDafBgNVHSMC
.....
Wp5Lz1zJA0u1VHbPVdQnXnlBkx7V65niLoat90Eqd6laliVlJHj7GBriJ90uvVGu
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** 管理サーバーのいずれかの場所にコピーします。この例では、以下の場所にファイルをコピーします。

- **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変更します。

- Windows: %NNM\_DATA%\shared\%nnm%\certificates
- UNIX: \$NNM\_DATA/shared/nnm/certificates

- 2 以下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します。

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



-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** と入力します。

このコマンドによる出力形式は以下のとおりです。

```
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 以下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します。

— 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
```

- 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** と入力します。

証明書をキーストアにインポートするときの出力例

6 トラストストアの内容を確認します。

- Windows:  
`%NnmInstallDir%\nonOV\jdk\nnm\bin\keytool -list ¥  
 -keystore nnm.truststore`
- UNIX:  
`$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -list ¥  
 -keystore nnm.truststore`

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以下のとおりです。

```
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。

7 以下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`

8 `com.hp.ov.nms.ssl.KEY_ALIAS` 変数を、`myserver.mydomain` で使用した値に更新します。忘れずに設定内容を保存してください。

9 以下のコマンドを実行して NNMi を再開始します。

- a `ovstop`
- b `ovstart`

10 次の構文を使用して NNMi コンソールへの HTTPS アクセスをテストします。

`https://<完全修飾ドメイン名>:<ポート番号>/nnm/` ブラウザーによって CA が信頼されると、NNMi コンソールへの HTTPS 接続が信頼されます。

## 自己署名証明書を使用するようにアプリケーションフェイルオーバーを設定する

図3 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` 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` ファイルにする必要があります。以下の手順を実行し、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、以下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で以下のディレクトリに変更します。
  - 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します。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で以下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` ファイルにマージします。

Windows:

```
nnmcertmerge.ovpl -keystore <keystore> -truststore
<truststore>
```

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します。これらのファイル保存場所は、以下のとおりです。
  - Windows: %NNM\_DATA%\shared\%nnm%\certificates
  - UNIX: \$NNM\_DATA/shared/nnm/certificates
- 5 Server XとServer Yの両方で以下のコマンドを実行します。完全修飾ドメイン名を含め、両方のサーバーからの表示結果が一致することを確認します。一致しない場合は続行せずに、手順1から手順7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
```

- 6 Server XとServer Yの両方で以下のコマンドを実行します。完全修飾ドメイン名を含め、両方のサーバーからの表示結果が一致することを確認します。一致しない場合は続行せずに、手順1から手順7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
```

- 7 432 ページの **手順 4** から、アプリケーションフェイルオーバー機能の設定を続行します。



**手順 4** で以下の自動アクションを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 NNMi\_active から NNM\_standby へ自動的に複製します。

## 認証機能を使用するようにアプリケーションフェイルオーバーを設定する

図 4 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 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 ファイルにする必要があります。以下の手順に従い、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、以下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 NNMi\_standby については、「[認証機関証明書を生成する](#)」（129 ページ）の手順に従います。
- 2 **手順 3** を完了する前に、Server Y で以下のディレクトリに変更します。
  - Windows: %NNM\_DATA%\shared\nnm\certificates
  - UNIX: \$NNM\_DATA/shared/nnm/certificates
- 3 nnm.keystore ファイルと nnm.truststore ファイルを Server Y から Server X の一時ファイル保管場所にコピーします。これ以降の手順では、これらのファイルの保管場所は <keystore> および <truststore> と呼びます。

- 4 Server Xで以下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` ファイルにマージします。

Windows:

```
nnmcertmerge.ovpl -keystore <keystore> -truststore
<truststore>
```

UNIX:

```
nnmcertmerge.ovpl -keystore <keystore> -truststore
<truststore>
```

- 5 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します。これらのファイル保存場所は、以下のとおりです。

- Windows: `%NNM_DATA%\shared\%nnm%\certificates`
- UNIX: `$NNM_DATA/shared/nnm/certificates`

- 6 Server X と Server Y の両方で以下のコマンドを実行します。hp.com 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せずに、手順 1 から手順 7 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
```

- 7 Server X と Server Y の両方で以下のコマンドを実行します。hp.com 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せずに、手順 1 から手順 7 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
```

- 8 432 ページの [手順 4](#) から、アプリケーションフェイルオーバー機能の設定を続行します。



135 ページの [手順 5](#) で以下の自動アクションを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情報を Server X から Server Y へ自動的に複製します。

## 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を設定する

図 5 HA での証明書の使用法



### 自己署名証明書を使用するように高可用性を設定する

NNMi HA を正しく設定するプロセスでは、プライマリノードとセカンダリ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。

### 新規証明書を使用するように高可用性を設定する

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。以下の手順を実行して、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、「共有 NNMi データ」(340 ページ) の説明に従って、NNMi に HA を設定する前または後に実行できます。

- 1 手順 2 を完了する前に、NNMi\_HA1 で以下のディレクトリに変更します。
  - Windows: `%NNM_DATA%\shared\nnm\certificates`
  - UNIX: `$NNM_DATA/shared/nnm/certificates`
- 2 NNMi\_HA1 で、以下のコマンドを実行して `newcert` を `nmm.keystore` ファイルにインポートします。
  - Windows: `%NnmInstallDir%\nonOV\jdk\nnm\bin\keytool -import -alias newcert_Alias -keystore nmm.keystore -file newcert`
  - UNIX: `$NnmInstallDir/nonOV/jdk/nnm/bin/keytool -import -alias newcert_Alias -keystore nmm.keystore -file newcert`
- 3 アクティブノード (NNMi\_HA1) とスタンバイノード (NNMi\_HA2) の両方で以下のファイルを編集します。
  - 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` ファイルで、以下の行を変更します。
 

```
com.hp.ov.nms.ssl.KEY_ALIAS = newcert_Alias
```
- 5 変更を保存します。

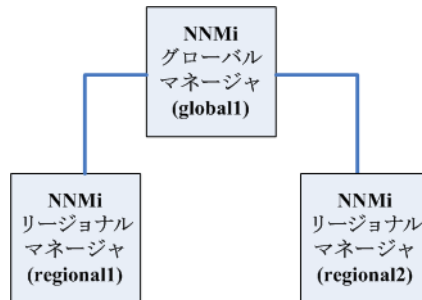


## 自己署名証明書を使用するようにグローバルネットワーク管理機能を設定する

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` 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図 6 のモデルを実現するとします。

図 6 グローバルネットワーク管理



以下の手順を実行し、図 6 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` で以下のディレクトリに変更します。
  - Windows: `%NNM_DATA%\shared\%nnm%\certificates`
  - UNIX: `$NNM_DATA/shared/nnm/certificates`
- 2 `nnm.truststore` ファイルを、上記の `regional1` および `regional2` の場所から、`global1` の任意の一時保管場所にコピーします。
- 3 `global1` で以下のコマンドを実行し、`regional1` および `regional2` の証明書を `global1` の `nnm.truststore` ファイルにマージします。

Windows:

- a `nnmcertmerge.ovpl -truststore regional1_nnm.truststore_location`
- b `nnmcertmerge.ovpl -truststore regional2_nnm.truststore_location`

UNIX

- a `nnmcertmerge.ovpl -truststore regional1_nnm.truststore_location`
- b `nnmcertmerge.ovpl -truststore regional2_nnm.truststore_location`

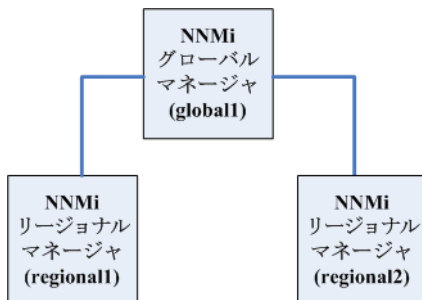
- 4 `global1` で、以下のコマンドを以下の順序で実行します。
  - a `ovstop`
  - b `ovstart`

## 認証機能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図 7 のモデルを実現するとします。

図 7 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、「[認証機関証明書を生成する](#)」(129 ページ) の手順に従います。
- 2 regional1 および regional2 の以下のディレクトリを変更してから、手順を実行します [手順 3](#)。
  - Windows: `%NNM_DATA%\shared\%nnm%\certificates`
  - UNIX: `$NNM_DATA/shared/nnm/certificates`
- 3 `nnm.truststore` ファイルを、上記の regional1 および regional2 の場所から、`global1` の任意の一時保管場所にコピーします。
- 4 `global1` で以下のコマンドを実行し、regional1 および regional2 の証明書を `global1` の `nnm.truststore` ファイルにマージします。

Windows:

- a `nnmcertmerge.ovpl -truststore regional1_nnm.truststore_location`
- b `nnmcertmerge.ovpl -truststore regional2_nnm.truststore_location`

UNIX

- a `nnmcertmerge.ovpl -truststore regional1_nnm.truststore_location`
- b `nnmcertmerge.ovpl -truststore regional2_nnm.truststore_location`

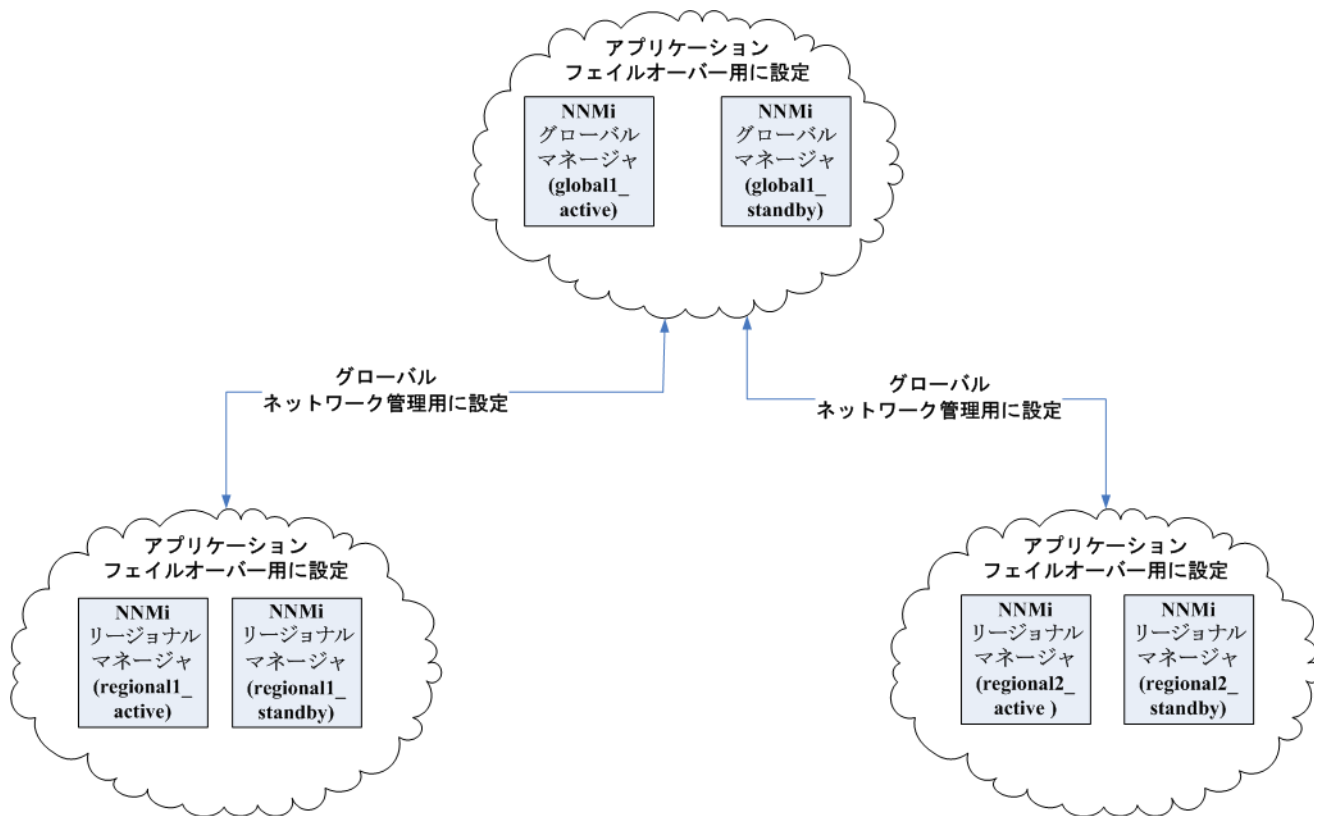
- 5 `global1` で、以下のコマンドを以下の順序で実行します。
  - a `ovstop`
  - b `ovstart`

## 自己署名証明書を使用するようにアプリケーションフェイルオーバーが有効なグローバルネットワーク管理を設定する

上の説明にあるように、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、[図 8](#) に示すアプリケーションフェイルオーバー機能のモデルを実現するとします。

図 8 アプリケーションフェイルオーバーが有効なグローバルネットワーク管理



以下の手順を実行し、上の図に基づいてアプリケーションフェイルオーバーが有効なグローバルネットワーク管理機能を設定します。

- 1 上の図に示すアプリケーションフェイルオーバークラスターごとに、「自己署名証明書を使用するようにアプリケーションフェイルオーバーを設定する」(132 ページ) に示す指示に従ってください。
- 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」(290 ページ) の指示に従ってアプリケーションフェイルオーバーを設定します。
- 3 「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」(137 ページ) に示す `regional1_active` and `regional2_active` に関する指示に従ってください。

## 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 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、以下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します。ディレクトリサービス管理者からこの証明書のテキストファイルのコピーを入手できます。
- 2 NNMi トラストストアが格納されているディレクトリに変更します。

- **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します。
  - a 以下のコマンドを実行します。

— Windows:

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

— UNIX:

```
$NnmInstallDir/nonOV/jdk/b/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。

- b キー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。

このコマンドによる出力形式は以下のとおりです。

```
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
```

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

4 トラストストアの内容を確認します。

- Windows:

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

- UNIX:

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

キー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以下のとおりです。

```
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 以下のコマンドを実行して NNMi を再開始します。

**a ovstop**

**b ovstart**

keytool コマンドの詳細については、<http://www.oracle.com/technetwork/java/index.html> で「鍵および証明書管理ツール」を検索してください。



# NNMi とシングル サインオンの 使用

HP Network Node Manager i Software (NNMi) シングルサインオン (SSO) を設定すると、NNMi コンソールから簡単に NNM iSPI にアクセスできるようになります。SSO を使用して NNMi コンソールにログオンすれば、NNM iSPI や他の HP アプリケーションにアクセスできます。再度ログインする必要はありません。SSO は、安全なアクセスレベルを維持しながら、より簡単に NNM iSPI や他の HP アプリケーションにアクセスできるようにする機能です。NNMi コンソールからサインアウト (または NNMi コンソールセッションがタイムアウト) した後に NNMi コンソールとは異なる NNM iSPI や他のアプリケーションの URL にアクセスするには、サインイン資格証明を再入力する必要があります。

インストール中に SSO は無効になっています。SSO が有効になっていても、ある NNMi 管理サーバーから別の管理サーバーへと参照すると、最初の管理サーバーからログアウトされ、利益はほとんどありません。これが起こらないようにするために、SSO は無効に初期設定されており、この章で説明されているように、NNMi 管理サーバー間で `initString` パラメーターと `protectedDomains` パラメーターの設定を調整できます。

この章には、以下のトピックがあります。

- 「[NNMi への SSO アクセス](#)」 (144 ページ)
- 「[1 つのドメインに対する SSO の有効化](#)」 (145 ページ)
- 「[異なるドメインに配置されている NNMi 管理サーバーに対する SSO の有効化](#)」 (145 ページ)
- 「[NNMi と NNM iSPI の SSO アクセス](#)」 (146 ページ)
- 「[SSO の無効化](#)」 (148 ページ)
- 「[SSO セキュリティに関する注意](#)」 (148 ページ)

## NNMi への SSO アクセス

複数の NNMi 管理サーバー間を移動するには、以下のいずれかを実行します。

- nms-ui.properties ファイルを編集して、com.hp.nms.ui.sso.initString と com.hp.nms.ui.sso.protectedDomains のパラメーター値を NNMi 管理サーバー間で同じ値にします。com.hp.nms.ui.sso.domain パラメーターを、NNMi 管理サーバーが配置されているドメインと一致するように設定してください。
  - NNMi 管理サーバーを1つのネットワークドメインにしか配置していない場合は、「[1つのドメインに対する SSO の有効化](#)」(145 ページ)の説明に従ってください。
  - NNMi 管理サーバーを複数のネットワークドメインに配置している場合、詳細については、「[異なるドメインに配置されている NNMi 管理サーバーに対する SSO の有効化](#)」(145 ページ)の説明に従ってください。
- nms-ui.properties file を編集し、SSO が無効になっていることを確認します。詳細については、「[SSO の無効化](#)」(148 ページ)を参照してください。

これらのアクションのいずれかが完了していないと、別の NNMi 管理サーバーに移動するたびに、直前の NNMi 管理サーバーから自動的にサインアウトします。

SSO と NNMi グローバルネットワーク管理機能を併用する場合、特別な考慮事項があります。詳細については、「[SSO およびアクションメニュー](#)」(243 ページ)および「[グローバルネットワーク管理用にシングルサインオンを設定する](#)」(243 ページ)を参照してください。

NNMi 管理サーバーのドメイン名が mycompany のように短く、ドット (.) がいない場合、NNMi コンソールによりただちにサインアウトされます。SSO ブラウザークッキーの制限には、mycompany.com のように、ドット (.) が少なくとも 1 つ付いているドメイン名が必要です。この状況を解決するには、以下の手順を実行します。

- 1 以下のファイルをテキストエディターで開きます。
  - **Windows:** %NNM\_PROPS%/nms-ui.properties
  - **UNIX:** \$NNM\_PROPS/nms-ui.properties

- 2 この例では、以下の文字列を検索します。

```
com.hp.nms.ui.sso.domain = mycompany
```

これを以下の文字列で置き換えます。

```
com.hp.nms.ui.sso.domain = mycompany.com
```

- 3 以下のコマンドを実行し、変更をコミットします。

```
nmssso.ovpl -reload
```

詳細については、[nmssso.ovpl](#) リファレンスページ、または UNIX のマンページを参照してください。



## 1つのドメインに対するSSOの有効化

1つのドメインでSSOを使用可能にするには、以下の手順を実行します。

1 以下のファイルを編集します。

- **Windows:** %NNM\_PROPS%\nms-ui.properties
- **UNIX:** \$NNM\_PROPS/nms-ui.properties

2 ファイルから、以下のようなセクションを特定します。

```
com.hp.nms.ui.sso.isEnabled = false
```

これを以下のように変更します。

```
com.hp.nms.ui.sso.isEnabled = true
```

3 ファイルから、以下のようなセクションを特定します。

```
com.hp.nms.ui.sso.domain = mycompany.com
```

**mycompany.com** を、NNMi 管理サーバーが配置されているドメインに変更します。  
1つのドメインでSSOを有効にするときは、1つのドメインのみがリストされていることを確認してください。

4 ファイルから、以下のようなセクションを特定します。

```
com.hp.nms.ui.sso.protectedDomains = mycompany.com
```

**mycompany.com** を、NNMi 管理サーバーが配置されているドメインに変更します。  
1つの保護ドメインでSSOを有効にするときは、1つの保護ドメインのみがリストされていることを確認してください。

5 以下のコマンドを実行し、変更をコミットします。

```
nnmssso.ovpl -reload
```

詳細については、**nnmssso.ovpl** リファレンスページ、またはUNIXのマニュアルを参照してください。

## 異なるドメインに配置されているNNMi管理サーバーに対するSSOの有効化

SSOを使用できるように複数のNNMi管理サーバーを設定できます。この例では、異なるドメインに配置されている3つのNNMi管理サーバーに対してSSOを設定する方法を説明します。SSOを使用できるように複数のNNMi管理サーバーを設定する必要がある場合に、これらのシステムが異なるドメインに配置されているときは、以下の手順を実行します。

1 以下のファイルを編集します。

- **Windows:** %NNM\_PROPS%\nms-ui.properties
- **UNIX:** \$NNM\_PROPS/nms-ui.properties

2 ファイルから、以下のようなセクションを特定します。

```
com.hp.nms.ui.sso.isEnabled = false
```

これを以下のように変更します。

```
com.hp.nms.ui.sso.isEnabled = true
```

- 3 ファイルから、以下のようなセクションを特定します。  
`com.hp.nms.ui.sso.domain = group1.mycompany.com`  
 ドメイン名に 1 つ以上のドット (.) があることを確認します。
- 4 ファイルから、以下のようなセクションを特定します。  
`com.hp.nms.ui.sso.protectedDomains=group1.mycompany.com`  
 これを以下のように変更します。  
`com.hp.nms.ui.sso.protectedDomains=group1.mycompany.com,  
 group2.yourcompany.com, group3.yourcompany.com`
- 5 ファイルから、以下のようなセクションを特定します。  
`com.hp.nms.ui.sso.initString = Initialization String`  
 1 つの SSO 設定で機能するように各 NNMi 管理サーバーの初期化ストリングを共有する必要があります。SSO 設定に含まれるすべての NNMi 管理サーバーの初期化ストリングを同じ値に変更します。
- 6 以下のコマンドを実行し、変更をコミットします。  
**`nnmssso.ovpl -reload`**  
 詳細については、`nnmssso.ovpl` リファレンスページ、または UNIX のマンページを参照してください。
- 7 手順 1 から手順 6 までをさらに 2 回繰り返し、残りの 2 つの NNMi 管理サーバーを設定します。残りの各 NNMi 管理サーバーについては、手順 3 で、`group2` または `group3` を `group1` に置き換えてください。

---

## NNMi と NNM iSPI の SSO アクセス

SSO が有効になったら、NNMi と NNM iSPI 間の SSO には `initString` 設定は必要ありません。

SSO を使用するには、以下のように NNMi にアクセスします。

- 以下の形式の正しい URL を使用します。  
**`<protocol>://<fully_qualified_domain_name>:<port_number>/nnm/`**  
`<protocol>` は `http` または `https` です。  
`<fully_qualified_domain_name>` は、NNMi 管理サーバーの正式な完全修飾ドメイン名 (FQDN) です。  
`<port_number>` は、NNMi コンソールに接続するためのポートです。これは、NNMi のインストール時に割り当てられ、以下のファイルで指定されます。
  - Windows: `%NnmDataDir%\%conf%\nnm\props\nms-local.properties`
  - UNIX: `$NnmDataDir/conf/nnm/props/nms-local.properties`
- 有効なアカウントを使用して NNMi にログオンします。

SSO が機能するには、NNMi と NNM iSPI への URL アクセスに共通するネットワークドメイン名が使用されている必要があります。さらに、IP アドレスが含まれていない URL である必要があります。NNMi 管理サーバー用の FQDN がない場合は、代わりに NNMi

管理サーバーの IP アドレスを使用できますが、その場合、NNMi iSPI のシングルサインオンが無効になるため、次回 NNMi iSPI にアクセスするときにもう一度ログオンする必要があります。

NNMi 管理サーバーの正式な FQDN を判別するには、以下のいずれかの方法を使用します。

- `nnmofficialfqdn.ovpl` コマンドを使用して、インストール中に設定した正式な FQDN の値を表示します。詳細については、`nnmofficialfqdn.ovpl` リファレンスページまたは UNIX のマンページを参照してください。
- NNMi コンソールで、[ヘルプ]>[システム情報] をクリックします。[サーバー] タブで、正式な FQDN ステートメントを特定します。

インストール中に設定された正式な FQDN を変更する必要がある場合は、`nnmsetofficialfqdn.ovpl` コマンドを使用します。詳細については、`nnmsetofficialfqdn.ovpl` リファレンスページまたは UNIX のマンページを参照してください。



インストール後、システムアカウントは有効なままになっています。システムアカウントは、コマンドラインのセキュリティと復旧の目的のみに使用します。

NNMi iSPI への SSO には、ユーザーが正式な FQDN を含む URL で NNMi コンソールにアクセスすることが要求されます。IP アドレスや短縮されたドメイン名など、正式ではないドメイン名を使用して NNMi コンソールにアクセスした場合に NNMi URL を正式な FQDN にリダイレクトするように NNMi を設定できます。URL をリダイレクトするように NNMi を設定する前に、該当する正式な FQDN が設定されている必要があります。詳細については、NNMi ヘルプを参照してください。

NNMi で URL へのリダイレクトを可能にした後、以下の点に注意してください。

- アクセスする NNMi 管理サーバーに適したホスト名を使用して、NNMi コンソールにログオンできます。たとえば、ユーザーが `http://localhost/nnm` を要求している場合、NNMi は `http://host.mydomain.com/nnm` などの URL にそれをリダイレクトします。
- `http://host.mydomain.com/nnm` を使用して NNMi コンソールにアクセスできない場合、以下の URL を使用して、NNMi コンソールに直接アクセスしてください。  
`<protocol>://`  
`<fully_qualified_domain_name>:<port_number>launch?cmd=showMain.`  
`<protocol>` は `http` または `https` です。  
`<fully_qualified_domain_name>` は、NNMi 管理サーバーの正式な完全修飾ドメイン名 (FQDN) です。  
`<port_number>` は、NNMi コンソールに接続するためのポートです。これは、NNMi のインストール時に割り当てられ、以下のファイルで指定されます。

- Windows: `%NnmDataDir%\conf\%nmm%\props\%nms-local.properties`
- UNIX: `$NnmDataDir/conf/nnm/props/nms-local.properties`

## SSO の無効化

SSO を無効にする必要がある場合は、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - **Windows:** %NNM\_PROPS%\nms-ui.properties
  - **UNIX:** \$NNM\_PROPS/nms-ui.properties
- 2 ファイルから、以下のようなセクションを特定します。
 

```
com.hp.nms.ui.sso.isEnabled = true
```

 これを以下のように変更します。
 

```
com.hp.nms.ui.sso.isEnabled = false
```
- 3 以下のコマンドを実行し、変更をコミットします。
 

```
nmssso.ovpl -reload
```

 詳細については、[nmssso.ovpl](#) リファレンスページ、または UNIX のマンページを参照してください。

## SSO セキュリティに関する注意

- 1 **SSO** セキュリティの機密 `initString` パラメーター。
 

SSO は、対象鍵暗号方式を使用して SSO トークンの検証と作成を行います。設定内の `initString` パラメーターは、秘密鍵の初期化に使用されます。アプリケーションはトークンを作成し、`initString` パラメーターを使用する各アプリケーションはそのトークンを検証します。

以下は、非常に重要な情報です。

  - `initString` パラメーターを設定せずに、SSO を使用することはできません。
  - `initString` パラメーターは機密情報であり、公開、移動、永続性において、機密情報として取り扱う必要があります。
  - 相互に統合するアプリケーションは、SSO を使用して `initString` を共有できます。
  - `initString` は最低 12 文字の長さです。
- 2 特に必要でない限り、SSO を無効にします。
- 3 最も弱い認証フレームワークを使用するアプリケーションやほかの統合アプリケーションに信頼される SSO トークンを発行するアプリケーションは、すべてのアプリケーションの認証セキュリティレベルを判断します。
 

HP は、強力で安全な認証フレームワークを使用するアプリケーションのみが SSO トークンを発行するように設定することを推奨します。
- 4 対象鍵暗号方式による影響について
 

SSO は、SSO トークンの発行と検証に対象鍵暗号方式を使用します。そのため、SSO を使用するアプリケーションは、同一の `initString` を共有しているその他のすべてのアプリケーションによって信頼されるトークンを発行できます。

initString を共有するアプリケーションが信頼されない場所にある、または信頼できない場所にアクセスできる場合に、この潜在的なリスクが浮上します。

## 5 ユーザーロール

**SSO** では、統合されたアプリケーション間でユーザーロールは共有されません。このため、統合されたアプリケーションはユーザーロールを監視する必要があります。**HP** は、すべての統合アプリケーションで、同一のユーザーレジストリ (**LDAP/AD** として) を共有することを推奨します。

ユーザーロールを管理できないと、セキュリティ違反やアプリケーションエラーが発生する場合があります。たとえば、統合アプリケーションで異なるロールに同じユーザー名が割り当てられることがあります。

ユーザーがアプリケーション **A** にログオンし、コンテナーやアプリケーション認証を使用するアプリケーション **B** にアクセスするとします。ユーザーロールを管理できないと、そのユーザーはアプリケーション **B** に手動でログオンし、ユーザー名を入力しなければなりません。このとき、ユーザーがアプリケーション **A** とは異なるユーザー名を入力すると、その後、アプリケーション **A** または **B** から **3** つ目のアプリケーション (アプリケーション **C**) にアクセスすると、アプリケーション **A** または **B** に使用したユーザー名でアプリケーション **C** にアクセスするという予期しない動作が発生することになります。

## 6 認証に Identity Manager が使用される

**Identity Manager** 内の保護されていないすべてのリソースは、**SSO** 設定に非セキュア URL 設定として設定されている必要があります。

## 7 SSO デモモード:

- デモの目的のみに **SSO** デモモードを使用します。
- セキュアでないネットワークでのみデモモードを使用します。
- デモモードを本番に使用しないでください。デモモードと本番モードを混ぜて使用しないでください。



# NNMi で使用する Telnet および SSH プロトコルを設 定する

[ **アクション** ] > [ **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 (NNMi コンソールを現在実行中の Web ブラウザーから)。[ **アクション** ] > [ **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 (NNMi コンソールを現在実行中の Web ブラウザーから)。デフォルトでは、Microsoft 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。telnet、SSH、または両方のプロトコルを各 NNMi ユーザーに設定して (システムごとに)、NNMi コンソールメニュー項目を変更できます。

この章には、以下のトピックがあります。


- 「Telnet または SSH メニュー項目の無効化」 (151 ページ)
- 「Windows 上のブラウザーへの Telnet または SSH クライアントの設定」 (152 ページ)
- 「Linux で Telnet または SSH を使用する Firefox の設定」 (159 ページ)
- 「Windows レジストリを変更するファイル例」 (160 ページ)

---

## Telnet または SSH メニュー項目の無効化

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 コンソールにログオ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、以下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** ] を選択します。
- 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、[ **開く** ]  をクリックします。
- 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します。

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、閉じます。

詳細については、NNMi ヘルプの「アクションメニューの制御」を参照してください。

## Windows 上のブラウザへの Telnet または SSH クライアントの設定

NNMi ユーザーの Web ブラウザー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]>[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ーおよび Web ブラウザーで実行する必要があります。

NNMi ユーザーの Web ブラウザー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]>[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ーおよび Web ブラウザーで実行する必要があります。

このセクションの手順を完了するには、コンピューター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]>[Internet Explorer のバージョン情報]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のみ使用可能です。

表 7 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 7 Windows での Telnet および SSH 設定手順のマトリクス

Web ブラウザー	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	<ul style="list-style-type: none"> <li>「Windows オペレーティングシステム提供の Telnet クライアント」(154 ページ)</li> <li>「サードパーティ Telnet クライアント (標準 Windows)」(155 ページ)</li> <li>「サードパーティ SSH クライアント (標準 Windows および Windows 上のウィンドウ)」(157 ページ)</li> </ul>
	64 ビット Windows 7	<ul style="list-style-type: none"> <li>「サードパーティ Telnet クライアント (標準 Windows)」(155 ページ)</li> <li>「サードパーティ SSH クライアント (標準 Windows および Windows 上のウィンドウ)」(157 ページ)</li> </ul>
	64 ビット Windows 7 以外	<ul style="list-style-type: none"> <li>「サードパーティ Telnet クライアント (Windows 上のウィンドウ)」(156 ページ)</li> <li>「サードパーティ SSH クライアント (標準 Windows および Windows 上のウィンドウ)」(157 ページ)</li> </ul>



表7 WindowsでのTelnetおよびSSH設定手順のマトリクス(続き)

Web ブラウザー	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 64ビット	64ビット	<ul style="list-style-type: none"> <li>「Windows オペレーティングシステム提供の Telnet クライアント」(154 ページ)</li> <li>「サードパーティ Telnet クライアント(標準 Windows)」(155 ページ)</li> <li>「サードパーティ SSH クライアント(標準 Windows および Windows 上のウィンドウ)」(157 ページ)</li> </ul>
Firefox	32ビット	<ul style="list-style-type: none"> <li>「Windows オペレーティングシステム提供の Telnet クライアント」(154 ページ)</li> <li>「サードパーティ Telnet クライアント(標準 Windows)」(155 ページ)</li> <li>「サードパーティ SSH クライアント(標準 Windows および Windows 上のウィンドウ)」(157 ページ)</li> </ul>
	64ビット Windows 7	<ul style="list-style-type: none"> <li>「サードパーティ Telnet クライアント(標準 Windows)」(155 ページ)</li> <li>「サードパーティ SSH クライアント(標準 Windows および Windows 上のウィンドウ)」(157 ページ)</li> </ul>
	64ビット Windows 7 以外	<ul style="list-style-type: none"> <li>「サードパーティ Telnet クライアント(Windows 上のウィンドウ)」(156 ページ)</li> <li>「サードパーティ SSH クライアント(標準 Windows および Windows 上のウィンドウ)」(157 ページ)</li> </ul>



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、「Windows レジストリを変更するファイル例」(160 ページ)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、以下の 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>

## 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、以下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



Windows オペレーティングシステムで提供される telnet クライアントは 64 ビット Windows オペレーティングシステムで実行される Internet Explorer の 32 ビットバージョンでは動作しません。これを解決するには、64 ビットバージョンの Internet Explorer を使用します。Windows 64 ビットオペレーティングシステムには、Internet Explorer の 32 ビットバージョンおよび 64 ビットバージョンの両方が含まれています。次のディレクトリでこれらの Internet Explorer バージョンを検索します。

- 64 ビットバージョン : %ProgramFiles%/Internet Explorer
- 32 ビットバージョン : %ProgramFiles(x86)%/Internet Explorer

Web ブラウザー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、以下の手順を実行します。

- 1 (Microsoft Windows 7、Microsoft Vista、または Microsoft Windows Server 2008 専用) オペレーティングシステムに該当する手順に従い、コンピューター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します。

Windows 7 または 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:

- 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 を有効化します。

- 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ーを使用して、[HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Internet Explorer\MAIN\FeatureControl\FEATURE\_DISABLE\_TELNET\_PROTOCOL] キーに以下の値を追加します。

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付けを設定します。

- 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ーを使用して、[HKEY\_CLASSES\_ROOT¥telnet¥shell¥open¥command] キーを以下の値で変更します。

名前	タイプ	データ
(デフォルト)	REG_SZ	rundll32.exe url.dll,TelnetProtocolHandler %l

- 4 %1 (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。次に例を示します。

```
"C:¥Windows¥system32¥rundll32.exe"  
"C:¥Windows¥system32¥url.dll",TelnetProtocolHandler %1
```

- 5 Web ブラウザーを再起動してから、ブラウザーのアドレスバーに telnet コマンドを入力します。

```
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 リンクを同様に処理する] チェックボックスをオンにします。

## サードパーティ Telnet クライアント (標準 Windows)

この手順は、以下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 ブラウザー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、以下の手順に従います。

- 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します。

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あげます。PuTTY クライアントは <http://www.putty.org> から使用できます。

- 2 (Internet Explorer 専用) telnet を使用する Internet Explorer を有効化します。

- 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ーを使用して、[HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Microsoft¥Internet Explorer¥MAIN¥FeatureControl¥FEATURE\_DISABLE\_TELNET\_PROTOCOL] キーに以下の値を追加します。

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付けを設定します。
  - 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ーを使用して、[HKEY\_CLASSES\_ROOT¥telnet¥shell¥open¥command] キーを以下の値で変更します。

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 (") とバックスラッシュ (¥) 文字はバックスラッシュ (¥) 文字でエスケープします。

- 4 Web ブラウザーを再起動してから、ブラウザーのアドレスバーに telnet コマンドを入力します。

```
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 リンクを同様に処理する] チェックボックスをオンにします。

## サードパーティ Telnet クライアント (Windows 上のウィンドウ)

この手順は、以下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)
- 32 ビットオペレーティングシステム上の 64 ビット Firefox

Web ブラウザー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、以下の手順に従います。

- 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します。

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あげます。PuTTY クライアントは <http://www.putty.org> から使用できます。

- 2 (Internet Explorer 専用) telnet を使用する Internet Explorer を有効化します。

- 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ーを使用して、[HKEY\_LOCAL\_MACHINE¥SOFTWARE¥Wow6432Node¥Microsoft¥Internet Explorer¥MAIN¥FeatureControl¥FEATURE\_DISABLE\_TELNET\_PROTOCOL] キーに以下の値を追加します。

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付けを設定します。

- 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ーを使用して、[HKEY\_CLASSES\_ROOT¥Wow6432Node¥telnet¥shell¥open¥command] キーを以下の値で変更します。

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 (") とバックスラッシュ (\) 文字はバックスラッシュ (\) 文字でエスケープします。

4 Web ブラウザーを再起動してから、ブラウザーのアドレスバーに telnet コマンドを入力します。

```
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 リンクを同様に処理する] チェックボックスをオンにします。

## サードパーティ SSH クライアント (標準 Windows および Windows 上のウィンドウ)

この手順は、以下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 ブラウザー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、以下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します。

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あげます。PuTTY クライアントは <http://www.putty.org> から使用できます。

2 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、以下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。

- 3 ssh プロトコルを定義します。
  - 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ーを使用して、[HKEY\_CLASSES\_ROOT¥ssh] キーに以下の値を追加します。

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	セキュアーシェル
URL プロトコル	REG_SZ	値なし

- 4 URL:ssh プロトコルファイルタイプのファイル関連付けを設定します。
  - 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ーを使用して、[HKEY\_CLASSES\_ROOT¥ssh¥shell¥open¥command] キーを以下の値で変更します。

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 (") とバックスラッシュ (\) 文字はバックスラッシュ (\) 文字でエスケープします。

- 5 Web ブラウザーを再起動してから、ブラウザーのアドレスバーに ssh コマンドを入力します。

**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 リンクを同様に処理する] チェックボックスをオンにします。

## Linux で Telnet または SSH を使用する Firefox の設定

**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、コンピューターの管理権限が必要です。

詳細については、[http://kb.mozillazine.org/Register\\_protocol](http://kb.mozillazine.org/Register_protocol) を参照してください。


### Linux 上の Telnet

**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、以下の手順に従います。

- 1 **telnet** プロトコルを定義します。
  - a `/usr/local/bin/nmmtelnet` ファイルを以下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/nmmtelnet
```
- 2 **telnet** 用の **Firefox** プリファレンスを設定します。
  - 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** を設定します。
  - 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]** > **[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/nmmtelnet` を選択します。
  - c **[今後 telnet リンクを同様に処理する]** チェックボックスをオンにします。

## Linux 上のセキュアシェル

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、以下の手順に従います。

- 1 ssh プロトコルを定義します。
  - a /usr/local/bin/nmssh ファイルを以下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/nmssh
```
- 2 SSH 用の Firefox プリファレンスを設定します。
  - 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、**[プール値]** 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 を設定します。
  - 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/nmssh を選択します。
  - c **[今後 ssh リンクを同様に処理する]** チェックボックスをオンにします。

---

## 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 ファイルで自動化することができます。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。



## nnmtelnet.reg の例

このレジストリの内容例は、「[Windows オペレーティングシステム提供の Telnet クライアント](#)」(154 ページ)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 %1"
```

## nnmputtytelnet.reg の例

このレジストリの内容例は、「[サードパーティ Telnet クライアント \(標準 Windows\)](#)」(155 ページ)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:0c000000
```

```
[HKEY_CLASSES_ROOT\telnet\shell\open\command]
@="%%C:%%Program Files%%PuTTY%%putty.exe" %1"
```

## nnmtelnet32on64.reg の例

このレジストリの内容例は、「[サードパーティ Telnet クライアント \(Windows 上のウィンドウ\)](#)」(156 ページ)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" %1"
```

## nnmssh.reg の例

このレジストリの内容例は、「[サードパーティ SSH クライアント \(標準 Windows および Windows 上のウィンドウ\)](#)」(157 ページ)に適用されます。

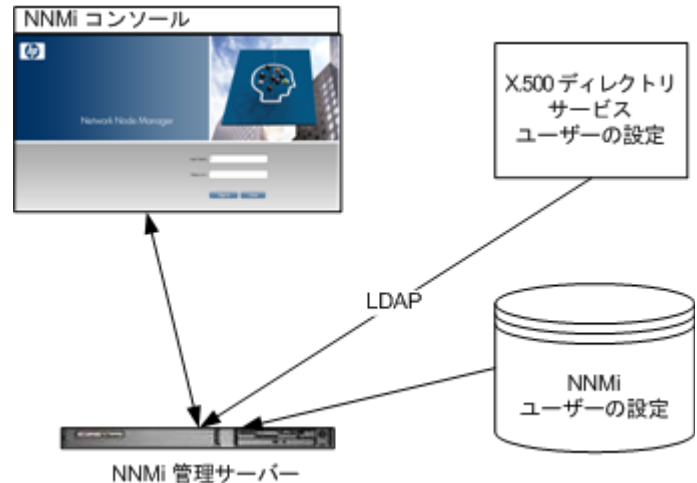
Windows Registry Editor Version 5.00

```
[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" %1"
```



# 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により、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存場所を統合する方法について説明します。内容は以下のとおりです。

- 「NNMi ユーザーのアクセス情報と設定オプション」(163 ページ)
- 「ディレクトリサービスにアクセスする NNMi の設定」(167 ページ)
- 「ディレクトリサービスのアクセス設定を変更し、NNMi のセキュリティモデルをサポートする」(175 ページ)
- 「ディレクトリサービスのクエリー」(178 ページ)
- 「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」(188 ページ)
- 「ディレクトリサービス統合のトラブルシューティング」(189 ページ)
- 「ldap.properties 設定ファイルリファレンス」(190 ページ)

## NNMi ユーザーのアクセス情報と設定オプション

NNMi ユーザーは、以下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、ユーザー名と関連付けられ、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**により、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。ユーザーグループの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、以下の説明にあるように、NNMi ユーザーアクセス情報の保存先としていくつかのオプションが用意されています。表 8 に、NNMi ユーザーアクセス情報を保存するデータベースを設定オプションごとに示します。

表 8 ユーザー情報の保存オプション

項目	ユーザー名	パスワード	ユーザーグループ	ユーザーグループメンバーシップ
1	NNMi	NNMi	NNMi	NNMi
2	両方	ディレクトリサービス	NNMi	NNMi
3	ディレクトリサービス	ディレクトリサービス	両方	ディレクトリサービス

NNMi を、ユーザーアクセス情報の一部またはすべてを保存するディレクトリサービスと統合すると、[システム情報] ウィンドウの[サーバー] タブのユーザーアカウントおよびユーザーグループ定義ステートメントに、LDAP クエリーによって取得した情報のタイプが示されます。

NNMi と他のアプリケーションの間のシングルサインオン (SSO) は、NNMi ユーザーアクセス情報の設定やその保存場所に関係なく機能します。

## オプション 1: NNMi データベースにすべての NNMi ユーザー情報を保存

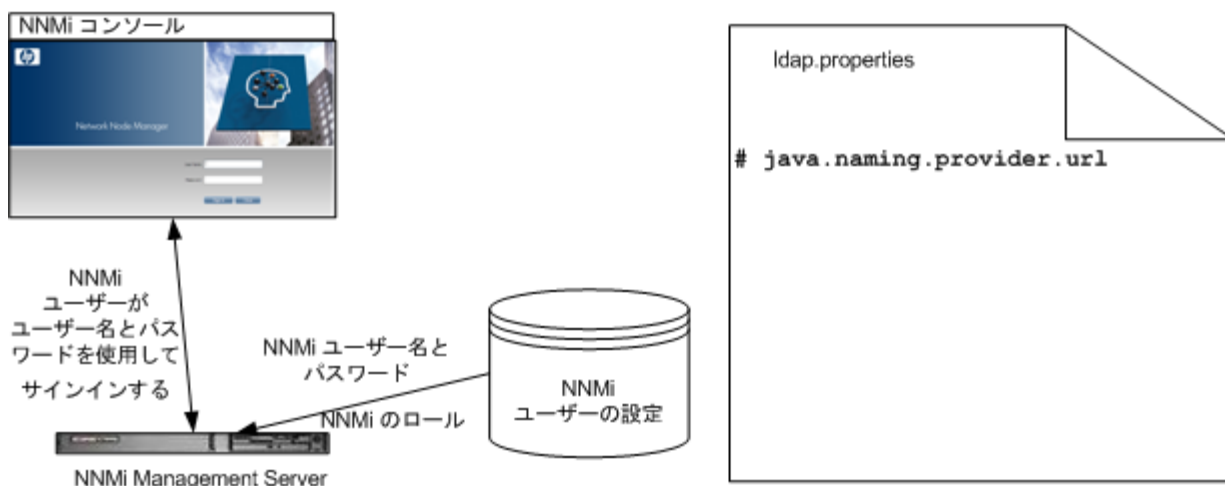
設定オプション 1 では、NNMi が、すべてのユーザーアクセス情報を取得するために NNMi データベースにアクセスします。それらの情報は、NNMi 管理者が NNMi コンソールで定義およびメンテナンスします。ユーザーアクセス情報は、NNMi にとってローカルの情報となります。NNMi はディレクトリサービスにアクセスせず、NNMi は (図 9 のコメント行に示されている) ldap.properties ファイルを無視します。

図 9 に、このオプションの情報フローを示します。この情報フローは、以下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 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 9 オプション 1 における NNMi ユーザーサインインの情報フロー



## オプション 2: 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

設定オプション 2 では、NNMi が、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、他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、以下で説明するように共同で行われます。

- ディレクトリサービス管理者は、ディレクトリサービス内のユーザー名とパスワードをメンテナンスします。
- NNMi 管理者は、(ディレクトリサービスで定義されている)ユーザー名、ユーザーグループ定義、ユーザーグループのマッピングを NNMi コンソールで入力します。
- NNMi 管理者は、NNMi に対するユーザー名のディレクトリサービスデータベーススキーマを記述する NNMi ldap.properties ファイルを設定します (図 10 のコマンドラインは、NNMi が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)。

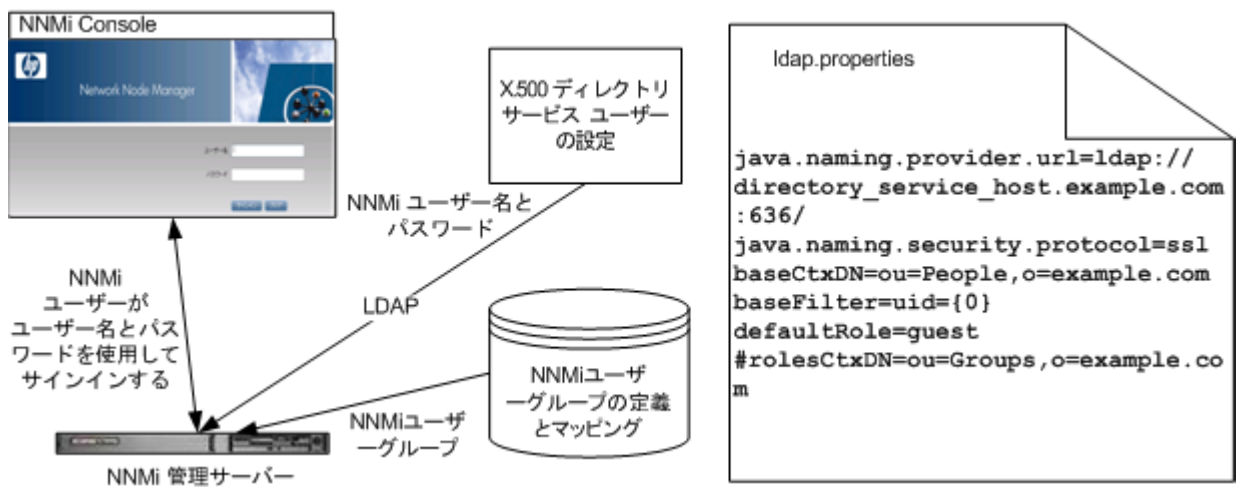
ユーザー名は、2 か所で入力する必要があるため、両方の場所でユーザー名のメンテナンスを行う必要があります。

図 10 に、このオプションの情報フローを示します。この情報フローは、以下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。

図 10 オプション 2 における NNMi ユーザーサインインの情報フロー



## 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保存

設定オプション 3 では、NNMi が、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、他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により、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、以下で説明するように共同で行われます。

- ディレクトリサービス管理者は、ディレクトリサービス内のユーザー名、パスワード、グループメンバーシップをメンテナンスします。
- NNMi 管理者は、ディレクトリサービスグループを NNMi ユーザーグループに NNMi コンソールでマッピングします。
- NNMi 管理者は、NNMi に対するユーザー名およびグループのディレクトリサービスデータベーススキーマを記述する NNMi ldap.properties ファイルを設定します

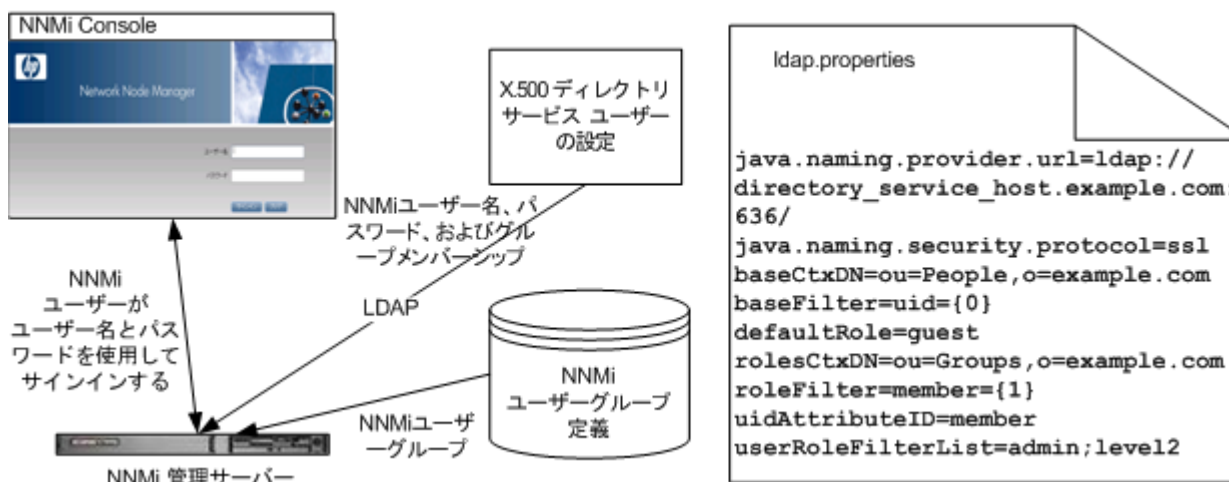
図 11 に、このオプションでの情報フローを示します。これは、NNMi にアクセスする必要があるユーザーで構成されるユーザーグループを含めるようにディレクトリサービスを変更することが可能な環境に適しています。

このオプションはオプション 2 の例を拡張した形態であるため、HP では以下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。

図 11 オプション 3 における NNMi ユーザーサインインの情報フロー



## ディレクトリサービスにアクセスする NNMi の設定

ディレクトリサービスへのアクセスは、以下のフィ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、「[ldap.properties 設定ファイルリファレンス](#)」(190 ページ)を参照してください。「[例](#)」(195 ページ)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、「[ディレクトリサービスのクエリー](#)」(178 ページ)を参照してください。

設定オプション 2 の場合は、以下のタスクを実行します。

- **タスク 1:** 現在の NNMi ユーザー情報をバックアップする
- **タスク 2:** オプション。ディレクトリサービスへのセキュア接続を設定する
- **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
- **タスク 4:** ユーザー名とパスワードの設定をテストする
- **タスク 9:** クリーンアップして NNMi への予期せぬアクセスを防止する
- **タスク 10:** オプション。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

設定オプション 3 の場合は、以下のタスクを実行します。

- **タスク 1:** 現在の NNMi ユーザー情報をバックアップする
- **タスク 2:** オプション。ディレクトリサービスへのセキュア接続を設定する
- **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
- **タスク 4:** ユーザー名とパスワードの設定をテストする
- **タスク 5:** (設定オプション 3 のみ)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



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、「[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](#)」(188 ページ)を参照してください。

- **タスク 6:** (設定オプション 3 のみ)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
- **タスク 7:** (設定オプション 3 のみ) NNMi ユーザーグループ設定をテストする
- **タスク 8:** (設定オプション 3 のみ) 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
- **タスク 9:** クリーンアップして NNMi への予期せぬアクセスを防止する
- **タスク 10:** オプション。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

### 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
```

## タスク 2: オプション。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、「ディレクトリサービスへの **SSL 接続を設定する**」(140 ページ) 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。

## 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、設定オプション **2** および **3** 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、以下のセクションが含まれます。

- **Microsoft Active Directory** の場合の簡単な方法
- 他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、「ユーザー識別」(183 ページ) を参照してください。)

### 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ーで開きます。
- 2 ファイルの内容を以下のテキストで上書きします。

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

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

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

defaultRole=guest

#rolesCtxDN=CN=Users,DC=<myhostname>,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** を指定します。上のテキストには以下の行があります。

```
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します。上のテキストには以下の行があります。

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

以下のように置き換えます。

- <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。  
平文のパスワードを保存する場合は、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。  
暗号化されたパスワードを指定する場合は、ldap.properties ファイルに保存する前に平文のパスワードを以下のコマンドで暗号化します。  
**nnmldap.ovpl -encrypt <mypassword>**



この暗号化パスワードは、その作成先の **NNMi** インスタンスでのみ機能します。他の **NNMi** インスタンスには使用しないでください。

詳細については、**nnmldap.ovpl** リファレンスページ、または **UNIX** のマンページを参照してください。

- 5 ディレクトリサーバ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します。上のテキストには以下の行があります。

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

<myhostname>、<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 (たとえばホスト名 myserver.example.com の場合は、DC=myserver,DC=example,DC=com と指定します)。

#### 他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ーで開きます。
- 2 ディレクトリサービスにアクセスするときの **URL** を指定します。上のテキストには以下の行があります。

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

以下を実行します。

- 行のコメントを解除します (# 文字を削除します)。
- <myldapservice> 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します。上のテキストには以下の行があります。

```
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します。上のテキストには以下の行があります。

```
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。

#### 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 と設定します (この値はいつでも変更できます)。
- 2 ldap.properties ファイルを保存します。
- 3 以下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ます。

```
nnmlldap.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 コンソールにログオンします。



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) を確認します。
  - 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。



各テストの後で、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 以下のコマンドを実行し、あるユーザーの設定をテストします。

```
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。推奨事項は以下のとおりです。

- 168 ページの **タスク 3** が正常に完了したことを確認します。
  - 「ユーザー識別」(183 ページ) 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** を繰り返します。
- 8 ログオンできたら、設定方法を選択します。
- 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 (設定オプション 2) 場合は、174 ページの **タスク 9** 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 (設定オプション 3) 場合は、次は **タスク 5** に進みます。

#### タスク 5:(設定オプション 3 のみ) 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、設定オプション 3 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、以下のセクションが含まれます。

- **Microsoft Active Directory** の場合の簡単な方法
- 他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、「ユーザーグループの識別」(186 ページ) を参照してください。)

##### 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ーで開きます。
- 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します。上のテキストには以下の行があります。

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

以下を実行します。

- 行のコメントを解除します (# 文字を削除します)。
- <myhostname>、<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 (たとえばホスト名 myserver.example.com の場合は、DC=myserver,DC=example,DC=com と指定します)。

## 他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ーで開きます。
- 2 ディレクトリサーバ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。上のテキストには以下の行があります。

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


以下を実行します。

- 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します。上のテキストには以下の行があります。

```
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。

## タスク 6:(設定オプション 3 のみ) 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します。
  - 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り、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いず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他のグループに、NNMi コンソールで新しいユーザーグループを作成します。
  - 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り、NNMi コンソールのトポ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。

### タスク 7:(設定オプション 3 のみ) NNMi ユーザーグループ設定をテストする

- 1 ldap.properties ファイルを保存します。
- 2 以下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ます。

```
nnmlldap.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 コンソールにログオンします。



NNMi データベースでまだ定義されておらず、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)を NNMi コンソールのタイトルバーで確認します。
  - ユーザーサインインが正しく動作したら、174 ページのタスク 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。



各テストの後で、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 以下のコマンドを実行し、あるユーザーの設定をテストします。


```
nnmlldap.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。推奨事項は以下のとおりです。

- 171 ページのタスク 5 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、172 ページのタスク 6 が正常に完了したことを確認します。
  - 「ユーザーグループの識別」(186 ページ)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 を繰り返します。

## タスク 8:(設定オプション 3 のみ) 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ーで開きます。
- 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 オペレーターが指定するように、userRoleFilterList パラメーター値を変更します。
- 
 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 (186 ページの表 11 で定義) をセミコロンで区切ったリストという形式です。
- 3 ldap.properties ファイルを保存します。
- 4 以下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ます。
 

```
nnmlldap.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 コンソールにログオンします。
- 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]** をクリックします。userRoleFilterList パラメーターによって指定されている各 NNMi ロールのユーザーに、インシデントを割り当てることができることを確認します。
- 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 の操作を繰り返してください。

## タスク 9:クリーンアップして NNMi への予期せぬアクセスを防止する

- 1 オプション。ldap.properties ファイルで、defaultRole パラメーターの値を変更するか、またはコメントを解除します。
- 2 (設定オプション 2 のみ) 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、以下の手順を実行して、NNMi データベースのユーザーアクセス情報をリセットします。
  - 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マッピングタスク」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付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 (設定オプション 3 のみ) ディレトリサービスのユーザーグループメンバーシップを使用するには、以下の手順を実行して、NNMi データベースのユーザーアクセス情報をリセットします。
  - 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付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。

#### タスク 10:オプション。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。

---

## ディレトリサービスのアクセス設定を変更し、NNMi のセキュリティモデルをサポートする

ここでは、`ldap.properties` ファイルを NNMi 8.1x または 9.0x 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。この改訂は、以下の条件の両方で必要となります。

- `ldap.properties` ファイルにより、NNMi ユーザーアクセス設定オプション 3 (ディレトリサービスにすべての NNMi ユーザー情報) が現在有効になっている。
- 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る。

NNMi 8.1x および 9.0x の場合、NNMi ユーザーは、定義済みの NNMi ロールのうちいず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。

NNMi 9.10 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せん。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ます。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、以下のように 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 トポロジオブジェクトのサブセットにアクセスできます。

NNMi 8.1x および 9.0x 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 設定ファイルの以下のパラメーターで、このグループ属性を指定していました。

- roleAttributeID
- roleAttributeIsDN
- roleNameAttributeID



NNMi 9.10 では、このパラメーターが廃止されます。今後のリリースではサポートされなくなります。

NNMi 9.10 では、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における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、以下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します。

```
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ーで開きます。



ldap.properties ファイルの詳細については、「[ldap.properties 設定ファイルリファレンス](#)」(190 ページ)を参照してください。廃止されたパラメーターの詳細については、前バージョンの NNMi の『[NNMi デプロイメントリファレンス](#)』を参照してください。



3 以下のパラメーターが存在する場合は、コメントアウトするか削除します。

- roleAttributeID
- roleAttributeIsDN
- roleNameAttributeID



roleAttributeID パラメーターは、NNMi ユーザーグループの識別にどの方法を使用するかを NNMi に指示するフラグです。roleAttributeID を設定すると、NNMi では NNMi 8.1x および 9.0x の方法が使用されます。roleAttributeID を設定しないと、NNMi では NNMi 9.10 の方法が使用されます。


4 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します。

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り、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いず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。

5 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します。

6 手順 5 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します。

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。



このマッピングにより、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。

- 7 オプション。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ます。  
 詳細については、NNMi ヘルプの「セキュリティの設定」を参照してください。

## 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。

この項では以下の内容について説明します。

- ディレクトリサービスアクセス
- ディレクトリサービスの情報
- ディレクトリサービス管理者が所有する情報
- ユーザー識別
- ユーザーグループの識別

### ディレクトリサービスアクセス

LDAP は、以下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ーの完全修飾名です。
- <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、「ディレクトリサービスの情報」と、以下のサイトにある RFC 1959 「An LDAP URL Format」を参照してください。

**labs.apache.org/webarch/uri/rfc/rfc1959.txt**

Web ブラウザーで 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 (たとえば、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 ブラウザーからの 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 ブラウザー (Apache Directory Studio に含まれる LDAP ブラウザーなど) を使用し、設定パラメーターの有効性を検証できます。

### 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。

以下の例は、**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限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。以下の情報は、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。
- 他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** ファイルにより、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以下の項目です。

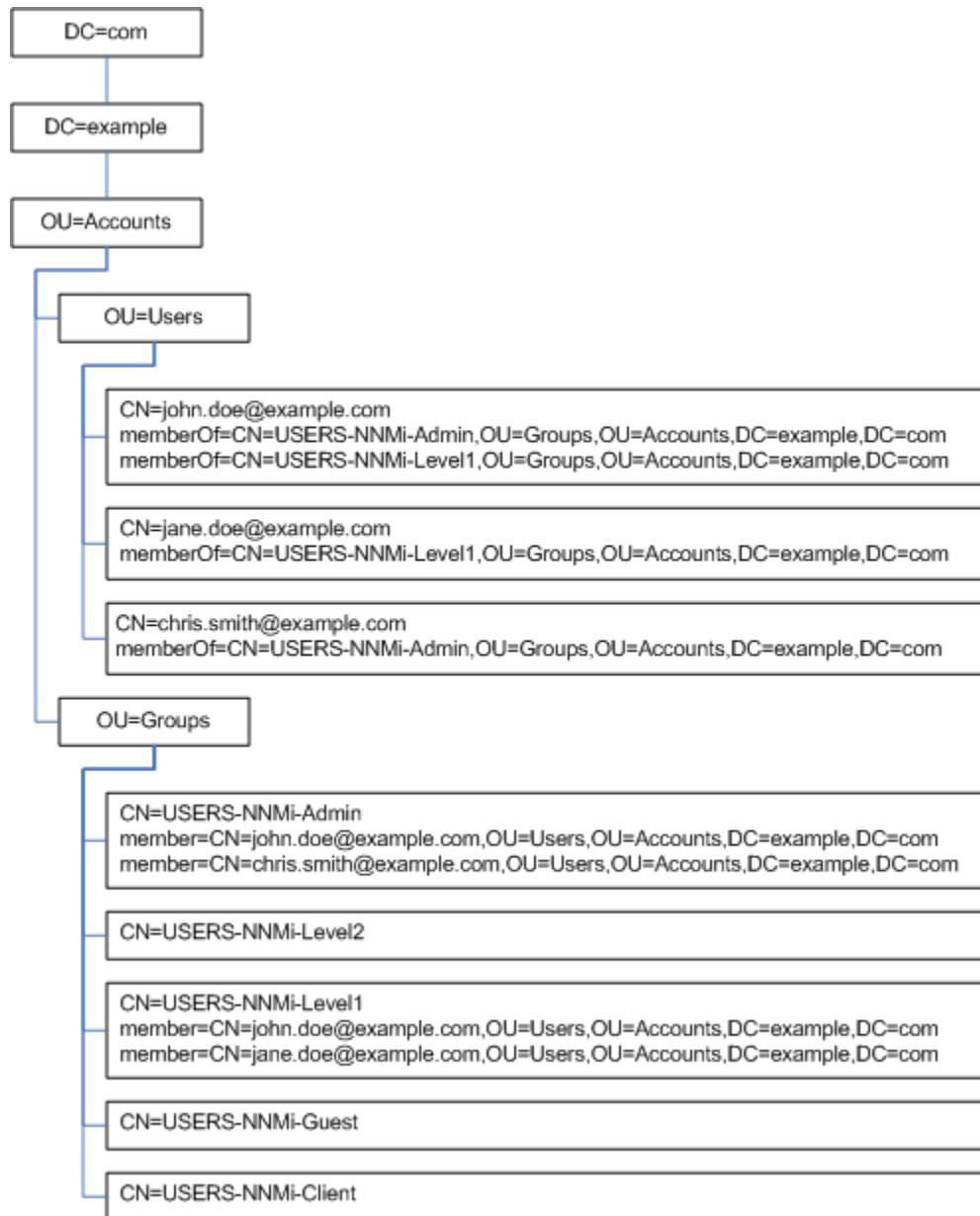
- ユーザー **John Doe** の識別名：  
CN=john.doe@example.com,OU=Users,OU=Accounts,DC=example,DC=com
- **USERS-NNMi-Admin** グループの識別名：  
CN=USERS-NNMi-Admin,OU=Groups,OU=Accounts,DC=example,DC=com
- ディレクトリサービスユーザー **ID** を保存するグループ属性：**member**

**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
```

180 ページの [図 12](#) に、このディレクトリサービスドメインを例示します。

図 12 Active Directory のドメイン例



他のディレクトリサービスの情報構造例

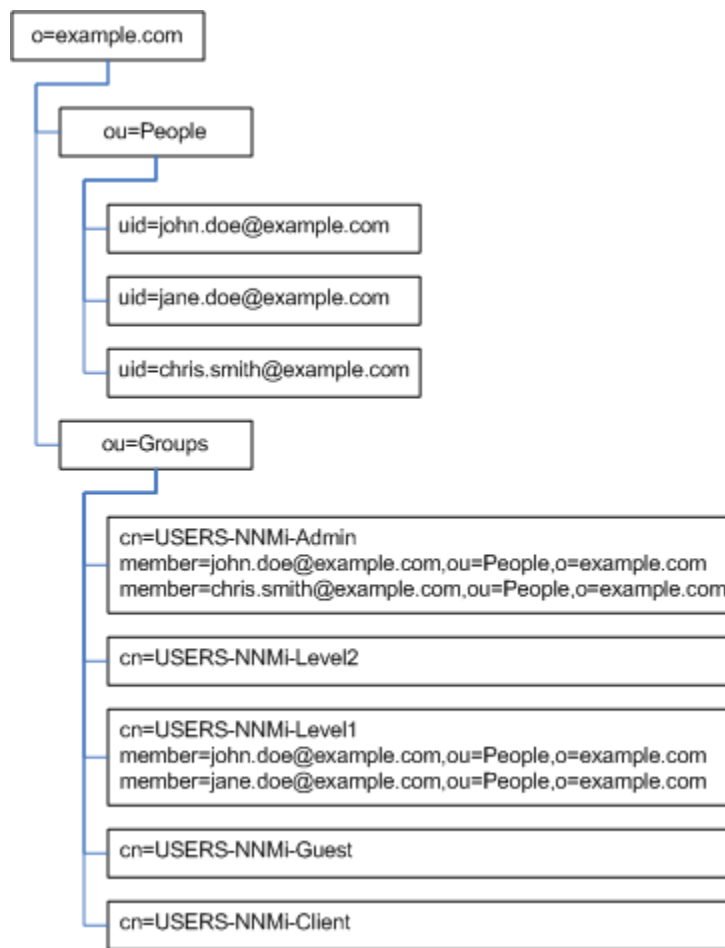
この例での関心の対象は以下の項目です。

- 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 13 他のディレクトリサービスのドメインの例



## ディレクトリサービス管理者が所有する情報

表 9 と表 10 には、ディレクトリサービスに LDAP 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取得する情報をリストしています。

- ユーザー名とパスワードのみにディレクトリサービスを使用する場合は (設定オプション 2)、表 9 の情報を収集します。
- すべての NNMi アクセス情報にディレクトリサービスを使用する場合は (設定オプション 3)、表 9 と表 10 の情報を収集します。

表 9 ユーザー名およびパスワードをディレクトリサービスから取得するための情報

情報	Active Directory の例	その他のディレクトリサービスの例
ディレクトリサービスをホストするコンピューターの完全修飾名	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	<ul style="list-style-type: none"> <li>• 非 SSL 接続の場合は 389</li> <li>• SSL 接続の場合は 636</li> </ul>	
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、「ディレクトリサービスへの SSL 接続を設定する」(140 ページ)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 グループメンバーシップをディレクトリサービスから取得するための情報

情報	Active Directory の例	その他のディレクトリサービスの例
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 ユーザー属性によりグループを識別します。	<ul style="list-style-type: none"> <li>• ou=Groups,o=example.com</li> <li>• cn=USERS-NNMi-*, ou=Groups,o=example.com</li> </ul>
グループ内のユーザーを識別する方法	<ul style="list-style-type: none"> <li>• CN=john.doe@example.com, OU=Users,OU=Accounts, DC=example,DC=com</li> <li>• CN=john.doe@example.com</li> </ul>	<ul style="list-style-type: none"> <li>• cn=john.doe@example.com, ou=People,o=example.com</li> <li>• cn=john.doe@example.com</li> </ul>
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

表 10 グループメンバーシップをディレクトリサービスから取得するための情報 (続き)

情報	Active Directory の例	その他のディレクトリサービスの例
NNMi アクセスに適用するディレクトリサービスのグループの名前	<ul style="list-style-type: none"> <li>• CN=USERS-NNMi-Admin, OU=Groups, OU=Accounts, DC=example, DC=com</li> <li>• CN=USERS-NNMi-Level2, OU=Groups, OU=Accounts, DC=example, DC=com</li> <li>• CN=USERS-NNMi-Level1, OU=Groups, OU=Accounts, DC=example, DC=com</li> <li>• CN=USERS-NNMi-Client, OU=Groups, OU=Accounts, DC=example, DC=com</li> <li>• CN=USERS-NNMi-Guest, OU=Groups, OU=Accounts, DC=example, DC=com</li> </ul>	<ul style="list-style-type: none"> <li>• cn=USERS-NNMi-Admin, ou=Groups, o=example.com</li> <li>• cn=USERS-NNMi-Level2, ou=Groups, o=example.com</li> <li>• cn=USERS-NNMi-Level1, ou=Groups, o=example.com</li> <li>• cn=USERS-NNMi-Client, ou=Groups, o=example.com</li> <li>• cn=USERS-NNMi-Guest, ou=Groups, o=example.com</li> </ul>

## ユーザー識別

ユーザー識別は、設定オプション 2 および 3 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。

設定オプション 2 の場合は、以下の情報が適用されます。

- NNMi コンソールアクセスの場合、NNMi は以下の情報を検討し、可能な限り高い権限をユーザーに与えます。
  - ldap.properties ファイルの defaultRole パラメーターの値
  - NNMi コンソールで定義済みの NNMi ユーザーグループにおける、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。

設定オプション 3 の場合は、以下の情報が適用されます。

- NNMi コンソールアクセスの場合、NNMi は以下の情報を検討し、可能な限り高い権限をユーザーに与えます。
  - ldap.properties ファイルの defaultRole パラメーターの値
  - NNMi コンソールで定義済みの NNMi ユーザーグループにマッピングされている ([ ディレクトリサービス名 ] フィールド) ディレクトリサービスグループにおける、このユーザーのメンバーシップ

- NNMi トポ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 を CN={0} に設定し、baseCtxDN を OU=Users,OU=Accounts,DC=example,DC=com に設定し、ユーザーが john.doe として NNMi にサインインする場合、ディレクトリサービスに渡される文字列は以下のとおりです。

```
CN=john.doe,OU=Users,OU=Accounts,DC=example,DC=com
```

#### その他のディレクトリ サービスでのユー ザー識別例

baseFilter を uid={0}@example.com に設定し、baseCtxDN を ou=People,o=example.com に設定し、ユーザーが john.doe として NNMi にサインインする場合、ディレクトリサービスに渡される文字列は以下のとおりです。

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

### 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)

168 ページのタスク 3 で説明した単純な方法が正しく動作しない場合は、以下の手順を実行します。

- 1 182 ページの表 9 にリストされている情報をディレクトリサービス管理者から取得します。
- 2 適切な手順を完了し、ディレクトリサービスにおけるユーザー名の形式を確認します。
  - Active Directory およびその他のディレクトリサービスの場合に LDAP ブラウザーを使用する方法: 「ディレクトリサービスでユーザーを識別する方法の判別 (LDAP ブラウザーを使用する方法)」 (185 ページ) を参照してください。
  - 他のディレクトリサービスの場合に Web ブラウザーを使用する方法: 「ディレクトリサービスでユーザーを識別する方法の判別 (Web ブラウザーを使用する方法)」 (185 ページ) を参照してください。
- 3 任意のテキストエディターで ldap.properties ファイルを開きます。



ldap.properties ファイルの詳細については、「ldap.properties 設定ファイルリファレンス」 (190 ページ) を参照してください。

- 4 java.naming.provider.url パラメーターを、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 に設定します。
  - LDAP ブラウザーを使用する方法: LDAP ブラウザー設定からこの情報を入手します。
  - Web ブラウザーを使用する方法: 「ディレクトリサービスでユーザーを識別する方法の判別 (Web ブラウザーを使用する方法)」 (185 ページ) から、<directory\_service\_host> と <port> 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、以下の行のコメントを解除 (または追加) します。

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



- 6 (Active Directory のみ) bindDN および bindCredential パラメーターを以下のよう  
に設定します。
- <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。  
平文のパスワードを保存する場合は、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。  
暗号化されたパスワードを指定する場合は、ldap.properties ファイルに保存  
する前に平文のパスワードを以下のコマンドで暗号化します。  
**nnldap.ovpl -encrypt <mypassword>**
-  この暗号化パスワードは、その作成先の NNMi インスタンスでのみ機能します。  
他の NNMi インスタンスには使用しないでください。
- 詳細については、nnldap.ovpl リファレンスページ、または UNIX のマンペー  
ジを参照してください。
- 7 baseCtxDN パラメーターを、複数のユーザーで同じになっている、識別ユーザー名の  
エレメントに設定します。
- 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時の方法と関連するように、baseFilter パラメーターを設定  
します。
- 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。
- 9 170 ページのタスク 4 の説明に従って設定をテストします。

#### 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 ブラウザーを使用する方法)

サードパーティの LDAP ブラウザーで、以下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し  
ます。
- 2 ユーザーのグループを識別し、そのグループに関連付けられているユーザーの識別名  
の形式を調べます。

#### 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 ブラウザーを使用する方法)

- 1 サポートされる Web ブラウザーで、以下の URL を入力します。

**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  
ターの完全修飾名です。
  - <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します。
    - 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ー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となります。<user\_search\_string> の値は、識別ユーザー名です。

## ユーザーグループの識別

ユーザーグループ識別は、設定オプション 3 に適用されます。

NNMi は、NNMi ユーザーのユーザーグループを以下のように判断します。

- 1 NNMi は、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します。
- 2 ユーザーグループが一致する場合、NNMi は、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します。

NNMi コンソールで、短いテキスト文字列により、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 設定ファイルの defaultRole および userRoleFilterList パラメーターも、このテキスト文字列を必要とします。表 11 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 11 NNMi ユーザーグループ名のマッピング

NNMi コンソールの NNMi のロール名 NNMi コンソール	NNMi 設定ファイルのユーザーグループの一意の名前およびテキスト文字列
管理者	admin
グローバルオペレーター	globalops
オペレーターレベル 2	level2
オペレーターレベル 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ーユーザーグループ (globalops) では、すべてのトポロジオブジェクトのみ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、ユーザーを他のいず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。

## 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 )

171 ページの **タスク 5** で説明した簡単な方法が正しく動作しない場合は、以下の手順を実行します。

- 182 ページの **表 10** にリストされている情報をディレクトリサービス管理者から取得します。
- 適切な手順を完了し、ディレクトリサービスにおけるグループ名およびグループメンバーの形式を確認します。
  - **Active Directory** の場合に **LDAP** ブラウザーを使用する方法: 「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** ブラウザーを使用する方法) 」 (187 ページ) を参照してください。
  - 他のディレクトリサービスの場合に **LDAP** ブラウザーを使用する方法: 「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 (他のディレクトリサービスの場合に **LDAP** ブラウザーを使用する方法) 」 (188 ページ) を参照してください。
  - 他のディレクトリサービスの場合に **Web** ブラウザーを使用する方法: 「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** ブラウザーを使用する方法) 」 (188 ページ) を参照してください。
- 任意のテキストエディターで `ldap.properties` ファイルを開きます。



`ldap.properties` ファイルの詳細については、「**ldap.properties** 設定ファイルリファレンス」 (190 ページ) を参照してください。

- `rolesCtxDN` パラメーターを、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します。
- 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関連するように、`roleFilter` パラメーターを設定します。実際のユーザー名を以下の式のいずれかで置き換えます。
  - サインインのために入力されたユーザー名を意味する場合は `{0}` を使用します (たとえば、`john.doe`)。
  - ディレクトリサービスによって返された認証済みユーザーの識別名を意味する場合は、`{1}` を使用します (たとえば、`uid=john.doe@example.com,ou=People,o=example.com`)。
- `uidAttributeID` パラメーターを、ユーザー ID を保存するグループ属性の名前に設定します。
- 173 ページの **タスク 7** の説明に従って設定をテストします。

### 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 ブラウザーを使用する方法)

サードパーティの **LDAP** ブラウザーで、以下の手順を実行します。

- 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します。
- 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付けられているグループの識別名の形式を調べます。

- 3 ディレクトリサーバ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します。
- 4 NNMi ユーザーグループに対応するグループを識別して、グループに関連付けられているユーザーの名前の形式を調べます。

#### 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 (他のディレクトリサービスの場合に LDAP ブラウザーを使用する方法)

サードパーティの LDAP ブラウザーで、以下の手順を実行します。

- 1 ディレクトリサーバ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します。
- 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ます。
- 3 また、グループに関連付けられているユーザーの名前の形式も調べます。

#### 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 ブラウザーを使用する方法)

- 1 サポートされる Web ブラウザーで、以下の URL を入力します。
 

**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ーの完全修飾名です。
  - <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します。
  - 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となります。
- 3 グループのプロパティを調べ、そのグループに関連付けられているユーザーの名前の形式を判断してください。

---

## 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 (設定オプション 3) 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。

## 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 以下のコマンドを実行して NNMi LDAP 設定を検証します。

```
nnmlldap.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 以下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ます。

```
nnmlldap.ovpl -reload
```

- 3 以下のコマンドを実行し、あるユーザーの設定をテストします。

```
nnmlldap.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します。Web ブラウザーまたはサードパーティの LDAP ブラウザー (Apache Directory Studio に含まれる LDAP ブラウザーなど) 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、以下のサイトの RFC 1959 「An LDAP URL Format」を参照してください。

<http://labs.apache.org/webarch/uri/rfc/rfc1959.txt>

- 5 %NnmDataDir%\log\%nnm%\jbossServer.log (Windows) または /var/opt/OV/log/nnm/jbossServer.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します。

- 以下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、「ディレクトリサービスへの SSL 接続を設定する」(140 ページ)の説明に従って SSL を有効にします。

```
javax.naming.AuthenticationNotSupportedException: [LDAP:  
error code 13 - confidentiality required]
```

- 以下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、nms-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]
```

## 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。このファイルは以下の場所にあります。

- **Windows:** %NNM\_SHARED\_CONF%\ldap.properties
- **UNIX:** \$NNM\_SHARED\_CONF/ldap.properties

ldap.properties ファイルでは、以下の規則が適用されます。

- 行をコメントアウトするには、その行の先頭を番号記号文字 (#) にします。
- 特殊文字には、以下のルールが適用されます。
  - バックスラッシュ文字 (\)、カンマ (,), セミコロン (;)、プラス記号 (+)、小なり記号 (<)、大なり記号 (>) を指定するには、バックスラッシュ文字でエスケープします。たとえば、¥¥ や ¥+ のように指定します。
  - 文字列の先頭文字または末尾文字としてスペース文字 ( ) を含めるには、バックスラッシュ文字 (\) でエスケープします。
  - 文字列の先頭文字としてシャープ記号 (#) を含めるには、バックスラッシュ文字 (\) でエスケープします。

ここで言及していない文字をエスケープしたり、引用符で囲んだりする必要はありません。

- ▶ ldap.properties ファイルを編集したら、以下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
```

表 12 に、ldap.properties ファイルのパラメーターの説明を示します。

- ▶ 初期の ldap.properties ファイルには、表 12 のリストにあるパラメーターの一部が含まれていない場合があります。必要なパラメーターを追加してください。

表 12 ldap.properties ファイルのパラメーター

パラメーター	説明
java.naming.provider.url	<p>ディレクトリサービスにアクセスするときの <b>URL</b> を指定します。</p> <p><b>URL</b> は、プロトコル (<b>ldap</b>) の後にディレクトリサービスの完全修飾ホスト名が続き、オプションとしてさらにポート番号が続く形式で指定します。次に例を示します。</p> <pre>java.naming.provider.url=ldap://ldap.example.com:389/</pre> <p>ポート番号を省略すると、以下のデフォルト値が適用されます。</p> <ul style="list-style-type: none"> <li>• 非 <b>SSL</b> 接続の場合、デフォルト値は 389 です。</li> <li>• <b>SSL</b> 接続の場合、デフォルト値は 636 です。</li> </ul> <p>複数のディレクトリサービスの <b>URL</b> を指定すると、<b>NNMi</b> 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、<b>NNMi</b> 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、以下同様に対処します。各 <b>URL</b> は 1 つのスペース文字で区切ります。次に例を示します。</p> <pre>java.naming.provider.url=ldap://ldap1.example.com/ ldap://ldap2.example.com/</pre> <p>このパラメーターを設定すると、<b>NNMi</b> とディレクトリサービス間の <b>LDAP</b> 通信が有効になります。<b>LDAP</b> 通信を無効にするには、このパラメーターをコメントアウトしてからファイルを保存します。これにより <b>NNMi</b> は、<code>ldap.properties</code> ファイルの設定を無視します。</p>
java.naming.security.protocol	<p>接続プロトコル指定を指定します。</p> <ul style="list-style-type: none"> <li>• <b>LDAP over SSL</b> 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、このパラメーターを <code>ssl</code> に設定します。次に例を示します。  <pre>java.naming.security.protocol=ssl</pre></li> <li>• ディレクトリサービスで <b>SSL</b> が不要な場合は、このパラメーターをコメントアウトしたままにします。</li> </ul> <p>詳細については、「<a href="#">ディレクトリサービスへの SSL 接続を設定する</a>」(140 ページ)を参照してください。</p>
bindDN	<p>匿名アクセスを許可しない (<b>Active Directory</b> 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。次に例を示します。</p> <pre>bindDN=region1¥¥john.doe@example.com</pre> <ul style="list-style-type: none"> <li>• 平文のパスワードを保存する場合は、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。次に例を示します。  <pre>bindCredential=PasswordForJohnDoe</pre></li> <li>• 暗号化されたパスワードを指定する場合は、<code>ldap.properties</code> ファイルに保存する前に平文のパスワードを以下のコマンドで暗号化します。  <pre><b>nnmlldap.ovpl -encrypt &lt;mypassword&gt;</b></pre> <p>例: <code>bindCredential={ENC}uaF22C+0CF9VozBVYj80Aw==</code></p> <p>この暗号化パスワードは、その作成先の <b>NNMi</b> インスタンスでのみ機能します。他の <b>NNMi</b> インスタンスには使用しないでください。詳細については、<a href="#">nnmlldap.ovpl</a> リファレンスページ、または <b>UNIX</b> のマンページを参照してください。</p> </li> </ul>



表 12 ldap.properties ファイルのパラメーター ( 続き )

パラメーター	説明
bindCredential	<p>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。次に例を示します。</p> <pre>bindCredential=PasswordForJohnDoe</pre>
baseCtxDN	<p>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します。</p> <p>形式は、ディレクトリサービスの属性名と値のカンマ区切りリストです。次に例を示します。</p> <ul style="list-style-type: none"> <li>• baseCtxDN=CN=Users,DC=ldapserver,DC=example,DC=com</li> <li>• baseCtxDN=ou=People,o=example.com</li> </ul> <p>詳細については、「<a href="#">ユーザー識別</a>」(183 ページ)を参照してください。</p>
baseFilter	<p>NNMi にサインインするユーザー名の形式を指定します。</p> <p>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に関連付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他の文字が含まれます。</p> <ul style="list-style-type: none"> <li>• 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。次に例を示します。 <ul style="list-style-type: none"> <li>- baseFilter=CN={0}</li> <li>- baseFilter=uid={0}</li> </ul> </li> <li>• 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。次に例を示します。 <ul style="list-style-type: none"> <li>- baseFilter=CN={0}@example.com</li> <li>- baseFilter=uid={0}@example.com</li> </ul> </li> </ul> <p>詳細については、「<a href="#">ユーザー識別</a>」(183 ページ)を参照してください。</p>
defaultRole	<p>オプション。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。このパラメーターの値は、(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。</p> <p>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。</p> <p>有効な値は、admin、level2、level1、または guest です。</p> <p>admin は有効な値ですが、デフォルトロールとしての admin の使用は慎重に検討する必要があります。</p> <p>この名前は、定義済み NNMi ユーザーグループ名の一意的な名前です (186 ページの表 11 で定義)。</p> <p>次に例を示します。</p> <pre>defaultRole=guest</pre> <p>コメントアウトまたは省略すると、NNMi はデフォルト値を使用しません。</p>



表 12 ldap.properties ファイルのパラメーター ( 続き )

パラメーター	説明
rolesCtxDN	<p>ディレクトリサーバ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。</p> <p>形式は、ディレクトリサービスの属性名と値のカンマ区切りリストです。次に例を示します。</p> <ul style="list-style-type: none"> <li>rolesCtxDN=CN=Users,DC=ldapserver,DC=example,DC=com</li> <li>rolesCtxDN=ou=Groups,o=example.com</li> </ul> <p>他のディレクトリサービス (<b>Active Directory</b> 以外) では、検索速度を高めるため、<b>NNMi</b> 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。たとえば、ディレクトリサービスに <b>USERS-NNMi-administrators</b> や <b>USERS-NNMi-level1Operators</b> などの名前のグループが含まれる場合は、以下のような検索コンテキストを使用できます。</p> <pre>rolesCtxDN=cn=USERS-NNMi-*,ou=Groups,o=example.com</pre> <p>このパラメーターを設定すると、<b>LDAP</b> を介した <b>NNMi</b> 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。</p> <p><b>LDAP</b> を介した <b>NNMi</b> 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、このパラメーターをコメントアウトしてからファイルを保存します。<b>NNMi</b> 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。</p> <p>詳細については、「<a href="#">ユーザーグループの識別</a>」(186 ページ) を参照してください。</p>
roleFilter	<p>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。</p> <p>形式は、ユーザー <b>ID</b> 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 <b>ID</b> の形式に関連付ける文字列で構成されます。ユーザー名文字列には、以下の式の 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他の文字が含まれています。</p> <ul style="list-style-type: none"> <li>式 {0} は、サインインで入力されたユーザー名を示します (たとえば、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ー例： roleFilter=member={0}</li> <li>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 (たとえば、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ー例： roleFilter=member={1}</li> </ul> <p>詳細については、「<a href="#">ユーザーグループの識別</a>」(186 ページ) を参照してください。</p>
uidAttributeID	<p>ディレクトリサービスユーザー <b>ID</b> を保存するグループ属性を指定します。</p> <p>次に例を示します。</p> <pre>uidAttributeID=member</pre> <p>詳細については、「<a href="#">ユーザーグループの識別</a>」(186 ページ) を参照してください。</p>

表 12 ldap.properties ファイルのパラメーター ( 続き )

パラメーター	説明
userRoleFilterList	<p>オプション。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のできる NNMi ユーザーグループを制限します。</p> <p>このリストのユーザーグループは、LDAP で認証されるディレクトリサービスユーザー名のみ適用されます。このパラメーターでは、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。</p> <p>1 つ以上の定義済み NNMi ユーザーグループ名の一意的名前 (186 ページの表 11 で定義) をセミコロンで区切ったリストという形式です。</p> <pre>userRoleFilterList=admin;globalops;level2;level1</pre>
searchTimeLimit	<p>オプション。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。</p> <p>次に例を示します。</p> <pre>searchTimeLimit=10000</pre>

## 例

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

Active Directory の場合の ldap.properties ファイルの例を以下に示します。

```
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
```

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

他のディレクトリサービスの場合の ldap.properties ファイルの例を以下に示します。

```
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
```



# NAT 環境の重複 IP アドレスの 管理

NNMi では、ネットワークアドレス変換 (NAT) の実装によって生じる重複 IP アドレスを含むネットワーク領域を容易に管理できます。

---

## 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パブリック（外部）インターネットと相互接続するために使用します。このテクノロジーは、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 (RFC 1918 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ーを求める声が強くなっていました。

具体的に言うと、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（パブリック）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によりこれを実現します。

---

## NAT の利点

NAT には、以下のような利点があります。

- 多数のホストを 1 つの動的なパブリック（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

## サポートされる NAT タイプ

NNMi では、以下のタイプの 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 (PAT) (ネットワークアドレスおよびポート変換 (NAPT) と呼ばれる) — 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。

## 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ー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ーは、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ー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、「[NNMi セキュリティおよびマルチテナント](#)」(211 ページ) を参照してください。

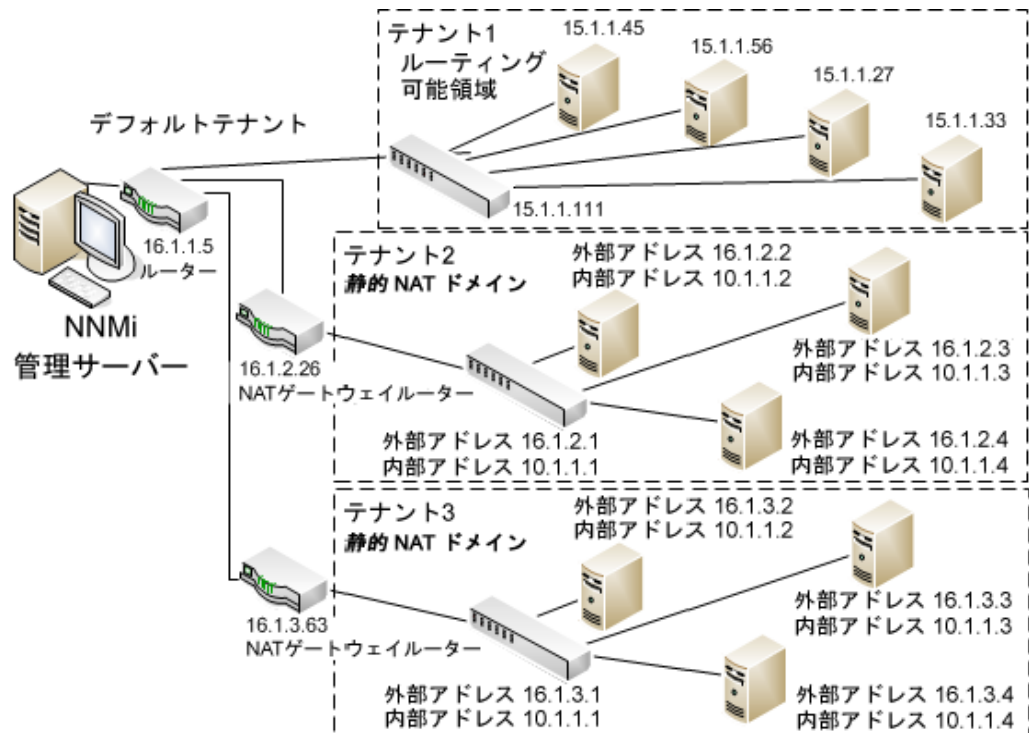
ネットワーク管理環境に重複アドレスドメインが含まれている場合、最低でも一意のテナントとして各ドメインを設定する必要があります。使用しているプロトコルによって、NNMi の実装方法や要件は異なる場合があります。たとえば、動的 NAT または動的 PAT を使用している場合、追加のハードウェアおよびライセンスが必要になります。使用している NAT プロトコルのタイプに基づいて、後続の適切な項を参照してください。

## 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、「[NNMi セキュリティおよびマルチテナント](#)」(211 ページ) 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 の設定例として図 14 を参照してください。

図 14 静的 NAT の設定例



▶ 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) はテナントをまたぐことができません。

▶ 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。これにより、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。

## 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。

## 静的 NAT での通信

### 静的 NAT 環境における管理アドレスの ICMP ポーリングの管理

NAT 環境では、ファイアウォールにより、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 ノードとやり取りすること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。

### NAT 環境における管理アドレスの ICMP ポーリングの有効化

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デフォルトで自動的に有効になります。NAT 環境がある場合、この設定を無効にしないことをお勧めします。

(無効になっている場合に) 管理アドレスの ICMP ポーリングを有効にするには、以下の手順を実行します。

- 1 ワークスペースのナビゲーションパネルで、[設定] ワークスペースを選択して [モニタリング] フォルダを展開し、[モニタリングの設定] を選択して [デフォルト設定] タブを探します。
- 2 [ICMP 管理アドレスポーリング] を有効にします。NNMi ヘルプの「デフォルトのモニタリングを設定する」を参照してください。

SNMP エージェントに対して [アクション] > [モニタリングの設定] を実行した後に NNMi が表示する情報を確認します。表示される情報に、NNMi が管理アドレスのポーリングを有効にしているかどうかが表示されます。

### NNMi に対する変更点

ICMP 管理アドレスポーリングが有効になっていると、NNMi が以下のように変更されます。

- [エージェント ICMP 状態] フィールドが、以下のフォームに表示されます。
  - [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 状態の表示場所を変更します。NNMi は、SNMP エージェントステータスの判断方法も変更します。

表 13 に、エージェント ICMP および IP アドレス状態のポーリングアクションを示します。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応じて、これらのアクションを実行します。表 13 の影付きの先頭行はデフォルト設定を示します。



表 13 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	エージェント ICMP 状態	IP アドレス状態
有効	無効	ポーリング	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

表 14 に、SNMP エージェントと ICMP の応答に合わせて APA によって決定される SNMP エージェントステータスに対する変更点を示します。

表 14 SNMP エージェントステータスの判断

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
応答	応答	正常域
応答	無応答	警戒域
無応答	応答	危険域
無応答	無応答	危険域

管理アドレスの ICMP ポーリングを有効にすると、APA は、結果とインシデントの生成時に、管理アドレス ICMP の応答と SNMP エージェントの応答を考慮するようになります。

## 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 領域内に存在する可能性があります。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、または UNIX のマン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。

## 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。本項では、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。

## SNMPv2c トラップ

表 15 に、SNMPv2c トラップの形式を示します。この表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。

表 15 SNMPv2c トラップの形式

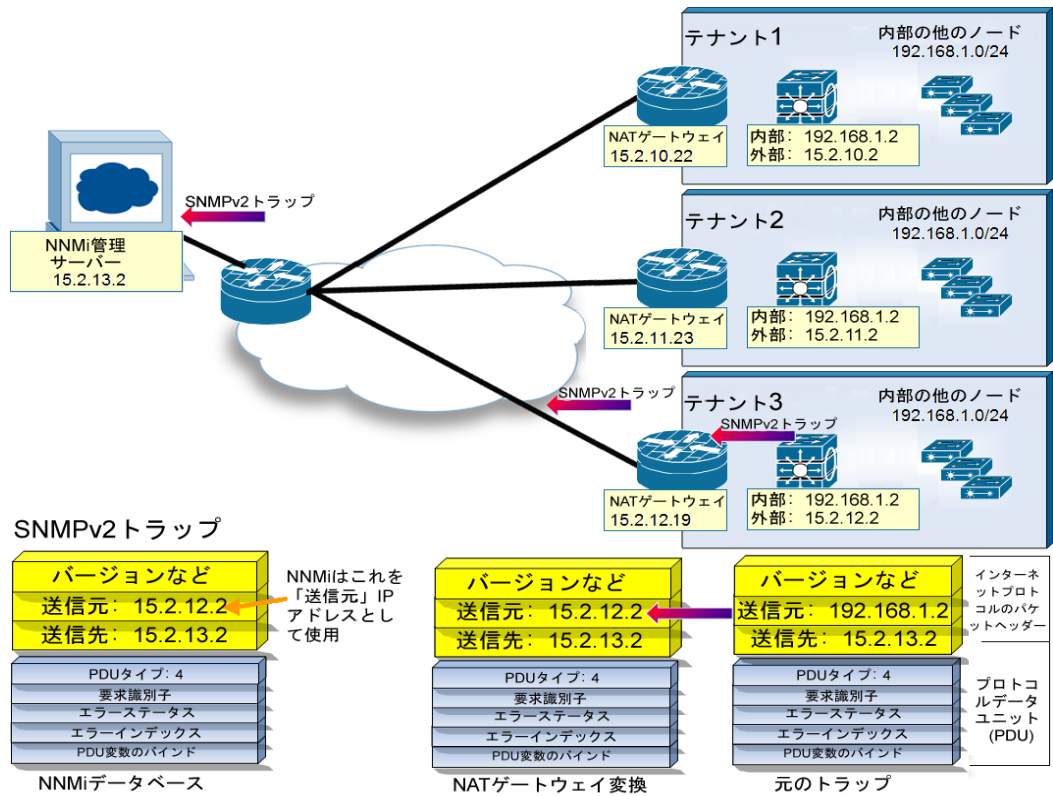
バージョンおよびその他の情報
ソースアドレス
デスティネーションアドレス
PDU タイプ: 4
要求識別子
エラーステータス
エラーインデックス
PDU 変数のバインド

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、IP パケットヘッダー内にはトラップのソースフィールドのみがあります。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 アドレスに関連付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。

図 15 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 15.2.13.2 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 15 SNMPv2c の例



### 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。表 16 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。

表 16 SNMPv1 トラップの形式

バージョンおよびその他の情報
ソースアドレス
デスティネーションアドレス
<b>PDU タイプ: 4</b>
エンタープライズ
エージェントアドレス
汎用トラップコード
固有トラップコード
タイムスタンプ
PDU 変数のバインド

エージェントアドレスはヘッダーではなく PDU に組み込まれているため、通常、この値は NAT ルーターによって変換されません。ヘッダーのアドレスを認識して、ペイロードのエージェントアドレスを無視するようにNNMiを設定するには、以下の手順を実行します。

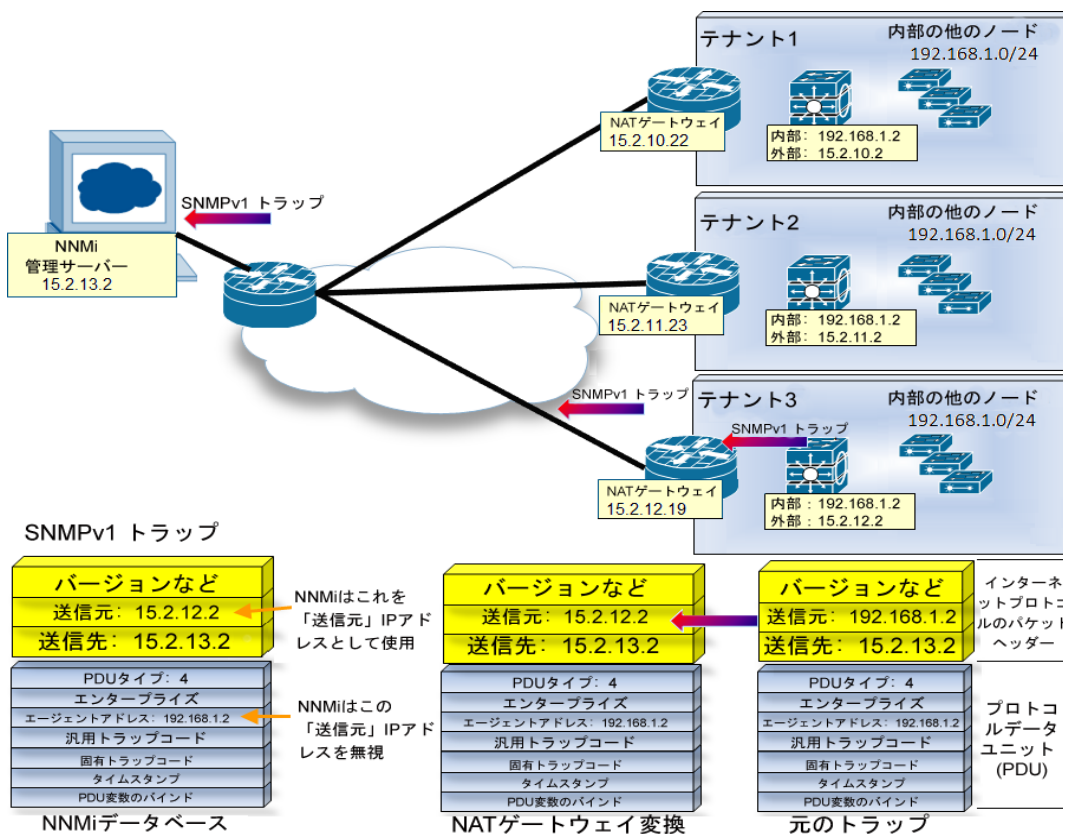
- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\nms-jboss.properties
  - UNIX: \$NNM\_PROPS/nms-jboss.properties
- 2 以下の行を探します。
 

```
#!com.hp.nnm.trapd.useUdpHeaderIpAddress=false
```
- 3 以下のように値を **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。

図 16 に、競合する IP 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 16 SNMPv1 の例





NNMi では、関連する以下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。

## サブネットと静的 NAT

サブネットおよび NAT に関しては、以下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ー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 つのテナントにのみ割り当てられます)。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、いずれかのテナントがデフォルトテナントである場合にのみ使用できます。

## グローバルネットワーク管理と静的 NAT

リージョナルマネージャーごとに、少なくとも 1 つの静的またはルーティング可能 (非変換) アドレスが存在している必要があります。これにより、NNMi 管理サーバーが相互に通信することができ、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、「[グローバルネットワーク管理](#)」(235 ページ) を参照してください。

---

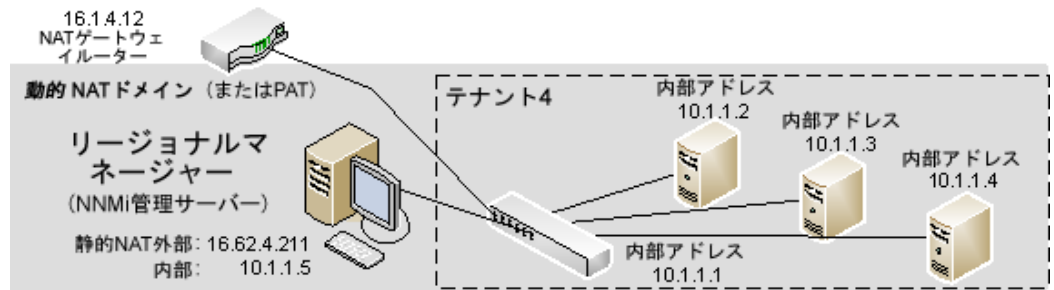
## 動的 NAT および動的 PAT の考慮事項

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的な同じテナントに属している必要があります。NNMi 管理サーバーは、リージョナルマネージャーとしてグローバルネットワーク管理環境に参加する必要があります。動的 NAT の設定例として以下の図を参照してください。



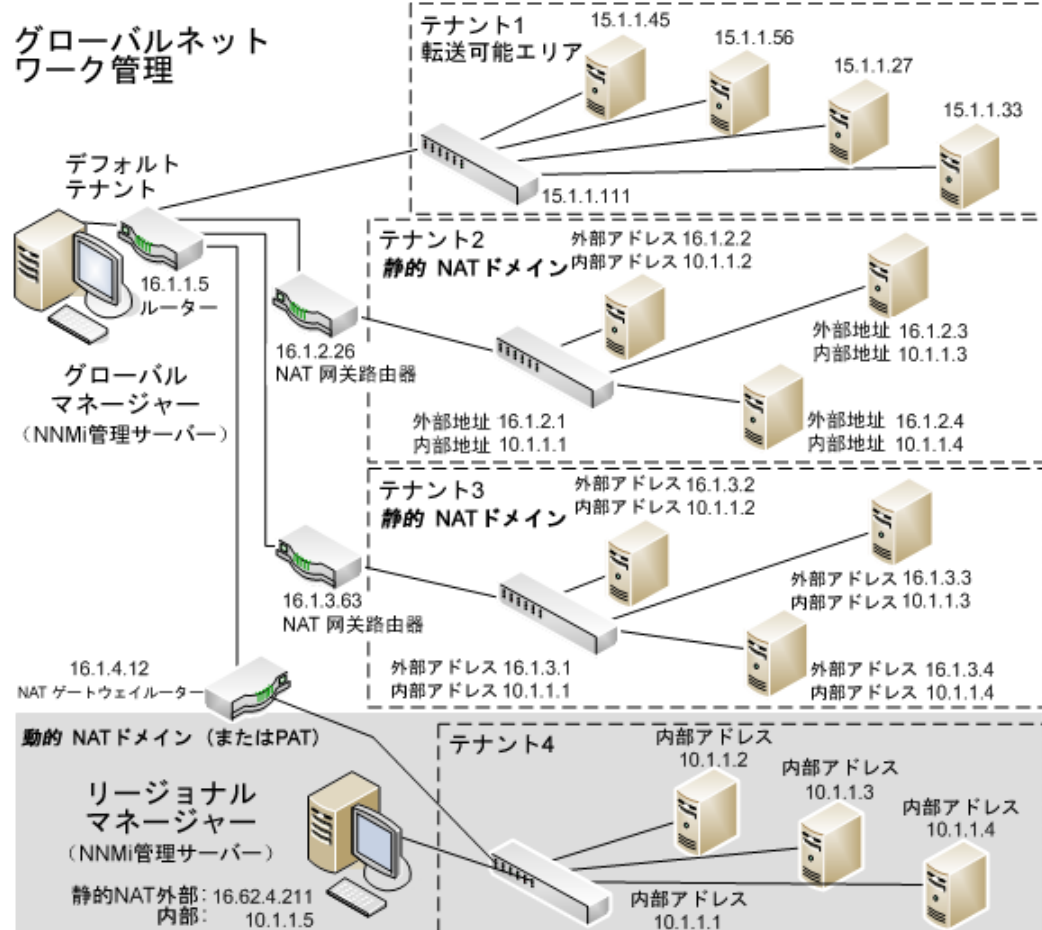
リージョナルマネージャーが NAT ファイアウォールの背後にある場合、その外部 (パブリック) アドレスは静的アドレスである必要があります。

図 17 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以下の図を参照してください。

図 18 NAT 環境内のグローバルネットワーク管理設定の例



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。これにより、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、「[グローバルネットワーク管理](#)」(235 ページ) 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。

## 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 Software が必要になります。

動的 NAT または動的 PAT で設定されたアドレスドメインごとに NNMi リージョナルマネージャーが必要です。

## 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 領域内に存在する可能性があります。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 (これはシード済み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ーフェースを使用)、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、UNIX のマンページ、または NNMi ヘルプを参照してください。

## サブネットと動的 NAT および動的 PAT

サブネットおよび NAT に関しては、以下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ー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 つのテナントにのみ割り当てられます)。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、いずれかのテナントがデフォルトテナントである場合にのみ使用できます。



## グローバルネットワーク管理と動的 NAT および動的 PAT

リージョナルマネージャーごとに、少なくとも 1 つの静的またはルーティング可能 (非変換) アドレスが存在している必要があります。これにより、NNMi 管理サーバーが相互に通信することができ、通信を隠ぺいしてセキュリティを確保できます。

▶ リージョナルマネージャーが NAT ファイアウォールの背後にある場合、その外部アドレスは静的アドレスである必要があります。

グローバルネットワーク管理の詳細については、「[グローバルネットワーク管理](#)」(235 ページ)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。

## 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「[NNMi セキュリティおよびマルチテナント](#)」(211 ページ)を参照してください。

オプション。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 ドメイン外に存在する場合、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。「[動的 NAT および動的 PAT の考慮事項](#)」(205 ページ)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可能性があります。

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、次のいずれかを実行します。

- NNMi コンソールで、[重複する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、`nnmloadipmappings.ovpl` リファレンスページ、または UNIX のマンページを参照してください。

## 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、以下の 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 (RFC 1918):



2012年5月

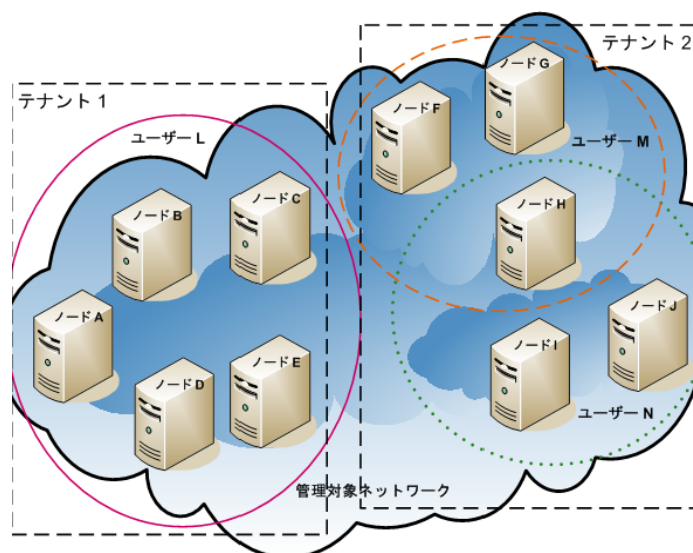
- 10.0.0.0 ~ 10.255.255.255 (24 ビットブロック)
- 172.16.0.0 ~ 172.31.255.255 (20 ビットブロック)
- 192.168.0.0 ~ 192.168.255.255 (16 ビットブロック)

IPv6 プライベートアドレス範囲：

- fc00::/7 アドレスブロック = RFC 4193 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 (RFC 3879)



# NNMi セキュリティおよびマルチテナント



▶ 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 NAT、または動的ポートアドレス変換 (PAT) 領域内に存在する可能性があります。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 (これはシード済み検出を使用して行います)。詳細については、「[NAT 環境の重複 IP アドレスの管理](#)」(197 ページ) および NNMi ヘルプを参照してください。

デフォルトでは、すべての 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。使用環境でこのデフォルト設定を許容できる場合、この章を読む必要はありません。

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ー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に役立ちます。また、サービスプロバイダーが NNMi を組織ごとに設定する場合にも役立ちます。

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。内容は以下のとおりです。

- 「[オブジェクトのアクセス制限による影響](#)」(212 ページ)
- 「[NNMi セキュリティモデル](#)」(213 ページ)
- 「[NNMi テナントモデル](#)」(218 ページ)
- 「[NNMi のセキュリティおよびマルチテナント設定](#)」(221 ページ)
- 「[NNMi セキュリティ、マルチテナント、およびグローバルネットワーク管理](#)」(230 ページ)
- 「[NPS レポートへの選択インタフェースの追加](#)」(232 ページ)

『[HP Network Node Manager i Software ステップバイステップガイド \(セキュリティグループの使用に関するホワイトペーパー\)](#)』(HP Network Node Manager i Software Step-by-Step Guide to Using Security Groups White Paper) も参照してください。

## 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以下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：
  - 各 NNMi コンソールユーザーには、それぞれのユーザーの NNMi ユーザーアカウント設定に対応するノードのみが表示されます。
  - 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、関連するノードの少なくとも 1 つを表示できる場合にのみ表示されます。
  - NNMi コンソールユーザーには、ノードグループの中の少なくとも 1 つ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のみが表示されます。
  - Network Performance Server (NPS) レポートの場合、NNMi 管理者はインタフェースのアクセス制御の継承を選択的に上書きできます。詳細については、「[NPS レポートへの選択インタフェースの追加](#)」(232 ページ)を参照してください。
- マップおよびパスビュー：
  - マップには、関与している両方のノードを表示する権限を NNMi コンソールユーザーが持っている接続が表示されます。
  - パスビューでは、NNMi コンソールユーザーがアクセスできないすべての中間ノードは省略されるか、クラウドとして表示されます。
  - NNM iSPI for MPLS および NNM iSPI for IP Multicast については、マップとパスビューに NNMi コンソールユーザーがアクセスできないノードが含まれている場合、NNM iSPI には接続中のインタフェースとノードの名前しか表示されません。アクセスできないノードのアイコンは白色で表示され、それらのノードのステータスと詳細情報を入手できないことが示されます。
  - NNM iSPI for IP Telephony については、マップとパスビューに NNMi コンソールユーザーがアクセスできないノードが含まれている場合、NNM iSPI には接続されているインタフェースとノードの名前しか表示されません。アクセスできないノードのアイコンには NNMi ステータスが表示されますが、アクションを行ってもすべて失敗します。
- インシデント：
  - ソースノードが NNMi トポロジ内にあるインシデントについては、NNMi コンソールユーザーには、その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のみが表示されます。
  - 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、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(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付けることで)決定します。
  - ソースノードが NNMi トポロジ内には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、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(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付けることで)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可能性があります。

- **NNMi コンソールアクション:**
  - 何も選択を行わずに実行されるアクションについては、**NNMi** コンソールユーザーには、そのユーザーが実行する権限を持っているアクションのみが表示されます。
  - 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。
  - マップビューや、**NNM iSPI** テーブルビューおよびフォームについては、**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を行うことができません。
- **MIB ブラウザーおよび Line Grapher:**
  - **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** にログオ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、使用可能なトポロジを制限します。

---

## NNMi 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、以下の利点があります。

- **NNMi** コンソールオペレーターのネットワークのビューを制限できます。オペレーター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ーアクセスをカスタマイズできます。オペレーター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)] ビューおよび **Network Performance Server** レポート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、以下のような場合に使用されます。

- **NNMi** オペレーターがサイト(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)のノードのみが表示される各サイトビューを **NNMi** オペレーターに提供する。

- 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ますが、NNMi オペレーターには導入したノードのみが表示されます。
- すべての NOC オペレーター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 オペレーターに完全なネットワークビューを提供し、地域の NOC オペレーターのビューを制限する。

## 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のみに関連付けられます。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付けることができます。

各ユーザーアカウントは、以下のユーザーグループにマッピングされます。

- 以下に示す事前設定された 1 つ以上の NNMi ユーザーグループ：
  - NNMi 管理者
  - NNMi グローバルオペレーター
  - NNMi レベル 2 オペレーター
  - NNMi レベル 1 オペレーター
  - NNMi ゲストユーザー

このマッピングは NNMi コンソールアクセスに必要で、これによって NNMi コンソール内で使用できるアクションが決まります。ユーザーアカウントがこれらの複数の NNMi ユーザーグループにマッピングされている場合、許可されるアクションのスーパーセットがユーザーに付与されます。



[NNMi Web サービスクライアント] ユーザーグループでは、NNMi コンソールへのアクセス権は付与されませんが、すべての NNMi オブジェクトへの管理者レベルのアクセス権が付与されます。



NNMi グローバルオペレーターユーザーグループ (globalops) では、トポロジオブジェクトのみ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、ユーザーを他のいずれかのユーザーグループ (level2、level1、または guest) に割り当てる必要があります。

globalops ユーザーグループはデフォルトですべてのセキュリティグループにマッピングされるため、管理者はこのユーザーグループをセキュリティグループにマッピングしないようにする必要があります。

- セキュリティグループにマッピングされる 0 個以上の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。たとえば、インタフェース **X** および **Y** を含むノードへのオブジェクトオペレーターレベル **1** のアクセス権限があるユーザーには、以下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ーレベル **1** のアクセス権限があります。

- 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、以下のセキュリティグループがあります。

- デフォルト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、[デフォルトセキュリティグループ] に関連付けられるノードと、[デフォルト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** トポロジ内には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付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付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

以下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** ユーザーグループのみ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 Service Client** 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 Service Client**] ユーザーグループにのみマッピングしてください。

## セキュリティグループ構造の例

図 19 内の 3 つの楕円形は、この **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、4 つの各サブグループが一意的セキュリティグループに対応している必要があります。一意的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する使用可能なユーザーアクセスのレベルを表すことができます。

216 ページの表 17 に、このトポロジにおける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可能性があります)。217 ページの表 18 に、このトポロジにおけるいく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 19 ユーザーアクセス要件に対応するトポロジの例

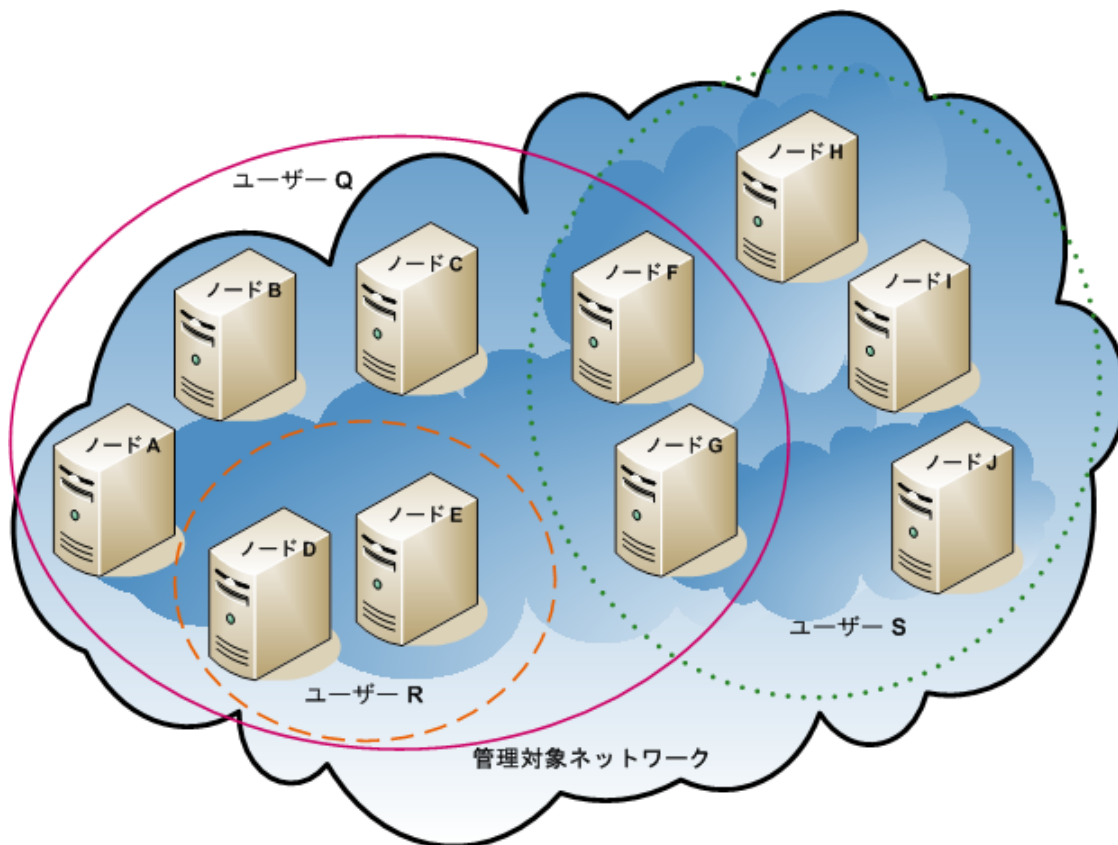


表 17 セキュリティグループマッピングの例

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限
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ーレベル 2
		UG1 レベル 1	オブジェクトオペレーターレベル 1
		UG1 ゲスト	オブジェクトゲスト
SG2	D、E	UG2 管理者	オブジェクト管理者
		UG2 レベル 2	オブジェクトオペレーターレベル 2
		UG2 レベル 1	オブジェクトオペレーターレベル 1
		UG2 ゲスト	オブジェクトゲスト



表 17 セキュリティグループマッピングの例 ( 続き )

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限
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ーレベル 2
		UG3 レベル 1	オブジェクトオペレーターレベル 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ーレベル 2
		UG4 レベル 1	オブジェクトオペレーターレベル 1
		UG4 ゲスト	オブジェクトゲスト

表 18 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 オペレーター	なし	このユーザーには、ピンクの楕円形 ( 実線 ) に含まれるノードへのオペレーターレベル 2 のアクセス権限があります。
	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 オペレーター	なし	このユーザーには、オレンジの楕円形 ( 破線 ) に含まれるノードへのオペレーターレベル 1 のアクセス権限があります。
	UG2 レベル 1	D、E	
ユーザー S	NNMi レベル 2 オペレーター	なし	このユーザーには、緑の楕円形 ( 点線 ) に含まれるノードへのオペレーターレベル 2 のアクセス権限があります。
	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 オペレーター	なし	このユーザーは、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。
	UG1 ゲスト	A、B、C	
	UG2 管理者	D、E	このユーザーには、ノード D および E への管理アクセス権がありますが、管理アクセス権が必要なツールのメニュー項目は表示できません。ユーザーに NNMi 管理サーバーへのアクセス権がある場合は、ノード D および E に対してのみ、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。
	UG3 レベル 2	F、G	
	UG4 レベル 1	H、I、J	

## NNMi 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ー（特に管理対象サービスプロバイダー）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、以下の利点があります。

- 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と **Network Performance Server** レポートを、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ー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。
- アドレス変換プロトコルを使用した場合、重複しているアドレスドメインを管理できます。

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ーに、異なる顧客ビューを提供することができます。



各インスタンスが一意的テナントで設定されている場合、1つの NNMi 管理サーバーで任意の数の静的ネットワークアドレス変換 (NAT) インスタンスを監視できます。詳細については、「[NAT 環境の重複 IP アドレスの管理](#)」(197 ページ) および NNMi ヘルプを参照してください。

### 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つのみです。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られるテナントを指定できます。NNMi によって、検出された他のすべてのノード（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、この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、([デフォルトセキュリティグループ] を介して) このテナントに関連付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**

以下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。
- ユーザーが組織をまたいで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つのテナントのみに対応するノードしか含まれないようにします。

**テナント構造の例**

図 20 に、NNMi トポロジ内に 2 つのテナントが含まれている様子を長方形の線で囲んで示します。これらの 3 つの楕円形は、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のみが必要です。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。

220 ページの表 19 に、このトポロジにおける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可能性があります）。220 ページの表 20 に、このトポロジにおけるいく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 20 複数のテナントのトポロジの例

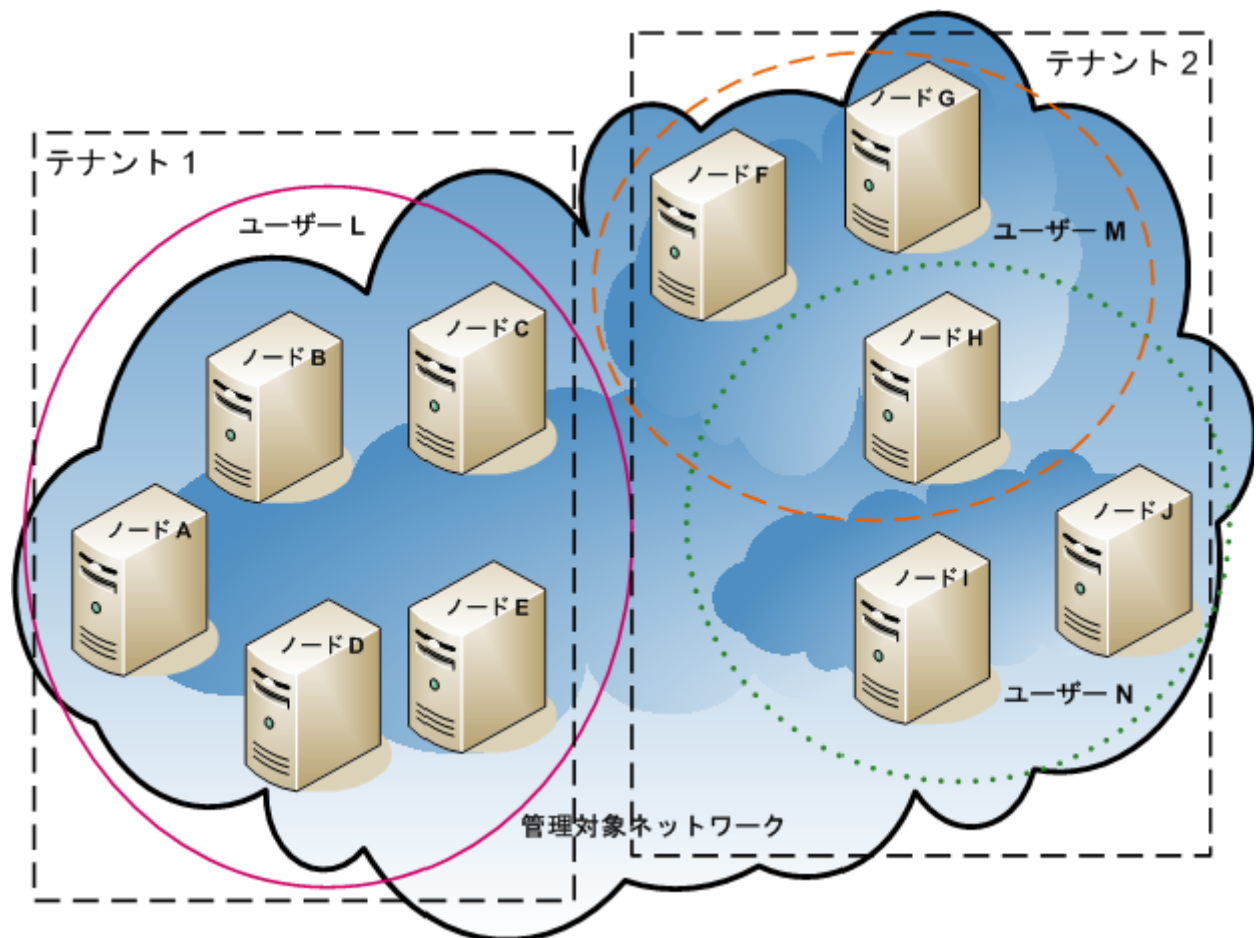


表 19 複数のテナントのセキュリティグループマッピングの例

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限
T1 SG	A、B、C、D、E	T1 管理者	オブジェクト管理者
		T1 レベル 2	オブジェクトオペレーターレベル 2
		T1 レベル 1	オブジェクトオペレーターレベル 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ーレベル 2
		T2_a レベル 1	オブジェクトオペレーターレベル 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ーレベル 2
		T2_b レベル 1	オブジェクトオペレーターレベル 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ーレベル 2
		T2_c レベル 1	オブジェクトオペレーターレベル 1
		T2_c ゲスト	オブジェクトゲスト

表 20 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 オペレーター	なし	このユーザーには、テナント 1 のすべてのノードをグループ化する、ピンクの楕円形 (実線) に含まれるノードへのオペレーターレベル 2 のアクセス権限があります。
	T1 レベル 2	A、B、C、D、E	
ユーザー M	NNMi レベル 1 オペレーター	なし	このユーザーには、テナント 2 のノードのサブセットをグループ化する、オレンジの楕円形 (破線) に含まれるノードへのオペレーターレベル 1 のアクセス権限があります。
	T2_a レベル 1	F、G	
	T2_b レベル 1	H	

表 20 複数のテナントのユーザーアカウントマッピングの例 ( 続き )

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 N	NNMi レベル 2 オペレーター	なし	このユーザーには、テナント 2 のノードのサブセットをグループ化する、緑の楕円形 ( 点線 ) に含まれるノードへのオペレーターレベル 2 のアクセス権限があります。
	T2_b レベル 2	H	
	T2_c レベル 2	I、J	

## NNMi のセキュリティおよびマルチテナント設定

▶ 各インスタンスが一意のテナントで設定されている場合、1 つの NNMi 管理サーバーで任意の数の静的ネットワークアドレス変換 (NAT) インスタンスを監視できます。詳細については、「[NAT 環境の重複 IP アドレスの管理](#)」(197 ページ) および NNMi ヘルプを参照してください。

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ーアクセス権限を表示および設定できます。

NNMi 管理者が少なくとも 1 つ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** ] 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。

NNMi 管理者が少なくとも 1 つ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。

### ベストプラクティス

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アクセス権限とに対応する 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のみ、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 2 オペレーターまたは NNMi レベル 1 オペレーターのユーザーグループに割り当てます ( レベル 1 オペレーターにはレベル 2 オペレーターよりも低いアクセス権が与えられています )。また、オブジェクト管理者オブジェクトアクセス権限を使用して、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て行われます。

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、「[NNMi と LDAP によるディレクトリサービスの統合](#)」(163 ページ)を参照してください。

## 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いく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] は、NNMi セキュリティ設定に関してのみ使用できます。テナント情報は含まれていません。

[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。

### 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。

以下の 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、以下の 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付けが表示されます。
- ユーザーアカウントマッピングに変更を行っても、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 コンソールに次回ログオンしたときに、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付けが表示されます。
- 初期設定の後、セキュリティグループマッピングに関連付けられたオブジェクトのアクセス権限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限を変更するには、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を行う場合に便利です。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、カンマ区切り値 (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。

## ベストプラクティス

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。

以下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
# 2つのユーザーを作成する
nnmsecurity.ovpl -createUserAccount user1 -password password -role level1
nnmsecurity.ovpl -createUserAccount user2 -password password -role level2

# 2つのグループを作成する
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
```

```
# 2つのセキュリティグループを作成する
nmmsecurity.ovpl -createSecurityGroup secgroup1
nmmsecurity.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
nmmsecurity.ovpl -assignUserGroupToSecurityGroup -userGroup local1 ¥
  -securityGroup secgroup1 -role level1
nmmsecurity.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
nmmsecurity.ovpl -assignNodeToSecurityGroup -node mplspe01
  -securityGroup secgroup1
nmmsecurity.ovpl -assignNodeToSecurityGroup -node vwan_router-1
  -securityGroup secgroup1
nmmsecurity.ovpl -assignNodeToSecurityGroup -node vwan_router-2
  -securityGroup secgroup1
nmmsecurity.ovpl -assignNodeToSecurityGroup -node data_center_1
  -securityGroup secgroup2
nmmsecurity.ovpl -assignNodeToSecurityGroup -node mplspe03
  -securityGroup secgroup2
```

## テナントの設定



各インスタンスが一意的テナントで設定されている場合、1つの NNMi 管理サーバーで任意の数の静的ネットワークアドレス変換 (NAT) インスタンスを監視できます。詳細については、「[NAT 環境の重複 IP アドレスの管理](#)」(197 ページ) および NNMi ヘルプを参照してください。

NNMi では、以下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** ] フォームは、個々のテナントを処理する際に役立ちます。
- nmmsecurity.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を行う場合に便利です。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。

各 NNMi トポロジオブジェクトをテナント (組織) に割り当てるために 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。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。

NNMi マルチテナントの設定に関しては、以下に注意してください。

- 検出されたノードに NNMi によって割り当てられるセキュリティグループは、そのノードに関連付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 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以下に示します。


- 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します。
  - 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のみ、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します。
- 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します。
- 4 事前に定義された NNMi ユーザーグループと、[デフォルト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します。
 

この手順により、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、そのユーザーに間違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 管理者のみです。
- 5 特定されたテナントを設定します。
  - 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いずれかに、[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、検出の準備を行います。
 

 ノードのグループを検出した後、[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、以下を実行します。
  - 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。
- 8 226 ページの [手順 4](#) を継続します。

## セキュリティグループの設定

 NNMi をディレクトリサービスと統合して、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、以下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** ] は、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は、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。

- `nmsecurity.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を行う場合に便利です。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。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。



この例では、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。たとえば、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、以下に注意してください。

- 検出されたノードに NNMi によって割り当てられるセキュリティグループは、そのノードに関連付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以下に示します。

- 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します。
- 2 事前に定義された NNMi ユーザーグループと、[デフォルト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します。

この手順により、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、そのユーザーに間違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 管理者のみです。

- 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します。特定のノードは 1 つのセキュリティグループにのみ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します。
  - 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する場合は、以下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 「設定の確認」(227 ページ)の説明に従って、設定を確認します。
  - 7 セキュリティ設定を管理します。
    - [デフォルト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。

## 設定の確認

セキュリティ設定が適切であるかを確認するために、設定の各側面を別個に確認します。このセクションでは、設定を確認するためのいく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。



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]で、`-displayConfigReport` オプションを `nnmsecurity.ovpl` コマンドに設定してアクセスします。

### 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確認する方法の1つとして、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る方法があります。

また、`-listNodesInSecurityGroup` 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。

### 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確認する方法の1つとして、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る方法があります。また、各マッピングのオブジェクトアクセス権限も確認します。

あるいは、[セキュリティウィザード]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する] 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。

また、`-listUserGroupsForSecurityGroup` 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。

### 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(高い方から順に表示)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 2 オペレーター
- NNMi レベル 1 オペレーター
- NNMi ゲストユーザー

その他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 グローバルオペレーターユーザーグループでは、トポロジオブジェクトのみにアクセス権が与えられます。globalops ユーザーが NNMi コンソールにアクセスできるユーザーグループ (level2、level1、または guest など) に関連付けられていない場合、そのユーザーは NNMi コンソールにはアクセスできません。

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** ] ページにリストされます。[ **ツール** ] > [ **セキュリティレポート** ] メニュー項目で、`-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。



NNMi コンソールの各 [ **ツール** ] および [ **アクション** ] メニュー項目には、デフォルトの NNMi ロールが関連付けられています (各 [ **アクション** ] メニュー項目に関連付けられているデフォルトの NNMi ロールを確認するには、NNMi ヘルプの「NNMi に用意されているアクション」を参照してください)。NNMi が提供するメニュー項目の設定をメニュー項目に割り当てられたデフォルトの NNMi ロールよりも低いレベルのロールに変更すると、NNMi はその変更を無視します。デフォルトの NNMi ロールよりも低いレベルのロールが割り当てられたすべてのユーザーグループは、メニュー項目にはアクセスできません。

#### 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確認する方法の 1 つとして、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る方法があります。

あるいは、[ **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ピング** ] 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。

また、`-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することもできます。

#### 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の 1 つとして、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以下の情報がリストされます。

- 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 2 オペレーター、NNMi レベル 1 オペレーター、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。この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。

- この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 ロールを設定する事前に設定された 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他のすべてのユーザーグループが含まれています。

## 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

表 21 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 (nnmconfigexport.ovpl -c で使用可能) を示しています。これらの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 21 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。

## NNMi セキュリティ、マルチテナント、およびグローバルネットワーク管理

グローバルネットワーク管理 (GNM) 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。GNM 環境では、指定されたノードのテナント UUID は各グローバルマネージャーとリージョナルマネージャーで同じです。

ノードのセキュリティグループは、トポロジにそのノードが含まれる各 NNMi 管理サーバーに設定されます。したがって、トポロジ内のオブジェクトへのユーザーアクセスは、GNM 環境の各 NNMi 管理サーバーに別個に設定されます。グローバルマネージャーとリージョナルマネージャーが使用するセキュリティグループ定義は、同じである場合も、異なる場合もあります。

グローバルマネージャーとリージョナルマネージャーに同様のユーザーアクセスを設定する場合、いくつかの裏技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を行う必要があります。

▶ 動的ネットワークアドレス変換 (NAT) または動的ポートアドレス変換 (PAT) の各グループには、NNMi グローバルネットワーク全体の管理設定内で一意のテナントに加えて、NNMi リージョナルマネージャーが必要です。「[NAT 環境の重複 IP アドレスの管理](#)」(197 ページ) を参照してください。NNMi ヘルプも参照してください。

### ベストプラクティス

グローバルマネージャーにすべてのテナントとセキュリティグループを定義します。nnmconfigexport.ovpl -c security を使用して、テナントとセキュリティグループ定義をエクスポートします。各リージョナルマネージャーで、nnmconfigimport.ovpl を使用してテナントとセキュリティグループ定義をインポートします。あるいは、nnmsecurity.ovpl コマンドを使用して、別の NNMi 管理サーバーの UUID と同じ UUID を使用して、テナントおよびセキュリティグループを作成することができます。この推奨手順に従うことで、GNM 環境内で、各テナントとセキュリティグループの UUID を同じにすることができます。

▶ ユーザーがグローバルマネージャーから NPS レポートを開始する場合、このベストプラクティスは設定の必須部分になります。

▶ テナント UUID は一意である必要がありますが、テナント名は再利用できます。NNMi は、名前が同じで UUID が異なる 2 つのテナントを、共有設定を持たない 2 つの別個のテナントであると見なします。

### ベストプラクティス

組織ごとに 1 つのリージョナルマネージャーをセットアップする場合は、リージョナルマネージャーのすべてのノードを 1 つのテナントに入れることができます。ただし、各リージョナルマネージャーに一意のテナントを設定し、グローバルマネージャーでトポロジデータが確実に分離されるようにしてください。

リージョナルマネージャーからグローバルマネージャーに転送されたインシデントに、セキュリティ情報とテナント情報を伝達するいく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以下の 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以下の CIA が含まれます。

- `cia.securityGroup.name`
- `cia.securityGroup.uuid`

この項では以下の内容について説明します。

- 「初期 GNM 設定」(231 ページ)
- 「GNM のメンテナンス」(232 ページ)

## 初期 GNM 設定

GNM の初期設定後、リージョナルマネージャーは、(GNM 設定に従って) リージョナルトポロジ内のノードに関する情報を使用して、グローバルマネージャーを更新します。

### デフォルトテナントのみ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 GNM 環境の場合、グローバルマネージャーでは、リモートで管理されているすべてのノードが、以下の設定でグローバルマネージャートポロジに追加されます。

- デフォルトテナント
- デフォルトテナントの [初期検出セキュリティグループ] として設定されるセキュリティグループ。

### カスタムテナントとのトポロジの同期

カスタムセキュリティグループとカスタムテナントを持つ GNM 環境の場合、グローバルマネージャーでは、リモートで管理されているすべてのノードが、そのノードに割り当てられているテナントの **UUID** を使用して、グローバルマネージャートポロジに追加されます。そのテナント **UUID** がグローバルマネージャーにない場合、以下のように、GNM プロセスによってグローバルマネージャーの **NNMi** 設定にテナントが作成されます。

- このテナント **UUID** は、リージョナルマネージャーの場合と同じ値です。
- テナント名は、リージョナルマネージャーの場合と同じ値です。
- [初期検出セキュリティグループ] の値は、テナントと同じ名前のセキュリティグループに設定されます (このセキュリティグループがグローバルマネージャーにない場合、**NNMi** によってそのセキュリティグループが作成されます)。

グローバルマネージャーのトポロジにノードが追加されると、そのノードは、グローバルマネージャーに設定されたテナント **UUID** に対応する [初期検出セキュリティグループ] に割り当てられます。このため、グローバルマネージャー上でのセキュリティグループの関連付けは、リージョナルマネージャー上でのセキュリティグループの関連付けから独立しています。

### ベストプラクティス

グローバルマネージャーでのセキュリティ設定を簡素化するために、以下をお勧めします。

- 各リージョナルマネージャーによって管理されるノードのスプレッドシートまたはその他のレコードを保持します。ノードごとに、リージョナルマネージャーとグローバルマネージャーのそれぞれに必要なセキュリティグループをメモしておきます。GNM 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行います。
- GNM 環境で、複数のリージョナルマネージャーによって 1 つのグローバルマネージャーが更新されている場合、そのグローバルマネージャーに対して GNM 設定を有効にするには、各リージョナルマネージャーから 1 つずつ設定を行ってください。

該当する場合は、各リージョナルマネージャーを GNM 設定に追加する前に、デフォルトテナント（またはカスタムテナント）の [初期検出セキュリティグループ] の値を変更できます。これを実行した場合、以前に設定されたリージョナルマネージャーのトポロジに新しいノードが追加されると、さまざまな結果が生じる可能性があることに注意してください。

- GNM を有効にする前に、グローバルマネージャー上で、リージョナルマネージャーで使用される各テナントの [初期検出セキュリティグループ] を、オペレーター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。これにより、グローバルマネージャー上の管理者は、ほかの NNMi コンソールオペレーターのために、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。

## GNM のメンテナンス

表 22 は、リージョナルマネージャーでのノードのテナントまたはセキュリティグループの割り当てへの変更が、グローバルマネージャーにどのように影響を及ぼすかを示しています。

**表 22** リージョナルマネージャーでの設定変更がグローバルマネージャーに及ぼす影響

アクション	影響
リージョナルマネージャーで、ノードを別のテナントに割り当てる。	グローバルマネージャーのノードは、その別のテナントに割り当てられるように変更されます。このテナント UUID がグローバルマネージャーにない場合は作成されます。
リージョナルマネージャーで、ノードを別のセキュリティグループに割り当てる。	グローバルマネージャーでは変更は行われません。NNMi 管理者は、その変更を手動で複製するよう選択できます。
リージョナルマネージャーで、テナントの設定（名前、説明、または初期検出セキュリティグループ）を変更する。	グローバルマネージャーでは変更は行われません。NNMi 管理者は、その変更を手動で複製するよう選択できます。
リージョナルマネージャーで、セキュリティグループの設定（名前または説明）を変更する。	グローバルマネージャーでは変更は行われません。NNMi 管理者は、その変更を手動で複製するよう選択できます。

## NPS レポートへの選択インタフェースの追加

Network Performance Server (NPS) は、NNM iSPI Performance for Metrics ソフトウェアとともにインストールされるデータベースサーバーです。

デフォルトで、ノードのすべてのコンポーネントは、そのノードと同じセキュリティグループに属します。個々のインタフェースに対して、このデフォルトの動作をオーバーライドし、インタフェースを別のセキュリティグループに割り当てることができます。このオーバーライドは、共有デバイスのテナント（顧客）向けの適切なインタフェースを含む



テナント固有のレポートを生成するために行います。このようにすると、各顧客には、自分のインタフェースに関するインタフェース情報が表示され、デバイス上のほかのインタフェースは表示されないようになります。

- ▶ セキュリティグループのオーバーライドは、**NPS** レポートにのみ反映されます。**NNMi** コンソールでユーザーに表示される内容や、ユーザーが実行できる事柄には影響は及ぼされません。

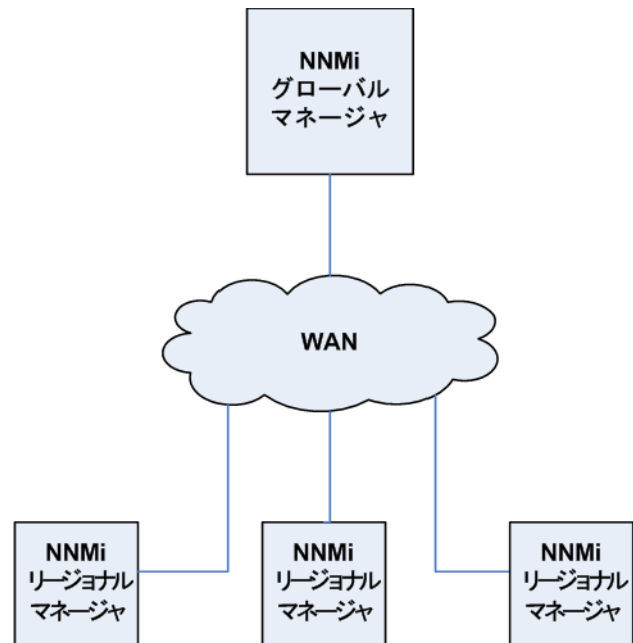
インタフェースのセキュリティグループ割り当てを変更するには、**[インタフェース]** フォームの **[カスタム属性]** タブ、または `nnmloadattributes.ovpl` コマンドを使用して、`InterfaceSecurityGroupOverride` カスタム属性をインタフェースに追加します。このカスタム属性の値をセキュリティグループの **UUID** に設定します。次に例を示します。

```
InterfaceSecurityGroupOverride=0826c95c-5ec8-4b8c-8998-301e0cf3c1c2
```

- ▶ インタフェースは、同時に **1** つのセキュリティグループにしか属することができません。インタフェースに `InterfaceSecurityGroupOverride` カスタム属性を設定すると、そのインタフェースと、ノードが属するセキュリティグループの間の関連付けが壊れます。



# グローバルネットワーク管理



この章には、以下のトピックがあります。

- グローバルネットワーク管理の利点
- グローバルネットワーク管理が自分のネットワークの管理に適しているかどうかを判断するには
- 実践的なグローバルネットワーク管理の例
- 要件のレビュー
- 初期準備
- グローバルネットワーク管理用にシングルサインオンを設定する
- リージョナルマネージャーでの転送フィルターの設定
- グローバルマネージャーとリージョナルマネージャーの接続
- global1 から regional1 と regional2 への接続ステータスの判定
- global1 インベントリの確認
- global1 と regional1 との通信の切断
- 追加情報
- グローバルネットワーク管理でアプリケーションフェイルオーバーの設定を行う
- 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
- グローバルネットワーク管理と NNM iSPI または第三者の統合
- 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

## グローバルネットワーク管理の利点

HP Network Node Manager i Software (NNMi) を地理的位置が異なる複数の NNMi 管理サーバーに導入しているとします。各 NNMi 管理サーバーでは、検出と監視のニーズに合うように、ネットワークの検出および監視を行っています。こうした既存の NNMi 管理サーバーと設定を使用して、特定の NNMi 管理サーバーをグローバルマネージャーとして指定することで、新たな検出を追加したりモニタリング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することができます。

NNMi グローバルネットワーク管理機能により、地理的位置が異なるネットワークを管理しながら、複数の NNMi 管理サーバーを連携させることができます。特定の NNMi 管理サーバーをグローバルマネージャーとして指定し、複数のリージョナルマネージャーを集約したノードオブジェクトデータを表示します。

NNMi グローバルネットワーク管理機能には、以下の利点があります。

- グローバルマネージャーから見た、企業のネットワークの全体像を表示できます。
- 以下のように容易に設定できます。
  - リージョナルマネージャーの管理者はそれぞれ、すべてのノードオブジェクトデータを指定するか、またはグローバルマネージャーレベルで参加する特定のノードグループを指定します。
  - 各グローバルマネージャーの管理者は、情報の提供を許可するリージョナルマネージャーを指定します。
- 各サーバーごとに、インシデントの生成と管理を行うことができます（各サーバーで使用可能なトポロジのコンテキスト内で生成されます）。

詳細については、NNMi ヘルプの「NNMi のグローバルネットワーク管理機能」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的なテナントに加え、NNMi リージョナルマネージャーが必要です。「[NAT 環境の重複 IP アドレスの管理](#)」(197 ページ) を参照してください。NNMi ヘルプも参照してください。

## グローバルネットワーク管理が自分のネットワークの管理に適しているかどうかを判断するには

以下の質問に答えることで、NNMi のグローバルネットワーク管理機能が自分のネットワーク管理に役立つかどうかを判断できます。

### マルチサイトネットワークを継続的に監視する必要がありますか？

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ますか？ 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、監視することができるようになります。

### 重要デバイスを表示できるか？

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ますか？ はい。リージョナルマネージャーに転送フィルターを設定します。このフィルターにより、リージョナルマネージャーからグローバルマネージャーに送信するノードオブジェクトデータを選択できます。たとえば、リージョナルマネージャーに対し転送フィルターを設定して、重要デバイスに関する情報のみをグローバルマネージャーに転送することができます。

## ライセンスの考慮事項

NNMi ライセンスキーの取得とインストールの詳細については、「[NNMi のライセンス](#)」(123 ページ)を参照してください。

グローバルマネージャー、リージョナルマネージャー両方について NNMi Advanced ライセンスが必要ですか？グローバルマネージャーとして使用する NNMi 管理サーバーには、NNMi Advanced ライセンスを購入してインストールする必要があります。NNMi 管理サーバーをリージョナルマネージャーとして使用する場合は、NNMi Advanced ライセンスは必要ありません。

1つの地域をカバーするのに十分な NNMi ライセンスを持っています。グローバルネットワーク管理機能を使用しながら、グローバルマネージャーに必要な新しいライセンスの数を抑えることはできますか？はい。IT グループが複数のサイトに配備された重要な装置をモニターする必要がある場合は、リージョナルマネージャーに転送フィルターを設定して、グローバルマネージャーに重要な装置に関する情報のみが転送されるようにすることができます。このようなフィルター設定を使用することで、既存のグローバルマネージャーのライセンスを最大限に活用し、NNMi への投資を無駄なく使用できます。

ライセンスを取得したノードの総数がグローバルマネージャーの NNMi Advanced ライセンスより多くなるように、リージョナルマネージャー用に NNMi ライセンスを増やしました。グローバルマネージャーには、すべての領域のすべてのノードの完全なインベントリがありません。十分なライセンスをグローバルマネージャー用に購入した後で、グローバルマネージャーをすべてのリージョナルマネージャーと同期させて、ライセンスが不十分だったために前回省略したノードを検索して作成するにはどうしたら良いでしょうか。グローバルマネージャーで十分な NNMi Advanced ライセンスを購入してインストールし、リージョナルマネージャーでインストールしたライセンス総数を上回るようにする必要があります。十分なライセンスをインストールしたら、以下のいずれかを実行します。

- すべてのリージョナルマネージャーで設定されている、すべての再検出間隔の時間が経過して、すべての領域ですべてのノードが再検出されるのを待機します。リージョナルマネージャーは、すべての領域ですべてのノードを再検出したら、再検出されたノードの情報をグローバルマネージャーに送信します。グローバルマネージャーはこのノード情報を受信し、各領域でノードごとにグローバルノードを作成します。
- 各リージョナルマネージャーで `nnmnode rediscover.ovpl -all` スクリプトを実行します。



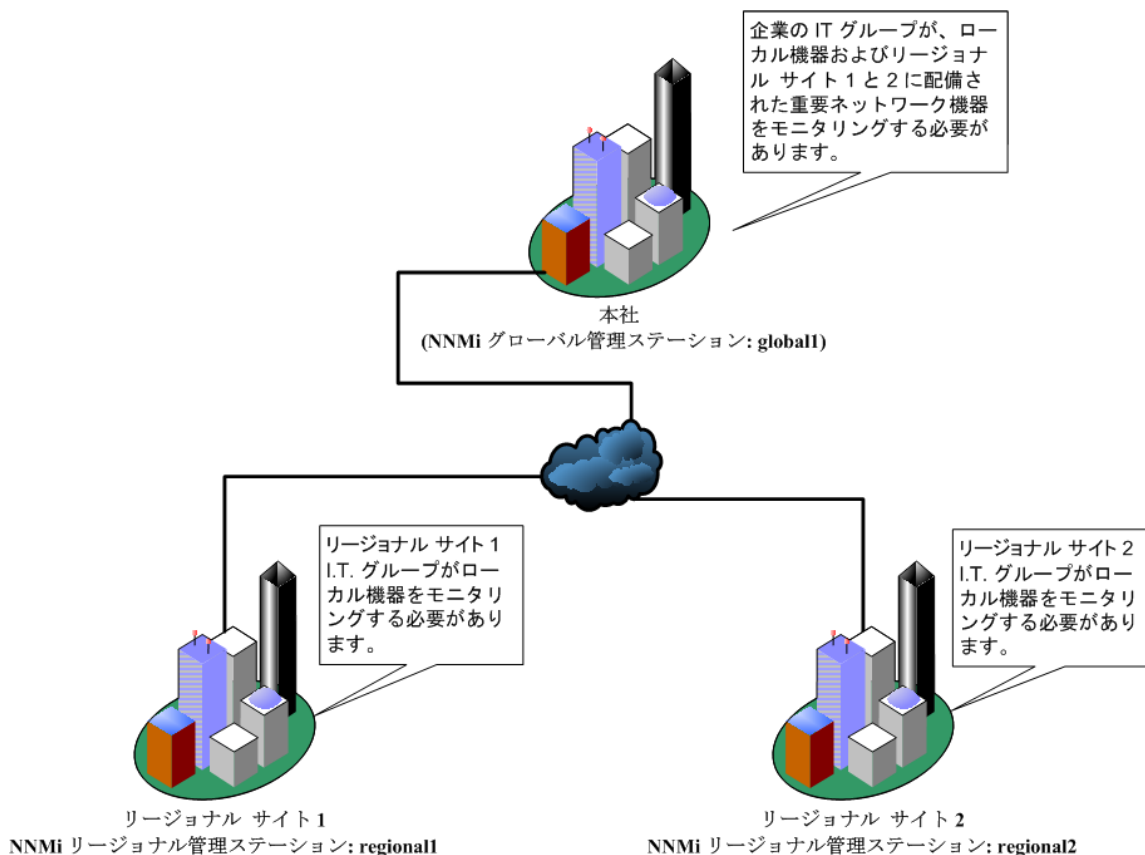
2番目のオプションでは、ネットワーク上のトラフィックが増加し、NNMi マネージャーのセット全体から多くの NNMi リソースが消費されることにもなります。このオプションは、最初の NNMi 検出ほどリソースの多くを消費しませんが、最初の検出を実行することに似ています。最適な方法では、ある程度の時間をおくか、現在のリージョナルマネージャーの負荷が減って正常になるのを待ち、領域ごとに間隔をおいてスクリプトを実行してから、次のリージョナルマネージャーの再検出を始めます。

## 実践的なグローバルネットワーク管理の例

238 ページの [図 21](#) 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 21 ネットワークの例



## 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 と見なします。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。



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してみる場合があります。このような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のみの監視を行い、ほかのデバイスの管理はしない予定です。以下の表に、監視のニーズをまとめます。

**表 23 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。

## リージョナルマネージャーとグローバルマネージャーの接続

グローバルネットワーク管理接続を設定するときに、以下の情報を考慮します。

- グローバルマネージャーとすべてのリージョナルマネージャーで、同じ NNMi バージョンおよびパッチレベルを使用します。異なる NNMi バージョンを使用したグローバルネットワーク管理設定はサポートされていません。
- NNMi では、リージョナルマネージャーと通信する 1 つ以上のグローバルマネージャーを設定できます。たとえば、regional1 と通信するために第 2 のグローバルマネージャー、global2 が必要な場合、NNMi では、regional1 と通信する global1 と global2 の両方を設定できます。詳細については、『HP Network Node Manager i Software システムとデバイス対応マトリックス』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。たとえば、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。たとえば、global1 は regional1 と通信する設定にはせず、regional1 は regional2 と通信する設定にします。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す。
- 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。たとえば、global1 は regional1 と通信する設定にはせず、regional1 は global1 と通信する設定にします。

## 初期準備

### 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 と 443 を設定します。ただし、インストール中にこれらの値は変更できます。



このセクションで説明した例では、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、以下のファイルを編集します。

- Windows: %NNM\_CONF%\nmm\props\nms-local.properties
- UNIX: \$NNM\_CONF/nnm/props/nms-local.properties

以下の表に、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 24** アクセス可能にしておく必要があるソケット

セキュリティ	パラメーター	TCP ポート
非 SSL	nmsas.server.port.web.http	80
	nmsas.server.port.hq	4457
SSL	nmsas.server.port.web.https	443
	nmsas.server.port.hq.ssl	4459

詳細については、[NNMi 9.20](#) およびウェルノウンポートを参照してください。

### 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、他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。「[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](#)」(137 ページ) に示した手順を実行してください。

### グローバルネットワーク管理でアプリケーションフェイルオーバーの設定を行う

NNMi のインストール中、NNMi インストールスクリプトでは、他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。グローバルネットワーク管理機能とともにアプリケーションフェイルオーバーを使用する場合は、追加の設定を行う必要があります。「[自己署名証明書を使用するようにアプリケーションフェイルオーバーが有効なグローバルネットワーク管理を設定する](#)」(139 ページ) に示した手順を実行してください。



## 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ー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、**HP Network Node Manager i Software インタラクティブインストールガイド**、**NNMi リリースノート**、**and the NNMi システムおよびデバイス対応マトリックス**を参照してください。

## システムクロックの同期化

global1、regional1、および regional1 サーバーをグローバルネットワーク管理設定に接続する前に、これらの NNMi 管理サーバークロックを同期化することが重要です。グローバルネットワーク管理（グローバルマネージャーとリージョナルマネージャー）やシングルサインオン（SSO）に属するネットワーク環境内のすべての NNMi 管理サーバーは、それぞれの内部タイムクロックを世界標準時で同期化する必要があります。たとえば、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「クロック同期の問題」または「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」と「**クロック同期**」（268 ページ）を参照してください。



サーバークロック同期の問題など、リージョナルマネージャーとの接続に問題がある場合、NNMi では NNMi コンソールの下部に警告メッセージが表示されます。

## 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。「自己署名証明書を使用するようにアプリケーションフェイルオーバーが有効なグローバルネットワーク管理を設定する」（139 ページ）を参照してください。

## グローバルネットワーク管理における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。「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」（137 ページ）を参照してください。

## グローバルネットワーク管理における認証機関の使用法

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。「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」（138 ページ）を参照してください。

## 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ー（これについては後で説明します）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、以下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す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。

## グローバルマネージャーとリージョナルマネージャーの管理ドメインの検討

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、後で 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。

以下の手順に従って、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、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、以下の手順を実行してこの情報を確認します。

- 1 ブラウザーで global1 の NNMi コンソールを指します。
- 2 サインインします。
- 3 **[インベントリ]** ワークスペースをクリックします。
- 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ます。
- 5 ブラウザーで regional1 の NNMi コンソールを指します。
- 6 サインインします。
- 7 **[インベントリ]** ワークスペースをクリックします。
- 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します。
- 9 ブラウザーで regional2 の NNMi コンソールを指します。
- 10 サインインします。
- 11 **[インベントリ]** ワークスペースをクリックします。
- 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します。

## 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、以下の手順を実行します。

- 1 NNMi ヘルプで、[ **検索** ] をクリックします。
- 2 [ 検索 ] フィールドに [ **グローバルネットワーク管理** ] と入力します。
- 3 [ **検索** ] をクリックします。

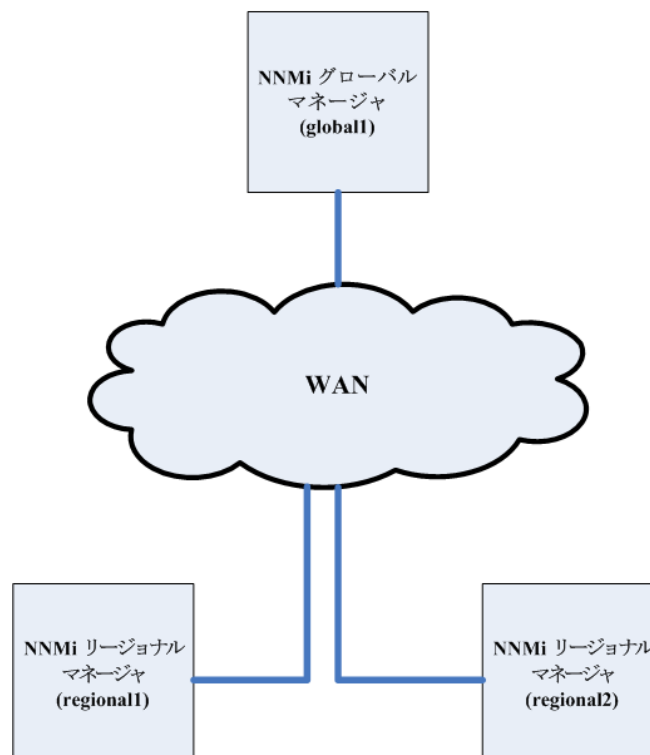
この検索により、グローバルネットワーク管理に関連する50以上のトピックが見つかります。

## SSO およびアクションメニュー

グローバルマネージャーの NNMi コンソールから、リージョナルマネージャーが管理するノードを選択した後に、[ **アクション** ] メニューを使用して、選択したノードに対するアクションを開始するとします。NNMi 管理サーバーの間で `initString` と `domain` のパラメーターを同一にしないと、グローバルマネージャーのセッション情報は新しいセッションに渡されず、アクションは開始されません。この問題を回避するには、「**グローバルネットワーク管理用にシングルサインオンを設定する**」(243 ページ) の設定手順に従ってください。

# グローバルネットワーク管理用にシングルサインオンを設定する

NNMi シングルサインオン (SSO) を設定すると、NNMi グローバルマネージャーから簡単に NNMi リージョナルマネージャーにアクセスできるようになります。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に接続する前に、この手順を完了しておく必要があります。詳細については、「**NNMi とシングルサインオンの使用**」(143 ページ) を参照してください。



SSO 機能は、NNMi 管理サーバー内のユーザー名を交換しますが、パスワードやロールは交換しません。たとえば、NNMi は 1 つの NNMi 管理サーバー (global1) の特定のユーザー名を、別の NNMi 管理サーバー (regional1 または regional2) の異なるロールに関連付けます。3 つの NNMi 管理サーバーで、同じユーザー名に異なるパスワードが関連付けられることもあります。

グローバルマネージャーとリージョナルマネージャーが同じ管理ドメインにあり、244 ページの **手順 4** に示したように **Initialization String** 値をグローバル NNMi 管理サーバーからリージョナル NNMi 管理サーバーにコピーしないと、NNMi コンソールのアクセスに問題が起こる場合があります。これを回避するには、以下の手順に従って SSO を正しく設定するか、「SSO の無効化」(148 ページ) で説明したように SSO を無効にします。

SSO をグローバルネットワーク管理機能と連携させるには、以下の手順を実行します。

- 1 global1、regional1、および regional2 で以下のファイルを編集します。
  - **Windows:** %NNM\_PROPS%\nms-ui.properties
  - **UNIX:** \$NNM\_PROPS/nms-ui.properties
- 2 global1、regional1、および regional2 ファイルで、以下のようなセクションを探します。
 

```
com.hp.nms.ui.sso.isEnabled = false
```

 これを以下のように変更します。
 

```
com.hp.nms.ui.sso.isEnabled = true
```
- 3 global1 の SSO NNMi 初期化文字列を探します。nms-ui.properties ファイルから、以下のようなセクションを特定します。
 

```
com.hp.nms.ui.sso.initString = Initialization String
```
- 4 global1 の nms-ui.properties ファイルにある **Initialization String** の値を、regional1 と regional2 の nms-ui.properties ファイルにコピーします。**Initialization String** は、すべてのサーバーで同じ値を使用する必要があります。変更を保存します。
 

▶ グローバル NNMi 管理サーバーからリージョナル NNMi 管理サーバーへの **Initialization String** 値のコピーは NNMi でサポートされます。この操作により、グローバルマネージャーから 2 つのリージョナルマネージャーに **Initialization String** 値がコピーされます。グローバルネットワーク管理機能で SSO を使用する場合は、**Initialization String** 値のコピーは、常に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に対して行ってください。

▶ グローバルマネージャーとリージョナルマネージャーが同じ管理ドメインにあり、**Initialization String** 値をグローバル NNMi 管理サーバーからリージョナル NNMi 管理サーバーにコピーしない場合は、SSO を無効にして、NNMi コンソールのアクセスに問題が起こらないようにします。詳細については、「SSO の無効化」(148 ページ) を参照してください。
- 5 global1、regional1、および regional2 が異なるドメインにある場合は、protectedDomains の内容を変更します。変更するには、nms-ui.properties ファイルの中から以下のようなセクションを探します。
 

```
com.hp.nms.ui.sso.protectedDomains=group1.mycompany.com
```

global1はglobal1.company1.comに、regional1はregional1.company2.comに、そして、regional2はregional2.company3.comにあるとします。global1、regional1、regional2にあるnms-ui.propertiesファイルのprotectedDomainsセクションを以下のように変更します。

```
com.hp.nms.ui.sso.protectedDomains=regional1.company1.com,
regional2.company2.com,regional3.company3.com
```

- 6 変更を保存します。
- 7 global1、regional1、regional2で、以下の一連のコマンドを実行します。
  - a ovstop
  - b ovstart



アプリケーションフェイルオーバー設定でシングルサインオンを有効にするときに、手動で行う設定手順はありません。たとえば、アプリケーションフェイルオーバー設定でシングルサインオンを設定する場合、NNMiによりアクティブ NNMi 管理サーバーからスタンバイ NNMi 管理サーバーに上記の変更を複製されます。

---

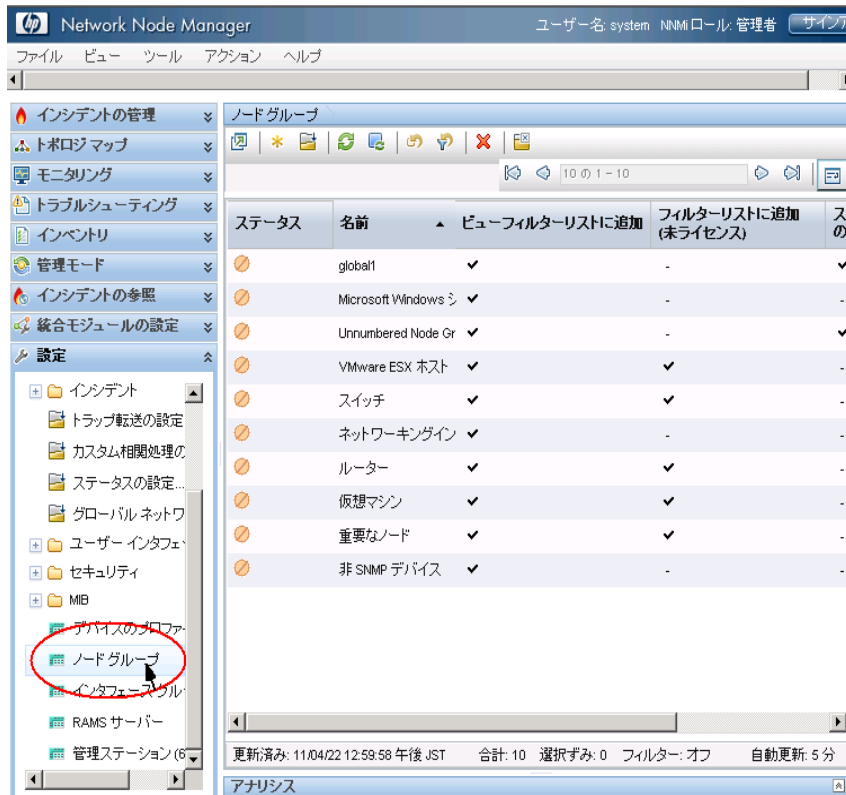
## リージョナルマネージャーでの転送フィルターの設定

この例では、global1はregional1とregional2の両方と通信します。グローバルマネージャーglobal1がリージョナルマネージャーregional1とregional2から受け取るノードオブジェクトデータを制御するには、regional1とregional2の両方で転送フィルターを設定する必要があります。

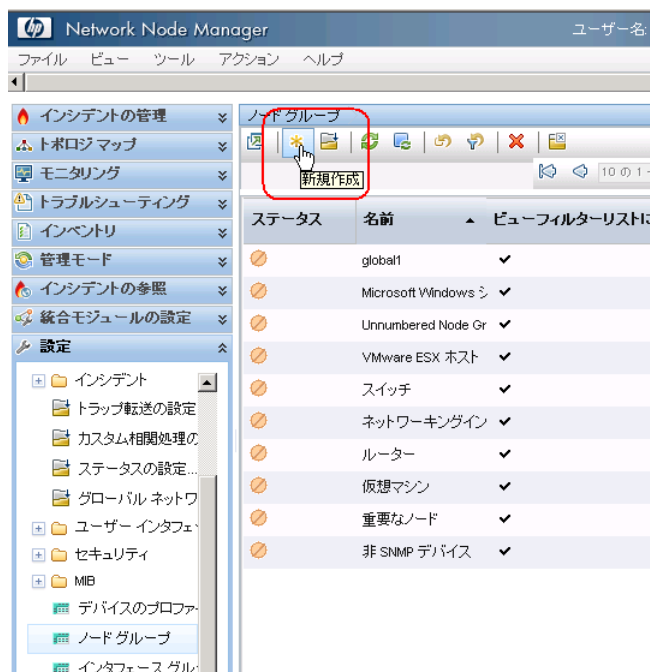
### 転送されるノードを制限する転送フィルターの設定

ノードグループを設定し、regional1からModel 3500yl ProCurve Switchのノード情報のみを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、以下の手順を実行します。

- 1 NNMi コンソールのregional1の[設定]ワークスペースから[ノードグループ]をクリックします。



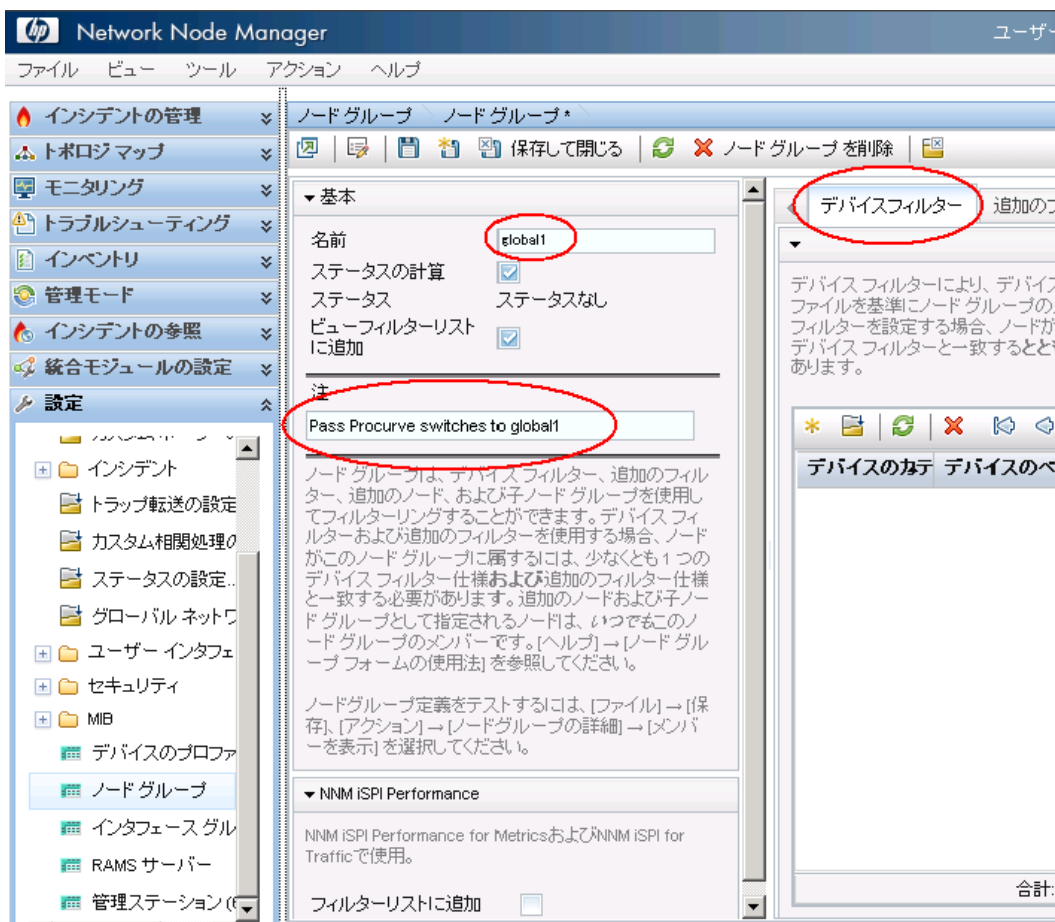
2 [新規作成] をクリックします。



この例では、ノードフィルターの新規作成し、そのフィルターを使用して **regional1** と **regional2** の転送フィルターを作成する方法を説明していますが、既存のフィルターを使用して、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ーを設定することもできます。

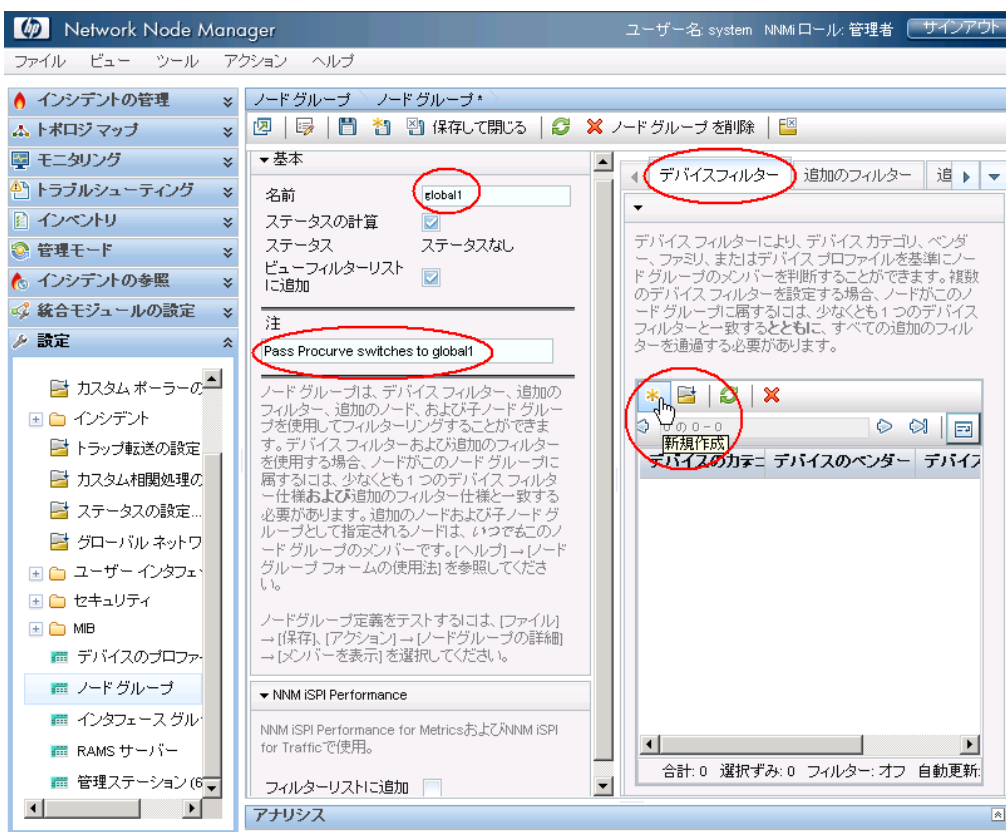
独自のデバイスもフィルターも含まれていないコンテナノードグループを作成して、このノードグループを使用して子ノードグループを指定できます。この方法を使用すると、1つのコンテナノードグループを使用して、ノードオブジェクトデータをグローバル NNMi 管理サーバーに転送できます。

- 3 [ **デバイスフィルター** ] タブをクリックします。フィルター名に `global1` と入力し、[ 注 ] フィールドに作成するフィルターの説明を入力します。

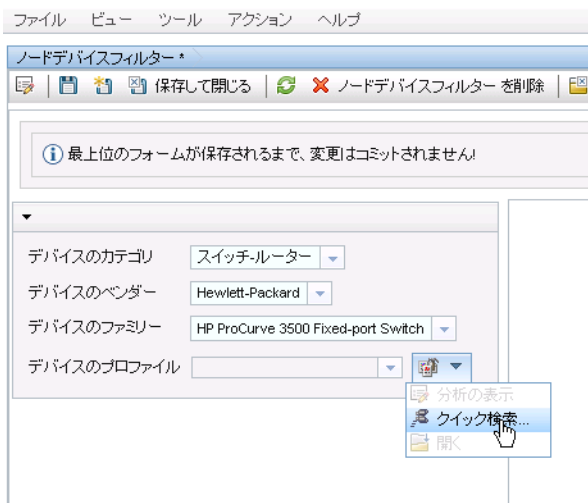




- 4 [新規作成] アイコンをクリックして、[ ノードデバイスフィルター ] フォームを開きます。



- 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 ] を選択します。
- 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 ] フォームを開きます。





- 7 3500yl HP ProCurve Switch のプロファイルを検索して選択し、[OK] をクリックします。

デバイスのモデル	SNMP のオブジェクトID	OUI	デバイスのファミリー	デバイスのベンダー
hpProCurveRWESMzl	.1.3.6.1.4.1.11.2.3.7.11.50.7	416T	HP ProCurve Wirel	Hewlett-Pe
hpProCurveONEService	.1.3.6.1.4.1.11.2.3.7.11.50.7	0NE	HP ProCurve ONE	Hewlett-Pe
hpProCurve5412zl	.1.3.6.1.4.1.11.2.3.7.11.51	540	HP ProCurve 5400	Hewlett-Pe
hpProCurve4204vl	.1.3.6.1.4.1.11.2.3.7.11.52	420	HP ProCurve 4200	Hewlett-Pe
hpProCurve4208vl	.1.3.6.1.4.1.11.2.3.7.11.53	420	HP ProCurve 4200	Hewlett-Pe
hpProCurve9400_9408s	.1.3.6.1.4.1.11.2.3.7.11.54	940	HP ProCurve 9400	Hewlett-Pe
hpProCurve2608-PWR	.1.3.6.1.4.1.11.2.3.7.11.55	260	HP ProCurve 2600	Hewlett-Pe
hpProCurve4202vl-48G	.1.3.6.1.4.1.11.2.3.7.11.56	420	HP ProCurve 4200	Hewlett-Pe
hpProCurve4202vl-72	.1.3.6.1.4.1.11.2.3.7.11.57	420	HP ProCurve 4200	Hewlett-Pe
hpProCurve3500yl-PWR	.1.3.6.1.4.1.11.2.3.7.11.58	350	HP ProCurve 3500	Hewlett-Pe
hpProCurve3500yl-48G	.1.3.6.1.4.1.11.2.3.7.11.59	350	HP ProCurve 3500	Hewlett-Pa
hpProCurve224	.1.3.6.1.4.1.11.2.3.7.11.6	200	HP ProCurve 200 f	Hewlett-Pe
hpProCurve6200yl-24G	.1.3.6.1.4.1.11.2.3.7.11.60	620	HP ProCurve 6200	Hewlett-Pe
hpProCurve2510-24	.1.3.6.1.4.1.11.2.3.7.11.61	250	HP ProCurve 2500	Hewlett-Pe
hpProCurve2510-48	.1.3.6.1.4.1.11.2.3.7.11.62	250	HP ProCurve 2500	Hewlett-Pe
hpProCurve2810-24G	.1.3.6.1.4.1.11.2.3.7.11.63	260	HP ProCurve 2600	Hewlett-Pe

- 8 [保存して閉じる] を 2 回クリックします。

ファイル ビュー ツール アクション ヘルプ

ノードデバイスフィルター

保存して閉じる

ノードデバイスフィルター を削除

最上位のフォームが保存されるまで、変更はコミットされません!

デバイスのカテゴリ: スイッチルーター

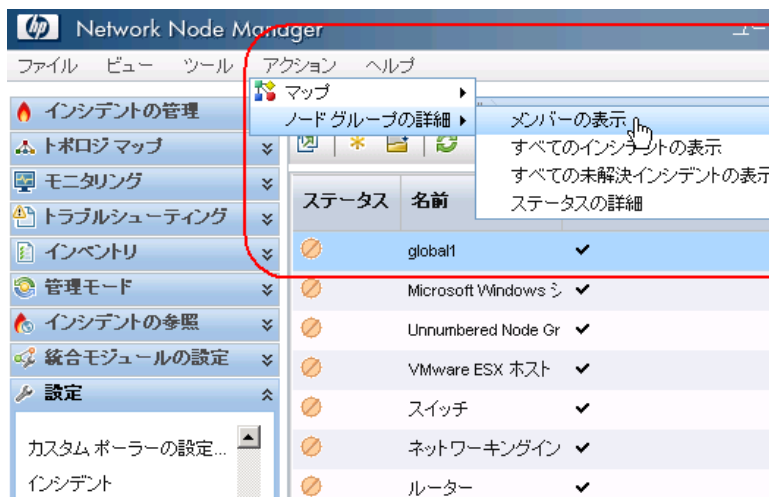
デバイスのベンダー: Hewlett-Packard

デバイスのファミリー: HP ProCurve 3500 Fixed-port Switch

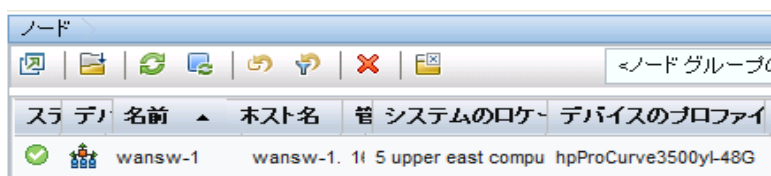
デバイスのプロファイル: hpProCurve 2810-48G

- 9 このフィルターをテストするため、[global1] を選択します。

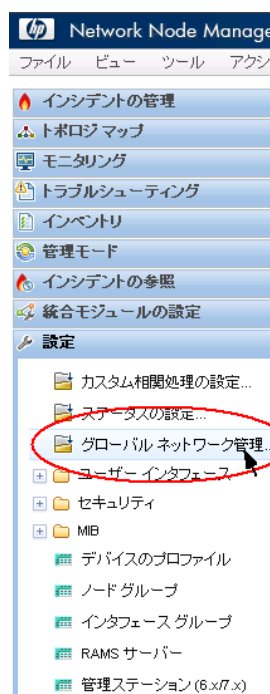
- 10 プルダウンメニューから、[メンバーの表示]をクリックします。



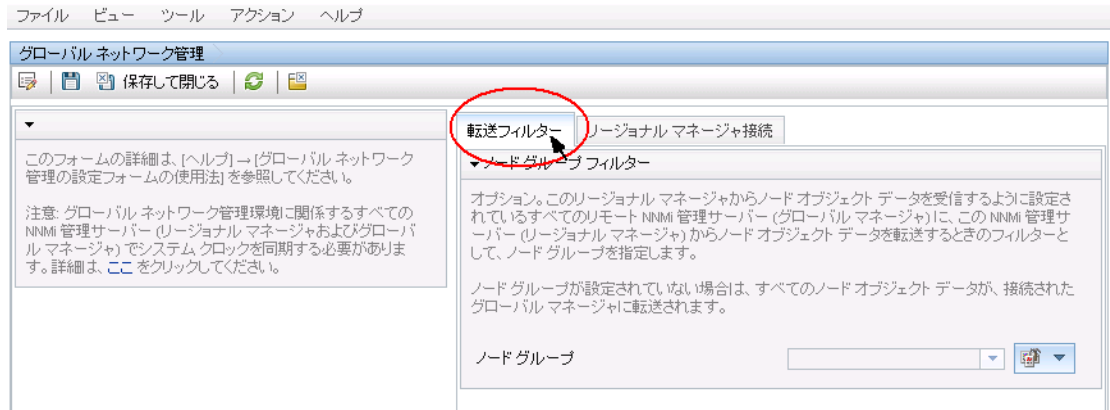
- 11 NNMi ではすでに HP 3500yl スイッチが 1 つ検出されています。これは、作成したフィルターが、設定した特定のスイッチモデルを検索していることを示しています。次のステップでは、今作成したこのノードフィルターを使用して転送フィルターを設定します。



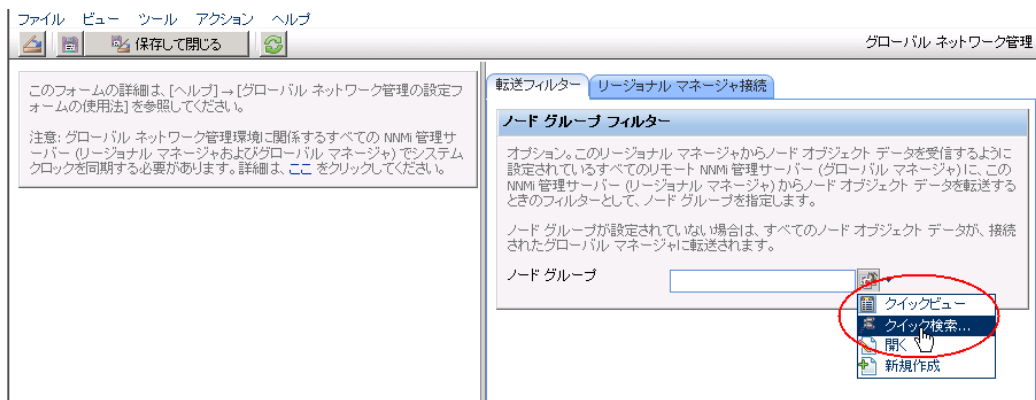
- 12 NNMi コンソールの regional11 の [設定] ワークスペースから [グローバルネットワーク管理] をクリックします。



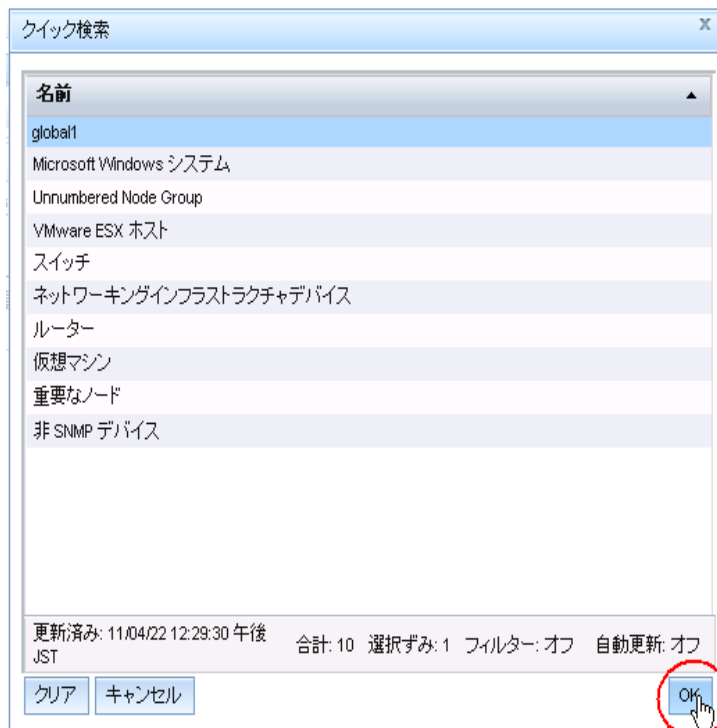
13 [転送フィルター] タブをクリックします。



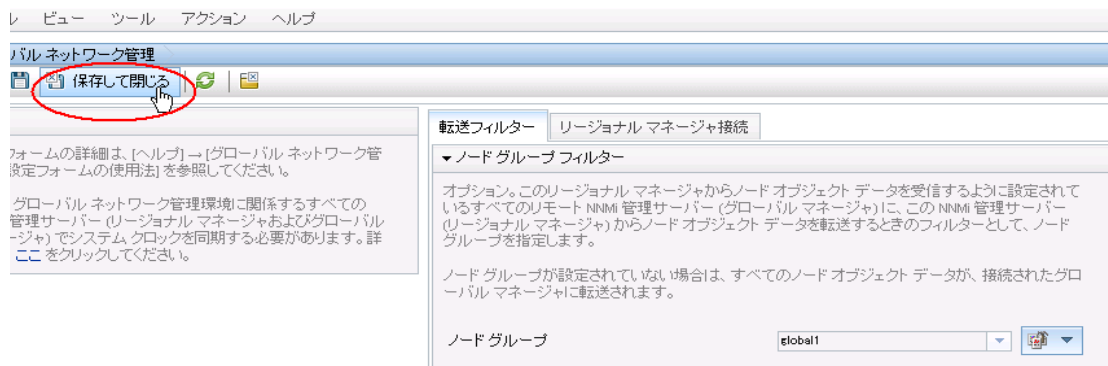
14 [クイック検索] をクリックします。



15 global1 フィルターを選択し、[OK] をクリックします。



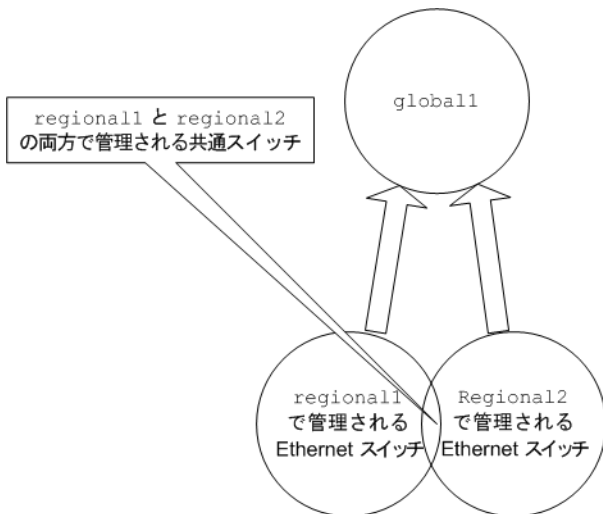
16 [保存して閉じる] をクリックします。



これで、`regional1` の転送フィルターの設定作業は完了です。`regional2` についても手順 1 から手順 16 を実行したら、次のセクションに進み、`global1` を `regional1` と `regional2` に接続します。

## グローバルマネージャとリージョナルマネージャの接続

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。この接続順により、`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みなします。`Global1` は、また、`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。



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、以下の手順を実行します。

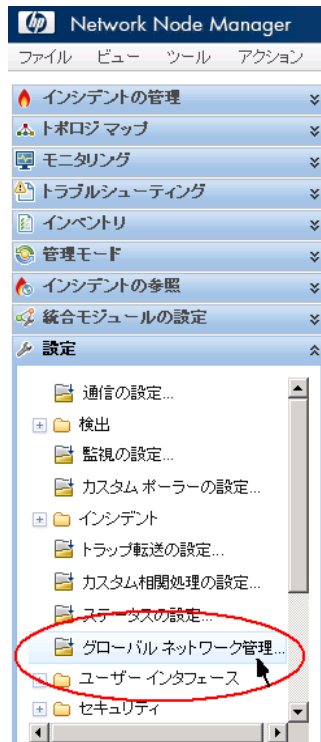
- 1 すでに述べたように、NNMi 管理サーバーのクロックを global1、regional1、および regional2 と同期化してから、グローバルネットワーク管理設定内のこれらのサーバーを接続します。詳細については、NNMi ヘルプの「クロック同期の問題」を参照してください。



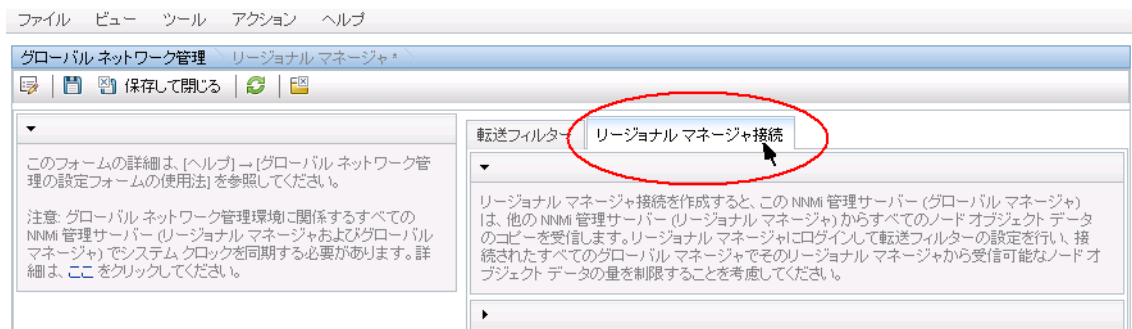
サーバークロック同期の問題など、リージョナルマネージャーとの接続に問題がある場合は、NNMi では警告メッセージが表示されます。

- 2 global1 から regional1 への接続を設定します。

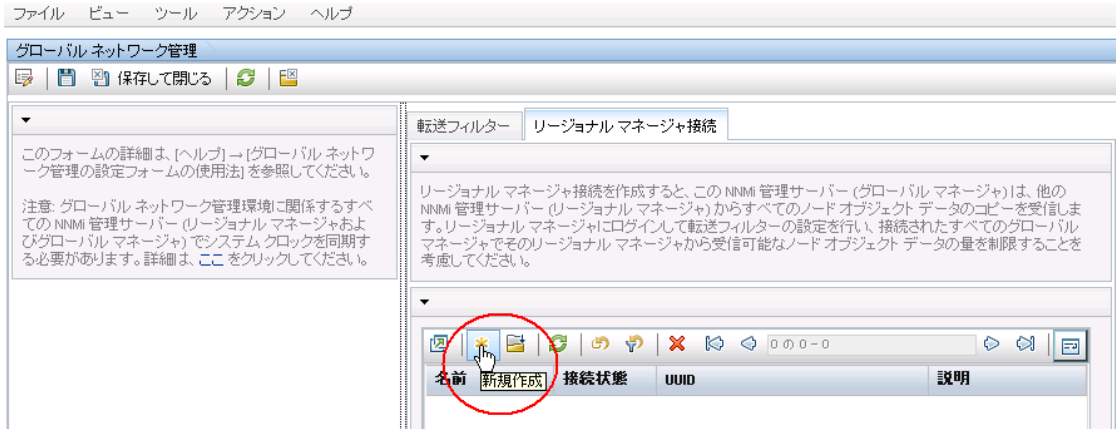
- a global1 の NNMi コンソールで、[設定] ワークスペースの [グローバルネットワーク管理] をクリックします。



- b [リージョナルマネージャ接続] をクリックします。



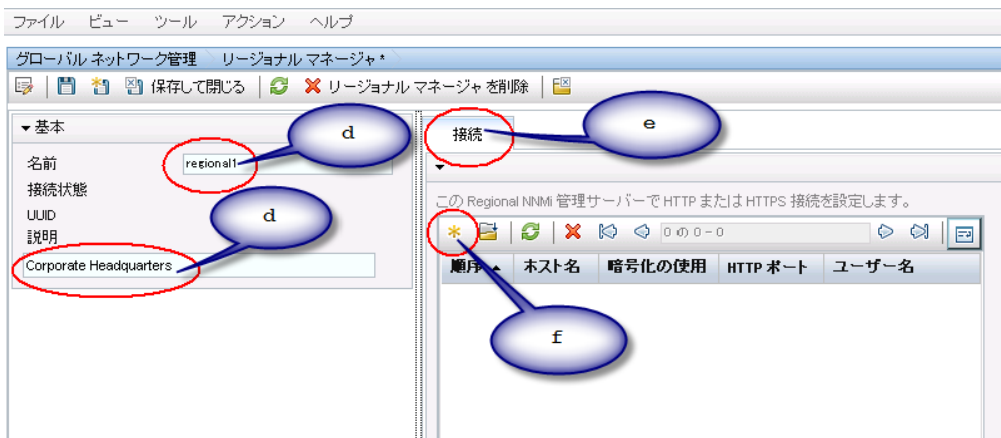
c [新規作成]アイコンをクリックして、リージョナルマネージャーを新規作成します。



d regional1 の名前と説明情報を追加します。

e [接続]タブをクリックします。

f [新規作成]アイコンをクリックします。



g regional1 の接続情報を追加します。

- ▶ このフォームで作成するエントリーに関する個別の情報については、NNMi ヘルプの「[ヘルプ]->[リージョナルマネージャ フォームの使用法]」を参照してください。

リージョナル マネージャの接続

保存して閉じる リージョナル マネージャの接続 を削除

① 最上位のフォームが保存されるまで、変更はコミットされません

▼

リモートリージョナル マネージャ サーバーのホスト名に完全修飾ドメイン名を指定します。詳細は、[ヘルプ] -> [リージョナル マネージャ接続フォームの使用法] を参照してください。

ホスト名 regional1.example.hp.com

暗号化の使用

HTTP ポート 80

ユーザー名 system

ユーザー パスワード

順序 20

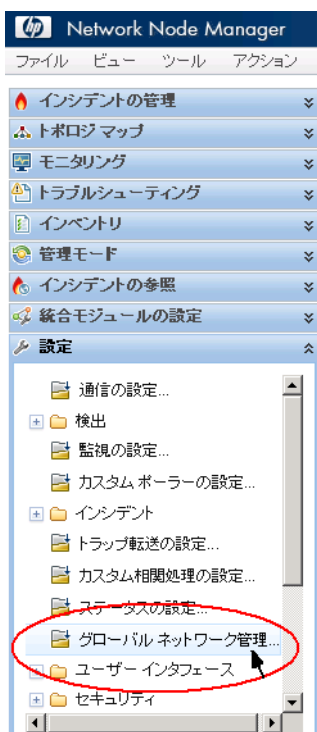
h [保存して閉じる] を 2 回クリックして作業を保存します。

- 3 global1 から regional2 への接続を確立するため、253 ページの**手順 a** から 255 ページの**手順 g** までを実行します。

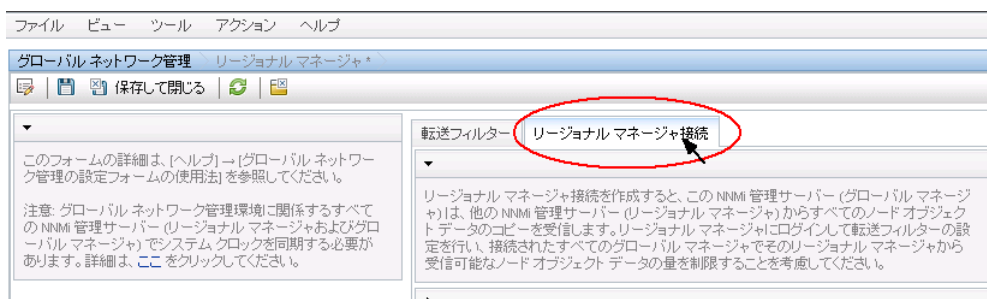
## global1 から regional1 と regional2 への接続ステータスの判定

global1 から regional1 および regional2 への接続の状態を確認するには、以下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 ] をクリックします。



- 2 [ リージョナルマネージャ接続 ] タブをクリックします。





- 3 regional1 と regional2 の接続ステータスを確認します。[Connected] と表示され、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。

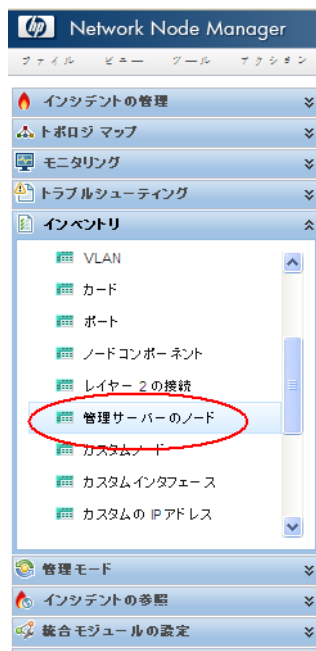
NNMi が検出を完了するまで、次のセクションには進まないでください。詳細については、『HP Network Node Manager i Software インタラクティブインストールガイド』の「検出の進行状況の確認」を参照してください。

## global1 インベントリの確認

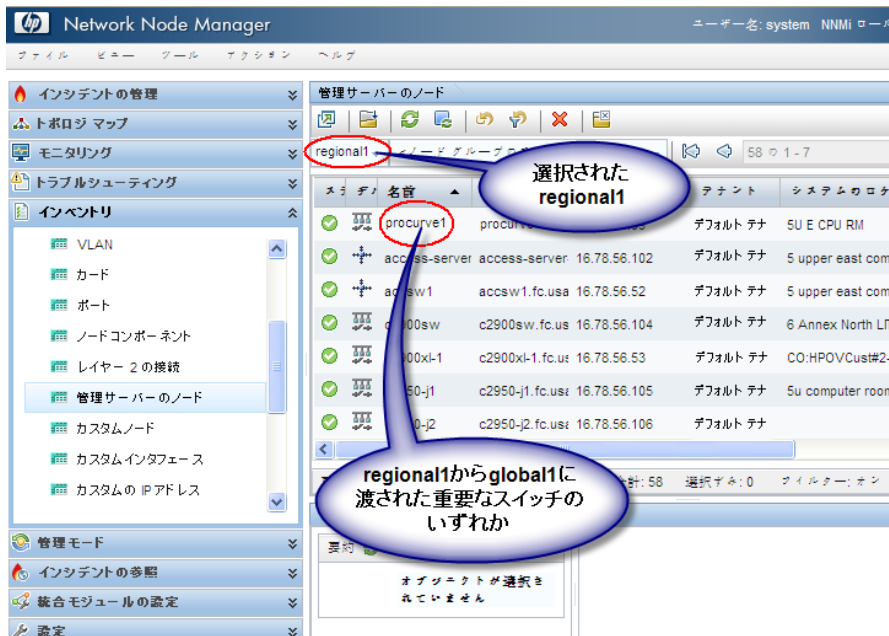
NNMi が検出を完了するまで、このセクションは実行しないでください。詳細については、『HP Network Node Manager i Software インタラクティブインストールガイド』の「検出の進行状況の確認」を参照してください。

global1に転送されるノード情報regional1を表示するには、以下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 [管理サーバーごとのノード] フォームに、global1 の NNMi コンソールから移動します。



- 2 スイッチ `procurve1.x.y.z` に関する情報が `regional1` から `global1` に転送されたと仮定します。**regional1** を選択すると、インベントリは以下のように表示されます。

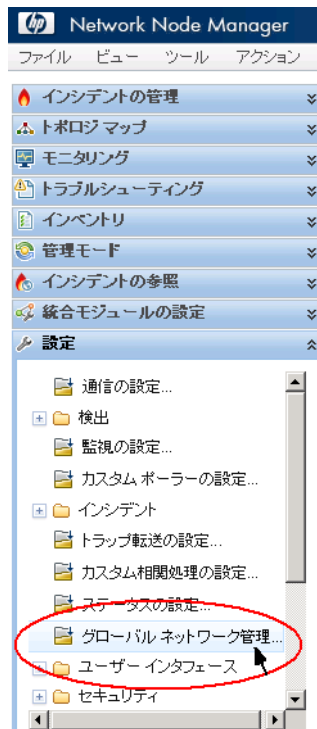


手順1から手順2を実行して、接続されている他のリージョナルマネージャーから `global1` に転送されたデバイスインベントリも表示します。

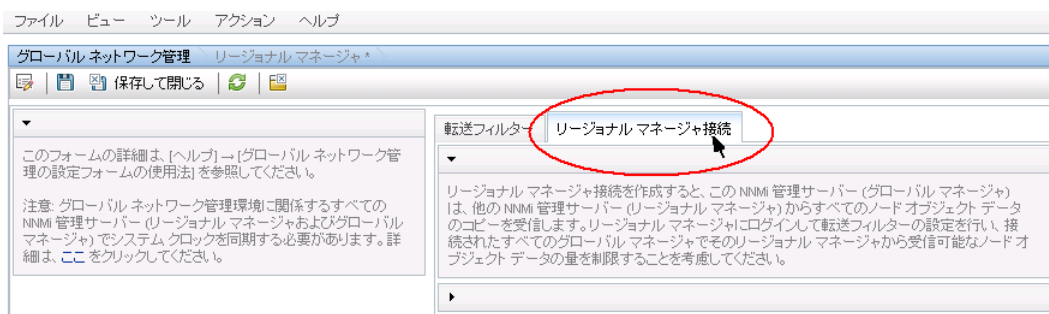
## global1 と regional1 との通信の切断

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 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。

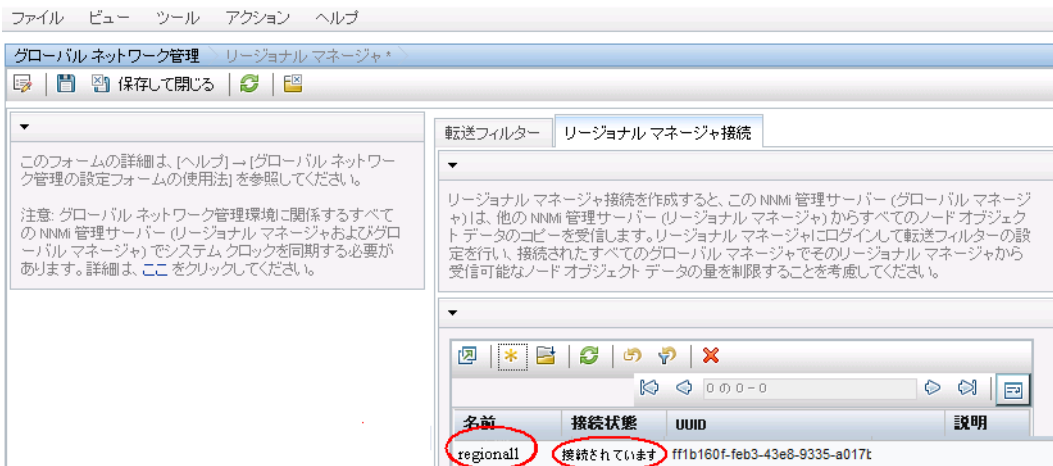
- 1 global1 の NNMI コンソールで、[ 設定 ] ワークスペースの [ グローバルネットワーク管理 ] をクリックします。



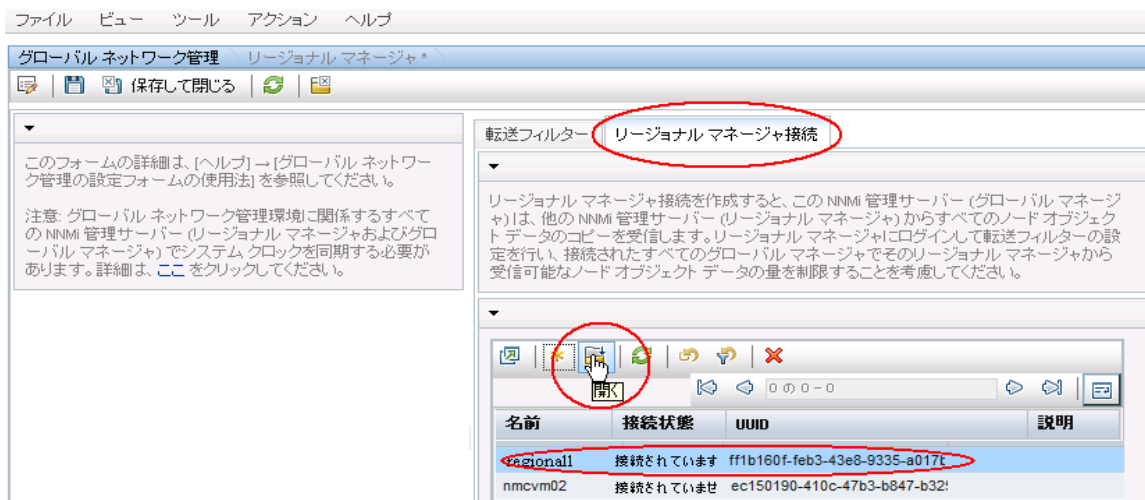
- 2 [ リージョナルマネージャ接続 ] をクリックします。



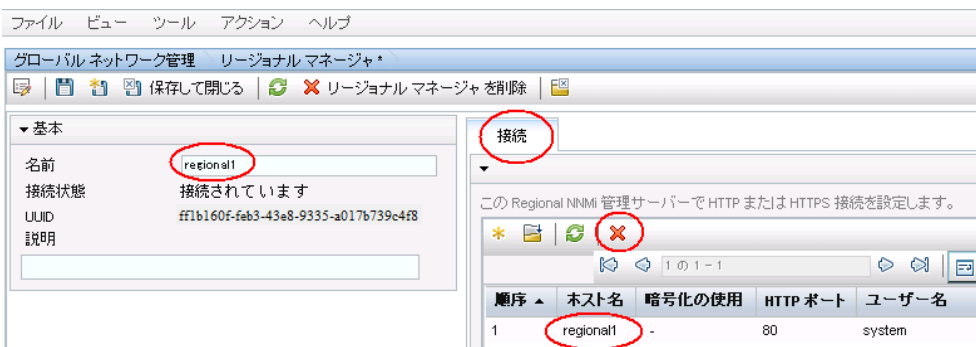
- 3 ステータスが [ 接続されています ] であることを確認します。ステータスが [ 接続されています ] ではない場合、処理を続行する前に、NNMi ヘルプの「グローバルネットワーク管理のトラブルシューティング」を参照して問題を診断します。



- 4 regional1 を選択して [ 開く ] アイコンをクリックします。



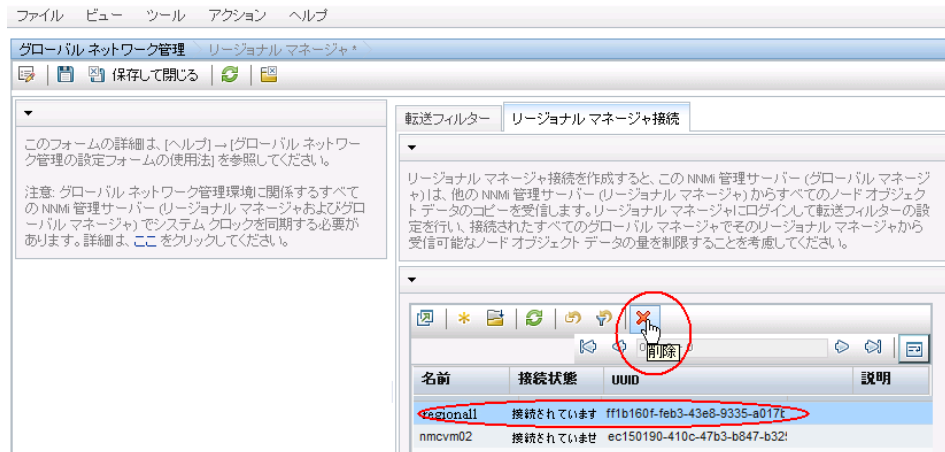
- 5 [ 接続 ] をクリックして [ regional1.x.y.z ] を選択してから [ 削除 ] をクリックします。



- 6 [ 保存して閉じる ] をクリックします。
- 7 [ リージョナルマネージャ接続 ] タブでは、regional1 の [ 名前 ] 属性に注意してください (大文字小文字は区別されます)。後のステップで、RemoteNNMiServerName 変数にこのテキスト文字列が必要になります。

- 8 [保存して閉じる]をもう一度クリックします。
- 9 global1 で、コマンドラインで以下のコマンドを入力します。  

```
nnmnodedelete.ovpl -rm regional1 -u NNMIadminUserName -p NNMIadminPassword
```
- 10 これらのコマンドにより、regional1 から転送されたノードレコードを global1 から削除します。コマンドでは、regional1 から global1 に転送されたノードに関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、以下を実行します。
  - 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。
- 12 regional2 など、global1 に接続している他のリージョナル NNMi 管理サーバーについても手順 1 から手順 11 を実行します。

## 追加情報

### 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、global1 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。



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。

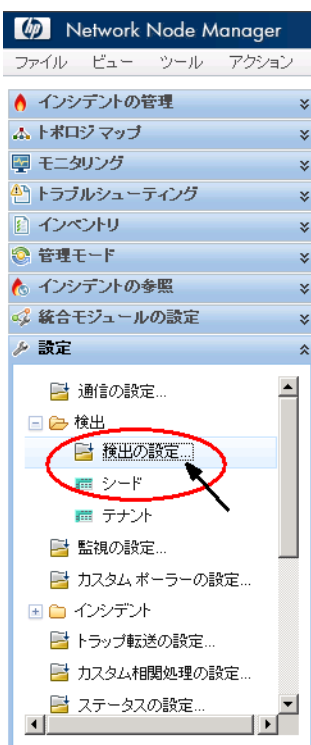
global1 は、regional1 または regional2 が検出を完了するたびに、regional1 と regional2 を同期します。NNMi は FDB (転送データベース) データを使用して、レイヤー 2 接続を計算します。FDB データは非常にダイナミックなもので、特に、1 つのグローバルサーバーに複数のリージョナルサーバーが接続しているような場合には、検出するごとに大きく異なります。



ユーザーが修正した属性やアプリケーションが修正した属性に対する変更は、グローバルサーバーでは同期中に更新されません。

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

regional1 または regional2 の NNMi コンソールから、[設定] ワークスペースの [検出の設定] をクリックします。



- 13 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]を調整します。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。

ファイル ビュー ツール アクション ヘルプ

検出の設定

保存して閉じる

▼ グローバル制御

再検出周期 1.00 日

**非応答ノードの削除の制御**

指定した非応答日数が経過すると、NNMIによってNNMIデータベースからノードが削除されます。ゼロ(0)は、非応答ノードを削除しないことを意味します。詳細は、[ここ](#)をクリックしてください。

応答のないノードを削除するまでの期間(日数) 0

**スパイラル検出 Ping スイープコントロール (IPv4のみ)**

このコントロールは、すべての自動検出ルールの(Ping スイープを有効にする) 選択肢を上書きします。

Ping スイープ なし

スイープ間隔 1.00 日

**ノード名の解決**

最初の選択 短いDNS名

2番目の選択 短いsysName

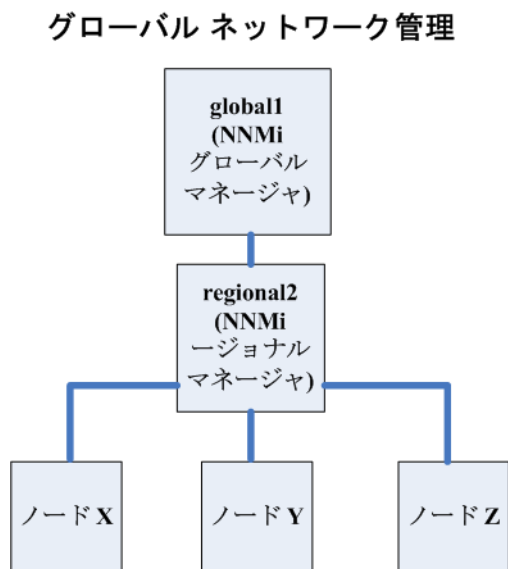
3番目の選択 IPアドレス

- 14 [保存して閉じる]をクリックします。

## デバイスのステータスの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 22 ノードのステータスの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、以下を実行します。

- 1 global1 から、**[インベントリ]**ワークスペースの**[ノード]**をクリックします。
- 2 ノードインベントリから Node X を選択します。
- 3 **[アクション]>[ステータスのポーリング]**メニュー項目を使用して、Node X のステータスのポーリングを要求します。
- 4 NNMi 管理サーバー global1 は、リージョナル NNMi 管理サーバー regional2 からのステータスのポーリングを要求し、結果を画面に表示します。ステータスのポーリング要求は、global1 と regional2 のどちらから発行しても問題はありません。ステータスの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には、以下を実行して 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]**をクリックします。
- 2 ノードインベントリから Node X を選択します。
- 3 **[アクション]>[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します。
- 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、結果を画面に表示します。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。



## グローバルマネージャーを使ったデバイスステータスの判定とNNMi インシデント生成

NNMi 管理サーバー global1 は、リージョナルマネージャー regional1 と regional2 からくるステータス変更をリッスンし、ローカルデータベースにあるステータスを更新します。

NNMi 管理サーバー regional1 と regional2 の NNMi StatePoller サービスは、監視するデバイスの状態の値を計算します。global1 は、regional1 と regional2 から状態の値の更新を受け取ります。global1 は、自分が検出するノードにポーリングしますが、regional1 と regional2 によって管理されているノードにはポーリングしません。

regional1 によって管理されているノードの管理モードを変更した後、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、転送フィルターを regional1 や regional2 に使用することは避けたほうがよいでしょう。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、転送フィルター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。HP では、グローバルマネージャーに過剰な負荷がかからないように、リージョナルマネージャーは量の少ない、重要なトラップを転送するよう設定することをお勧めします。NNMi は、転送されたトラップが TrapStorm インシデントを引き起こすような場合、転送されたトラップを削除します。NNMi コンソールで TrapStorm Management Event の詳細を参照してください。

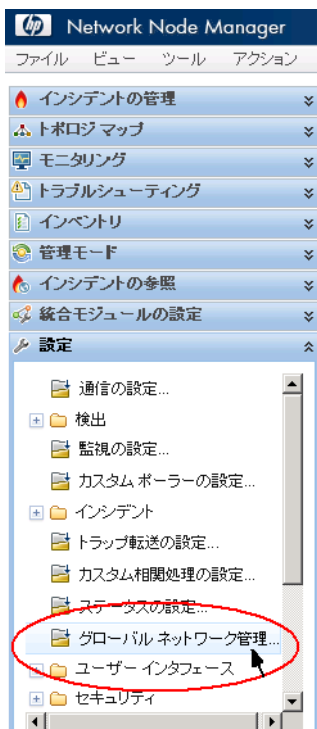
## グローバルネットワーク管理でアプリケーションフェイルオーバーの設定を行う

グローバルマネージャーとリージョナルマネージャーの両方を、アプリケーションフェイルオーバーを使用するよう設定できます。グローバルマネージャーとリージョナルマネージャーは、アクティブなシステムを自動的に検出して接続します。

### グローバルマネージャーでの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を認識するよう global1 を設定するには、以下を実行します。

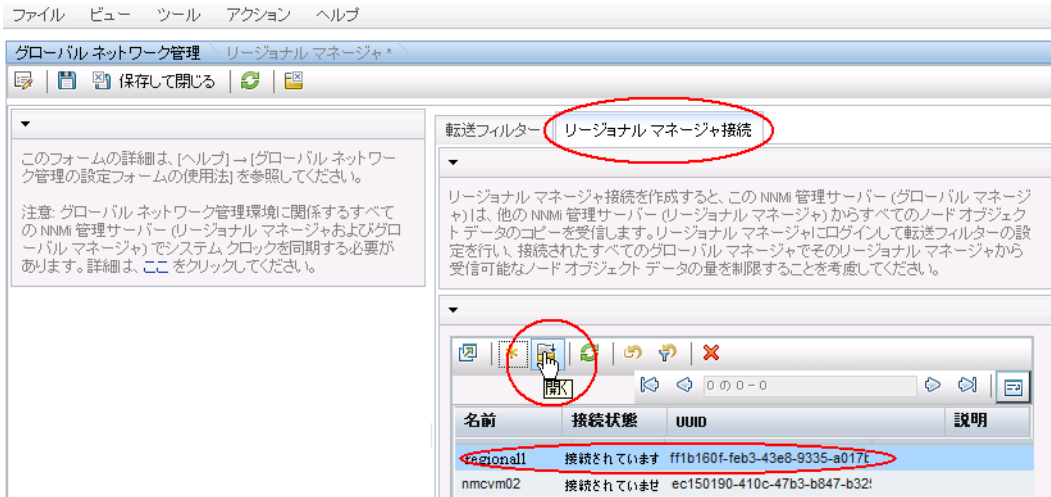
- 1 global1 の NNMi コンソールで、[ 設定 ] ワークスペースの [ **グローバルネットワーク管理** ] をクリックします。



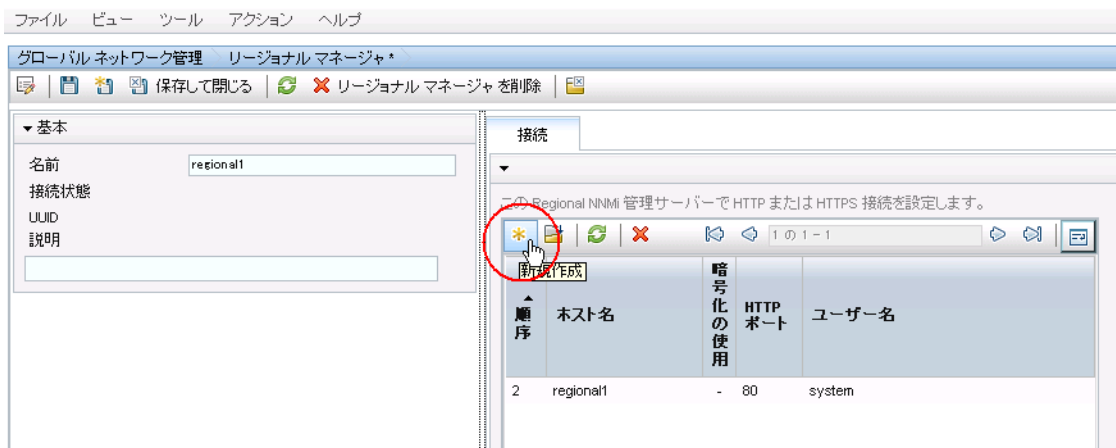
regional1 をアプリケーションフェイルオーバー用に設定し、セカンダリサーバーとして regional1\_backup を設定したと想定します。

- 2 [ **リージョナルマネージャ接続** ] をクリックします。

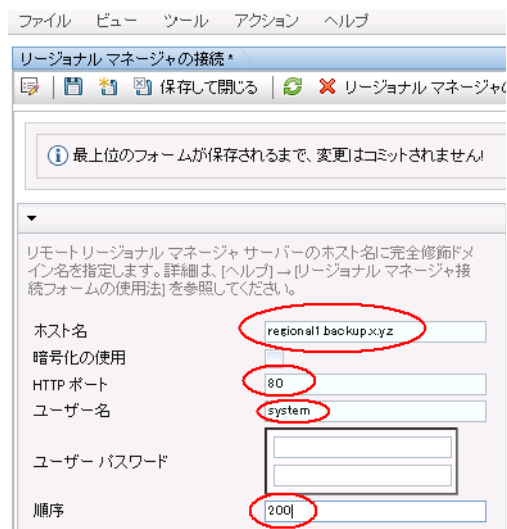
- 3 regional1 を選択して [開く] アイコンをクリックします。



- 4 [新規作成] アイコンをクリックします。



- 5 [ホスト名]、[HTTP ポート]、[ユーザー名]、および [順序] に値を入力します。順番の値には、regional1 より大きな値を設定します。



6 [保存して閉じる] を 3 回クリックして作業を保存します。

リージョナルマネージャーが失敗すると、グローバルマネージャーは以下を実行します。

- a プライマリに問い合わせます。
- b プライマリからの応答がない場合、セカンダリに問い合わせます。

グローバルシステムでアクティブシステムが応答しないことを検出すると、順序の番号が最も小さいものから再接続を試みます。

## 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

### NNMi ヘルプのトラブルシューティング情報

グローバルネットワーク管理のトラブルシューティング情報については、NNMi ヘルプの「グローバルネットワーク管理のトラブルシューティング」を参照してください。

### クロック同期

グローバルネットワーク管理 (グローバルマネージャーとリージョナルマネージャー) やシングルサインオン (SSO) に属するネットワーク環境内のすべての NNMi 管理サーバーは、それぞれの内部タイムクロックを世界標準時で同期化する必要があります。たとえば、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 がリージョナルマネージャーに接続されていません。[ヘルプ] > [システム情報] をクリックし、[グローバルネットワーク管理] タブを選択します。

グローバルマネージャーの nnm.0.0.log ファイルに次のメッセージがないか確認します。

警告: <number of seconds> のクロックの違いにより、システム <serverName> には接続されません リモート時間は、<date/time> です。

クロックが合わなくなり、再同期化が必要です。グローバルマネージャーの nnm.0.0.log ファイルに次のメッセージがないか確認します。

警告: <number of seconds> のクロックの違いにより、システム <serverName> には接続されません リモート時間は、<date/time> です。

この警告が表示されて数分以内に、NNMi はリージョナルマネージャ接続を切断します。また、NNMi コンソールの下部に次のメッセージが表示されます。

NNMi がリージョナルマネージャーに接続されていません。[ヘルプ] > [システム情報] をクリックし、[グローバルネットワーク管理] タブを選択します。

### グローバルネットワーク管理システム情報

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。

## グローバルマネージャーからのリージョナルマネージャー検出の同期化

global1 と regional2 の間で情報に矛盾があることに気がついたと想定します。それを解決するため、global1 から **nnmnode rediscover.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。

264 ページの [図 22](#) 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。以下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：**nnmnode rediscover.ovpl -u username -p password -rm regional2**



**nnmnode rediscover.ovpl** コマンドで **-fullsync** フラグを使用して、ポーリングされるオブジェクトのすべての状態とステータスを同期することができます(ただし、この処理には時間がかかり、システム負荷が増加する可能性があります)。詳細については、**nnmnode rediscover.ovp** リファレンスページまたは **UNIX** のマンページを参照してください。



以下の点に注意してください。

- 手動による再同期に続いて **NNMi** で再同期するときに、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。
- この再同期中に以下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、アップグレード、アプリケーションフェイルオーバー、バックアップの復元または手動による再同期の後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。

- この再同期中に **NNMi** を停止しないでください。再同期を確実に完了するには、手動による再同期の後で **NNMi** を数時間実行し続けます。

## 破損した global1 上のデータベースの修復

global1 のサービスを停止し、データベースを復元する必要がある場合、いくつかの方法があります。

- 1 global1のデータベースを正しく復元すると、regional1と regional2はglobal1を使用してキャッシュされた情報を同期化します。global1 をオンラインに戻した後、手動で行う手順はありません。
- 2 global1 のサービスが長時間停止すると、**手順 1** は正常に機能しないことがあります。これを解消するには、global1 で **nnmnode rediscover.ovpl** スクリプトを実行して global1、regional1 および regional2 で新たな検出を開始します。この場合、さらに迅速に更新されたステータス情報を入手するため、キーデバイスに対してステータスのポーリングを実行できます。
- 3 global1 のデータベースを復元できない場合、**nnmsubscription.ovpl** スクリプトを使用して古い global1 データを regional1 と regional2 のデータベースから消去するには、サポートに問い合わせる必要があります。

# NNMi 9.0x/9.1xからのNNMi 9.20へのグローバルマネージャーとリージョナルマネージャーのアップグレード

## 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

グローバルマネージャーが、NNMi 9.0x パッチ 2 またはそれより前のレベルのパッチを実行しているリージョナルマネージャーに接続されている場合、グローバルマネージャーとリージョナルマネージャー間のSNMPクエリーは機能しません。これを解決するには、リージョナルマネージャーをNNMi 9.0x パッチ 3 またはそれより後のレベルのパッチにアップグレードします。最良の結果を得るには、グローバルマネージャーのバージョンとNNMi パッチレベルが、リージョナルマネージャーと同じである必要があります。

- ▶ HP では、NNMi 9.20 が実行されているグローバルマネージャーに接続された、NNMi 9.0x または 9.1x が実行されているリージョナルマネージャーはサポートしていません。グローバルマネージャーとリージョナルマネージャーの両方で、同一バージョンのNNMiを実行する必要があります。

## 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 9.20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ーとリージョナルマネージャー間の接続は、グローバルネットワークマネージャーとリージョナルマネージャーの両方が9.20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HPはお勧めします。

たとえば、以下の手順でNNMi管理サーバーをアップグレードできます。

- 1 リージョナルマネージャーをNNMi 9.20にアップグレードし、正しく動作することを確認します。リージョナルマネージャーのアップグレード中、グローバルマネージャーは切断されたままになります。
- 2 グローバルマネージャーをNNMi 9.20にアップグレードします。グローバルマネージャーで完全な再同期が実行され、グローバルマネージャーとリージョナルマネージャーの接続が切断している間に発生したすべてのイベントが取得されます。管理者がグローバルマネージャーから `nnmnode rediscover.ovpl -all -fullsync` を発行すると同じ結果が得られます。詳細については、`nnmnode rediscover.ovpl` のリファレンスページまたはUNIXのマニュアルを参照してください。

- ▶ 以下の点に注意してください。

- アップグレードに続いてNNMiが再同期するときに、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。
- この再同期中に以下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、アップグレード、アプリケーションフェイルオーバー、バックアップの復元または手動による再同期の後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。

- この再同期中にNNMiを停止しないでください。再同期を確実に完了するには、アップグレードの後でNNMiを数時間実行し続けます。

---

## グローバルネットワーク管理と NNM iSPI または第三者の統合

NNM iSPI または第三者の統合は、導入にあたりそれぞれ独自のガイドラインがあります。この章の例では、複数の NNM iSPI を regional1 のみ、global1 のみ、または regional1 と global1 の両方に配備できます。その他の NNM iSPI または第三者の統合については、regional1 と global1 の両方にインストールされている必要があります。詳細については、NNM iSPI または第三者の統合に関するドキュメントを参照してください。

### HP Network Node Manager iSPI Performance for Metrics Software

NNMi がグローバルネットワーク管理環境で配備されている場合は、以下を実行する必要があります。

- 1 NNMi 管理サーバーごとに Network Performance Server (NPS) の1つのインスタンスを配備します。すべてのリージョナルマネージャーおよびグローバルマネージャーには、NPS の別個のインスタンスがインストールされ、配備されている必要があります。
- 2 すべてのリージョナルマネージャーおよびグローバルマネージャーで、イネーブルメントスクリプトを1回実行します。

---

## 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的なテナントに加え、NNMi リージョナルマネージャーが必要です。「[NAT 環境の重複 IP アドレスの管理](#)」(197 ページ) を参照してください。NNMi ヘルプも参照してください。





# IPv6 用 NNMi Advanced の設定

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 管理により、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可能です。ただし、いく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。

この章には、以下のトピックがあります。

- 機能説明
- 前提条件
- ライセンス
- サポートされる設定
- NNMi のインストール
- IPv6 機能のアクティブ化
- IPv6 機能の非アクティブ化

---

## 機能説明

NNMi IPv6 管理機能には、以下の機能があります。

- 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付け

- 以下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 ] ワークスペースにある [ 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使用したシード済みデバイスの検出
- 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、および関連付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
  - 通信設定
  - 検出の設定
  - モニタリングの設定
  - 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 SDK Web サービスサポート
- IPv6 インタフェースに対する NNM iSPI Performance for Metrics のサポート

NNMi IPv6 管理機能には、以下は含まれません。

- IPv6 サブネット接続の検出
- 検出のための IPv6 ping スィープの使用
- 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

## 前提条件

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、『NNMi デプロイメントリファレンス』、『NNMi リリースノート』、および『NNMi システムおよびデバイス対応マトリックス』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必要があります。つまり、IPv4 と IPv6 両方を使用して通信するということです。

**重要 :** HP NNMi で IPv6 検出を設定していて、HP Universal CMDB (HP UCMDB) 統合を使用している場合、UCMDB HP Discovery and Dependency mapping (DDM、検出および依存関係マッピング) インポートタスクは失敗します。HP NNMi で HP UCMDB 統合を使用するには、IPv6 検出を無効にする必要があります。

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、『NNMi システムおよびデバイス対応マトリックス』を参照してください。その他に、以下の要件があります。

- 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 を有効化し、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、グローバルユニキャストアドレスまたは一意のローカル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。



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。たとえば、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、DSN から IPv6 アドレスへのホスト名と IPv6 アドレス から DSN へのホスト名を解決する必要があります。たとえば、AAAA DNS レコードからのホスト名と AAAA DNS へのホスト名を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。

## 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、「NNMi のライセンス」(123 ページ)を参照してください。

NNMi 製品には、インスタントオンライ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、永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。

## サポートされる設定

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、『NNMi システムおよびデバイス対応マトリックス』を参照してください。

### 管理サーバー

以下の表に、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 25 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
IPv6 シードを使用した検出	非対応	対応
IPv6 アドレスおよびサブネットインベントリ	非対応	対応

表 25 管理サーバーの機能 ( 続き )

機能	IPv4 専用	デュアルスタック
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	非対応	対応
ICMPv6を使用したIPv6アドレスステータス	非対応	対応
IPv4 専用管理ノード	対応	対応
IPv4 シードを使用したノード検出	対応	対応
IPv4 シードを使用したノード検出	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
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
IPv4アドレスおよびサブネットインベントリ	対応	対応

## IPv6 をサポートする SNMP MIB

NNMi では、IPv6 用の以下の 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

## NNMi のインストール

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。

## IPv6 機能のアクティブ化

IPv6 専用デバイスの検出や IPv6 アドレスステータスの監視など、IPv6 通信を必要とする機能では、NNMi 管理サーバーに IPv6 グローバルユニキャストアドレスが設定され機能することが必要です。

以下に示す手順は、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します。NNMi ライセンスの取得とインストールの詳細については、「[NNMi のライセンス](#)」(123 ページ)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 ファイルを編集します。以下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します。

▶ 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、以下のプロパティをコメント解除します。  
java.net.preferIPv4Stack=false

▶ プロパティをコメント解除するには、行の先頭から #! 文字を削除します。

- b NNMi で IPv6 通信全体を有効化するには、以下のプロパティをコメント解除します。  
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します。
  - 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 以下のコマンドを使用して、NNMi プロセスを確認します。

**ovstatus -v ovjboss**

起動に成功すると、以下のように表示されます。

```
object manager name: ovjboss
state:                RUNNING
PID:                  <Process ID #>
last message:        Initialization complete.
exit status:          -
additional info:
```

SERVICE	STATUS
CommunicationModelService	サービスが起動されました
CommunicationParametersStatsService	サービスが起動されました
CustomPoller	サービスが起動されました
IslandSpotterService	サービスが起動されました
ManagedNodeLicenseManager	サービスが起動されました
MonitoringSettingsService	サービスが起動されました
NamedPoll	サービスが起動されました
msApa	サービスが起動されました
NmsCustomCorrelation	サービスが起動されました
NmsDisco	サービスが起動されました
NmsEvents	サービスが起動されました
NmsEventsConfiguration	サービスが起動されました
NmsExtensionNotificationService	サービスが起動されました
NnmTrapService	サービスが起動されました
PerformanceSpiAdapterTopologyChangeService	サービスが起動されました
PerformanceSpiConsumptionManager	サービスが起動されました
RbaManager	サービスが起動されました
RediscoverQueue	サービスが起動されました
SpmdjbossStart	サービスが起動されました
StagedIcmp	サービスが起動されました
StagedSnmp	サービスが起動されました
StatePoller	サービスが起動されました
TrapConfigurationService	サービスが起動されました
TrustManager	サービスが起動されました

- 6 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ます。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。
- 7 オプションで、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します。デュアルスタック管理ノードは、IPv4 または IPv6 いずれかを使用して通信できるノードです。これには、以下の手順を実行します。
  - a NNMi コンソールで、[設定] ワークスペースにある [通信の設定] をクリックします。
  - b [管理アドレスの選択] セクションを見つけます。[IP バージョン設定] フィールドで、[IPv4]、[IPv6]、または [いずれか] を選択します。
  - c 変更を保存します。

d NNMi 管理サーバーを再起動します。

NNMi 管理サーバーで `ovstop` コマンドを実行します。

NNMi 管理サーバーで `ovstart` コマンドを実行します。

スピードアップを図るには、デュアルスタックノードとわかっているノードを選択し、NNMi コンソールで [ **アクション** ] > [ **設定のポーリング** ] コマンドを使用します。nnmnodediscover.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、nnmnodediscover.ovpl のリファレンスページまたは UNIX のマン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。

## IPv6 機能の非アクティブ化

以下のいずれかの方法を使用して、管理上 IPv6 機能を無効化することができます。

- 1 nms-jboss.properties ファイルの IPv6 マスタースイッチをオフにし、NNMi を再起動します。
- 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ます。  
NNMi ライセンスの変更の詳細については、「[NNMi のライセンス](#)」(123 ページ)を参照してください。

以下のセクションでは、IPv6 を無効化した後の 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。

### 非アクティブ化後の 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** ] > [ **モニタリングの設定** ] を使用すると、関連する [ 監視設定 ] ルールで [ IP アドレスの障害のポーリング ] が有効になっている場合でも、NNMi は " 障害 ICMP ポーリングの有効化: false" と表示します。

### 非アクティブ化後の 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、以下の場合には、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後で、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。NNMi は 非 SNMP IPv6 ノードを削除しません。IPv6 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。基本ライセンスは、検出されたノードすべての管理を続行するのに十分な機能があります。



NNMi は非 SNMP IPv6 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。

- **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。基本ライセンスは、検出したノードすべての管理を続行するのに十分な機能はありません。NNMi はすぐに、非 SNMP IPv6 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 [unmanaged] 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、以下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 unmanaged 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、または UNIX のマン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]>[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。nnmnode rediscover.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、nnmnode rediscover.ovpl のリファレンスページまたは UNIX のマン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、以下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 unmanaged 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、または UNIX のマン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]>[設定のポーリング]** コマンドを使用して、NNMi が unmanaged から managed に変更したノードを検出できるようにします。nnmnode rediscover.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、nnmnode rediscover.ovpl のリファレンスページ、または UNIX のマン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]>[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。

## 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。たとえば、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的设计上、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]>[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。

# Solaris ゾーン環境での NNMi の実行

Solaris オペレーティングシステムのサポート対象バージョンでは、Solaris ゾーン環境で特別な設定を行わなくても、HP Network Node Manager i Software (NNMi) が動作します。

この章には、以下のトピックがあります。

- Solaris ゾーンでの NNMi のインストール
- Solaris ゾーンでのトラップ転送
- Solaris ゾーン環境での NNMi アプリケーションフェイルオーバーの実行
- Solaris ゾーン環境の HA での NNMi の実行

---

## Solaris ゾーンでの NNMi のインストール

Solaris ゾーン環境で NNMi アプリケーションフェイルオーバーを実装する場合は、「Solaris ゾーン環境での NNMi アプリケーションフェイルオーバーの実行」(284 ページ)を参照してください。

高可用性 (HA) で Solaris ゾーンを実行する場合は、「Solaris ゾーン環境の HA での NNMi の実行」(284 ページ)を参照してください。

他のすべての導入モデルの場合は、『HP Network Node Manager i Software インタラクティブインストールガイド』の説明に従って NNMi をインストールします。

---

## Solaris ゾーンでのトラップ転送

NNMi が管理対象デバイスから受信した SNMP トラップを別のアプリケーションに転送する場合を考えます。これを行うには、[設定] ワークスペースの [トラップ転送の設定] に移動します。詳細については、NNMi ヘルプを参照してください。

Solaris ゾーン環境では未処理トラップの転送がサポートされていないため、[元のトラップ] 転送オプションは選択しないでください。Solaris ゾーン環境で NNMi を実行する場合、他のいずれかの転送オプションを選択してください。

## Solaris ゾーン環境でのNNMiアプリケーションフェイルオーバーの実行

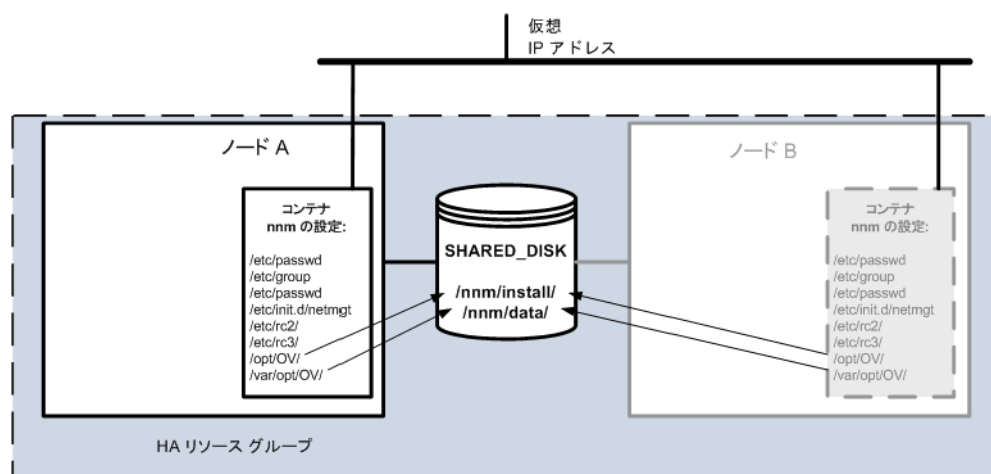
Solaris ゾーン環境でNNMiアプリケーションフェイルオーバー機能を使用する場合、2つの各物理システムのゾーンにNNMiをインストールします。

「アプリケーションフェイルオーバー構成のNNMiの設定」(289 ページ)の説明に従ってアプリケーションフェイルオーバーを設定します。この手順全体で、「サーバー X」は一方のゾーンを表し、「サーバー Y」はもう一方のゾーンを表します。

## Solaris ゾーン環境の HA での NNMi の実行

Solaris ゾーン環境では、HA クラスタでNNMiを実行するためにNNMiが提供するソリューションを実装する必要はありません。Veritas Cluster Server (VCS) はゾーンを認識するため、図 23 に示すようにゾーンの HA リソースグループを設定します。

図 23 HA で実行される Solaris ゾーンでの NNMi



これは、この環境で NNMi を実行するための最低限の設定です。NNMi のインストールプロセスでは、**nmsdb** グループに **nmsdbmgr** ユーザーが作成され、起動設定がホストシステムに追加されます。このセットアップは HA クラスタの 2 番目のノードに複製されます。

NNMi をインストールして、HA リソースグループ内のゾーンで実行するには、以下の手順を実行します。

- 1 共有ディスクで、NNMi インストールフォルダーを作成します。
  - /nmm/install
  - /nmm/data
- 2 ノード A で、**nmm** という新しいゾーンを作成して準備します。
  - a Solaris ゾーンのマニュアルの説明に従って、ゾーン **nmm** を作成します。  
すべての設定パラメーターはゾーン作成時に設定されます。
  - b ゾーン **nmm** を起動します。

- c ゾーン **nnm** にログオンし、以下のシンボリックリンクを作成します。
    - 共有ディスクの /nnm/install/ を指し示す /opt/OV/
    - 共有ディスクの /nnm/data/ を指し示す /var/opt/OV/
  - d ゾーン **nnm** からログオフして、そのゾーンをシャットダウンします。
- 3 ノード B で、**nnm** という同一の新しいゾーンを作成して、**NNMi** をインストールします。
- a ノード A のゾーン **nnm** と同じプロパティ (IP アドレスなど) を使用して、ゾーン **nnm** を作成します。
  - b ゾーン **nnm** を起動します。
  - c ゾーン **nnm** にログオンし、以下のシンボリックリンクを作成します。
    - 共有ディスクの /nnm/install/ を指し示す /opt/OV/
    - 共有ディスクの /nnm/data/ を指し示す /var/opt/OV/
  - d 以下のコマンドを入力して **NNMi** インストーラーがシンボリックリンクに従うように指定します。
 

```
PKG_NONABI_SYMLINKS=true
```
  - e **nnm** ゾーン内に **NNMi** をインストールします。  
**NNMi** は、共有ディスクの /nnm/install/ および /nnm/data/ ディレクトリにインストールされます。
  - f **nnm** ゾーン外からアクセスできる一時保存場所 (共有ディスクなど) に以下のファイルをコピーします。
    - /etc/passwd
    - /etc/group
    - /etc/shadow
    - /etc/init.d/netmgt
  - g ゾーン **nnm** からログオフして、そのゾーンをシャットダウンします。
- 4 ノード A で、**NNMi** で変更されたシステムファイルをコピーして **NNMi** を起動します。
- a ゾーン **nnm** を起動します。
  - b ゾーン **nnm** にログオンし、手順 3 で特定した一時保存場所からゾーンの適切な場所にファイルをコピーします。
    - /etc/passwd
    - /etc/group
    - /etc/shadow
    - /etc/init.d/netmgt

- c (ノードBでNNMiをインストールするときに作成された設定を複製するために)以下のシンボリックリンクを作成します。
  - /etc/init.d/netmgt を指し示す /etc/rc0.d/K01netmgt
  - /etc/init.d/netmgt を指し示す /etc/rc1.d/K01netmgt
  - /etc/init.d/netmgt を指し示す /etc/rc2.d/K01netmgt
  - /etc/init.d/netmgt を指し示す /etc/rc3.d/S98netmgt
  - /etc/init.d/netmgt を指し示す /etc/rcS.d/K01netmgt
- d 以下のコマンドを実行して NNMi を起動します。

**ovstart**

- 5 ノードAとノードBの両方のゾーン **nnm** を含むリソースグループが作成されるように、**Veritas Cluster Server** を設定します。  
詳細については、**VCS** のマニュアルを参照してください。

# 復元

HP Network Node Manager i Software (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 つの方法がサポートされます。

- NNMi のアプリケーションフェイルオーバーでは、組み込み 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 (NNMi で Oracle データベースが使用されている場合は、2 つのシステムが同一のデータベースに別々の時間に接続されます)。
- 高可用性 (HA) クラスタで NNMi を実行すると、組み込み 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 管理サーバーがほぼ 100 パーセント利用されます (NNMi で Oracle データベースが使用されている場合は、共有ディスクに NNMi 設定ファイルが含まれ、2 つのシステムが同一のデータベースに別々の時間に接続されます)。

両方の手法では、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。

表 26 では、NNMi データ復元の 2 つの方法のさまざまな側面を比較しています。

表 26 NNMi データ復元の比較

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 または NNMi Advanced	<ul style="list-style-type: none"> <li>• NNMi または NNMi Advanced</li> <li>• 個別に購入する HA 製品</li> </ul>
フェイルオーバーにかかる時間	組み込み NNMi データベース：トランザクションログを処理する時間 (通常の状態では、NNM iSPI を使用しない NNMi の場合、10-60 分間)。 Oracle NNMi データベース：ほぼ瞬時。	通常の状態では NNM iSPI を使用しない NNMi の場合、5-30 分間。
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だったもの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 コンソールに接続する必要があります。一部のアプリケーションは NNMi 管理サーバーの動作に従いますが、大部分のアプリケーション (NNM iSPI など) は従いません。	完全。すべての接続では 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 製品のみ)
購入ライセンス	機能ごと <ul style="list-style-type: none"> <li>• 商用ライセンスは、最初のアクティブサーバーの IP アドレスが対象になります。</li> <li>• 非商用ライセンスは、最初のスタンバイサーバーの IP アドレスが対象になります。</li> </ul>	機能ごと NNMi HA リソースグループの仮想 IP アドレスが対象になる商用ライセンスまたは非商用ライセンス

表 26 NNMi データ復元の比較

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
インストールするライセンス	<ul style="list-style-type: none"> <li>最初のアクティブサーバーには商用ライセンスキー。</li> <li>最初のスタンバイサーバーには非商用ライセンスキー。</li> </ul>	<ul style="list-style-type: none"> <li>最初のアクティブサーバーには、共有ディスクで管理される非商用ライセンスキー。</li> </ul>
NNM iSPI のサポート	さまざまなサポートがあります。各 NNM iSPI のマニュアルを参照してください。	
グローバルネットワーク管理とのインタラクション	<ul style="list-style-type: none"> <li>アプリケーションフェイルオーバーまたは HA 用に各グローバルマネージャーを設定可能。</li> <li>アプリケーションフェイルオーバーまたは HA 用に各リージョナルマネージャーを設定可能。</li> <li>それぞれの設定には、2 つの物理または仮想システムが必要です。<sup>a</sup></li> <li>グローバルマネージャーまたはリージョナルマネージャーがフェイルオーバーすると、NNMi は、グローバルマネージャーとリージョナルマネージャー間の接続を再確立します。</li> </ul>	
NNMi のメンテナンス	パッチまたはアップグレードを適用する前に、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 にパッチおよびアップグレードを適用できます。

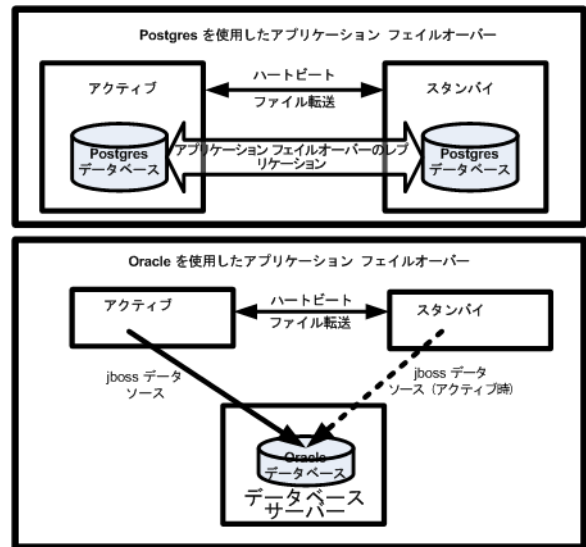
a. HA の仮想マシンサポートは、HA ソフトウェアベンダーによる仮想システムのサポートに依存します。

この項では以下の章について説明します。

- アプリケーションフェイルオーバー構成の NNMi の設定
- 高可用性クラスタに 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



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 HP Network Node Manager i Software (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。

この章には、以下のトピックがあります。

- 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構成の NNMi の設定
- アプリケーションフェイルオーバー機能の使用
- フェイルオーバー後、元の設定に戻る
- NNM iSPI およびアプリケーションフェイルオーバー
- 統合アプリケーション
- アプリケーションフェイルオーバーの無効化
- 管理タスクおよびアプリケーションフェイルオーバー
- ネットワークレイテンシ/帯域に関する考慮

## 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機能は、組み込みデータベースまたは **Oracle** 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、セカンダリサーバーに **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、以下の用語と定義を使用しています。

- **アクティブ**: **NNMi** プロセス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** クラスタ内のシステム。このシステムは **NNMi** プロセス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** が使用する組み込みデータベース。
- **Cluster Manager**: アプリケーションフェイルオーバー機能における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。

## 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。この章では、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のみが **NNMi** サービスを実行します。

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が正しく機能するには、**NNMi** 管理サーバーが以下の要件を満たしている必要があります。

- 両方の **NNMi** 管理サーバーが同じ種類のオペレーティングシステムを実行している必要があります。たとえば、アクティブサーバーが **HP-UX** オペレーティングシステムを実行している場合、スタンバイサーバーも **HP-UX** 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。
- 両方の **NNMi** 管理サーバーは同じバージョンの **NNMi** を実行している必要があります。たとえば、アクティブサーバーで **NNMi 9.20** を実行している場合、スタンバイサーバーでも同一の **NNMi** バージョンである **NNMi 9.20** がインストールされている必要があります。**NNMi** パッチレベルについても、同一レベルのパッチが両サーバーに適用されている必要があります。
- 両方の **NNMi** 管理サーバーのシステムパスワードが同一である必要があります。
- **Windows** オペレーティングシステムの **NNMi** インストールでは、`%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** 管理サーバーは同じデータベースを実行している必要があります。たとえば、両方の **NNMi** 管理サーバーで **Oracle** を実行しているか、両方の **NNMi** 管理サーバーで組み込みデータベースを実行している必要があります。アプリケーションフェイルオーバー機能を使用する場合、種類の異なるデータベースを組み合わせることはできません。
- 両方の **NNMi** 管理サーバーのライセンス属性が同一である必要があります。たとえば、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。
- **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、「[検出の評価](#)」(69 ページ)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。この条件を満たしたら、「[アプリケーションフェイルオーバー構成の NNMi の設定](#)」(292 ページ)に示した手順を実行してください。詳細については、「[NNMi 9.20 およびウェルノウンポート](#)」(439 ページ)を参照してください。

▶ 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。

▶ **NNMi 9.20** のインストールまたはアップグレード時に、**NNMi** インストールによって **NNMi** クラスター通信用のネットワークインタフェースが選択されます。通常、選択されたネットワークインタフェースは、システムの最初に非ループバックインタフェースになります。**NNMi** クラスターが設定された場合、選択されたインタフェースがその設定で使用されます。インタフェースを調整する必要がある場合は、以下の手順を実行します。

1 以下のファイルを編集します。

— **Windows:**

```
%NnmDataDir%\shared\%nnm%\conf\props\%nms-cluster.properties
```

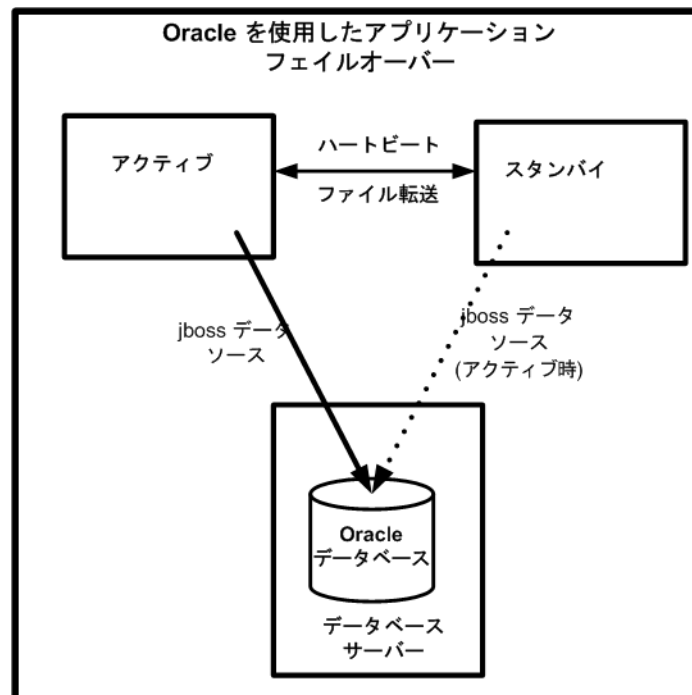
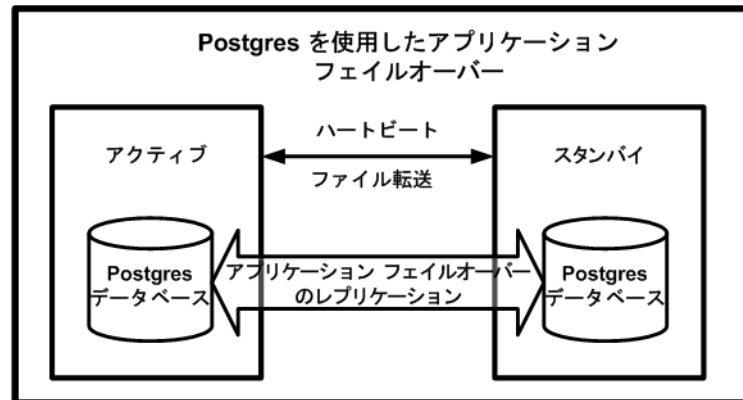
— **UNIX:** \$NnmDataDir/shared/nnm/conf/props/  
nms-cluster.properties

最小値と最大値を含む `nms-cluster.properties` ファイルのパラメーターは、`nms-cluster.properties` ファイル内にそれぞれ記載されています。

2 目的のインタフェースを指し示すように `com.hp.ov.nms.cluster.interface` パラメーターを調整します。

## アプリケーションフェイルオーバー構成の NNMi の設定

- 1 HP Network Node Manager i Software インタラクティブ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 をインストールします。



- 2 「[NNMi のライセンス](#)」(123 ページ)に記載されているように、サーバー X の各ライセンスに対し、サーバー Y に使用する同じ非商用のライセンスを取得し、サーバー Y にインストールします。
- 3 各サーバーで `ovstop` コマンドを実行して NNMi をシャットダウンします。



Oracle データベースでアプリケーションフェイルオーバーを使用している場合は、スタンバイサーバーの NNMi プロセスはすでに停止しています。

- 4 Oracle データベースでアプリケーションフェイルオーバーを使用している場合、「[アプリケーションフェイルオーバー構成の NNMi の手動設定](#)」(431 ページ)の設定手順を実行します。

## NNMi クラスタセットアップウィザードを使用したクラスタの設定 (組み込みデータベースユーザーのみ)

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、以下の操作ができます。

- クラスタ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 ブラウザーに以下を入力して、クラスタセットアップウィザードを起動します。

`http://<NNMIserv>:<port>/cluster`

- <NNMIserv> 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 にサインインします。
  - 3 [ローカルホスト名] と [リモートクラスタード] の値を入力してクラスタードを定義し、[次へ] をクリックします。
  - 4 [通信結果] ページで、通信の検証結果を確認します。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 [クラスタードプロパティを定義] ページで、[クラスタード名] を入力して [バックアップ周期 (時間)] を定義します。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 [クラスタードポートを定義] ページで、[開始クラスタードポート] と [ファイル転送ポート] の値を入力します。



NNMi クラスタードでは、[開始クラスタードポート] で始まる 4 個の連続したポートが使用されます。

- 7 [次へ] をクリックします。
- 8 入力した情報の概要を確認します。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ード設定を保存します。
- 9 最後の概要には、設定が成功したかが示されます (項目ごとに「成功」というメッセージが示されます)。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。

クラスタードのセットアップに成功している場合、[終了] をクリックします。

- 10 両方のノードで `ovstop` を実行して、両方のノードの NNMi を直ちに停止します。
- 11 両方のノードで `nnmcluster` コマンドを実行して、2 つのノードをクラスタード構成にできることを確認します。ノードをクラスタード構成にできない場合は、「アプリケーションフェイルオーバー構成の NNMi の手動設定」(431 ページ) を参照してください。
- 12 `nnmcluster` コマンドを使用して、アクティブにするノード上の NNMi を起動します。NNMi が ACTIVE をレポートするまで待機します (「アプリケーションフェイルオーバー構成の NNMi の手動設定」(431 ページ) を参照)。
- 13 `ovstart` コマンドを使用して、スタンバイノードを起動します。

## クラスター通信の設定 (オプション)

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) に対してクエリを実行し、クラスター通信に使用する NIC を特定します (使用可能な最初の NIC が選択されます)。システムに複数の NIC が存在する場合、以下の手順を実行して、`nnmcluster` 操作に使用する NIC 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します。詳細については、`nnmcluster` リファレンスページ、または UNIX のマンページを参照してください。
- 2 以下のファイルを編集します。
  - **Windows:** `%NnmDataDir%\conf\%nm%\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ます。  
`com.hp.ov.nms.cluster.interface =< 値 >`
- 4 必要に応じて値を変更します。
- 5 `nms-cluster-local.properties` ファイルを保存します。

---

## 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ーマネージャーが実行しているため (アクティブノードとスタンバイノード)、クラスターマネージャーを使用してクラスターのステータスを表示できます。クラスターマネージャーには 3 つのモードがあります。

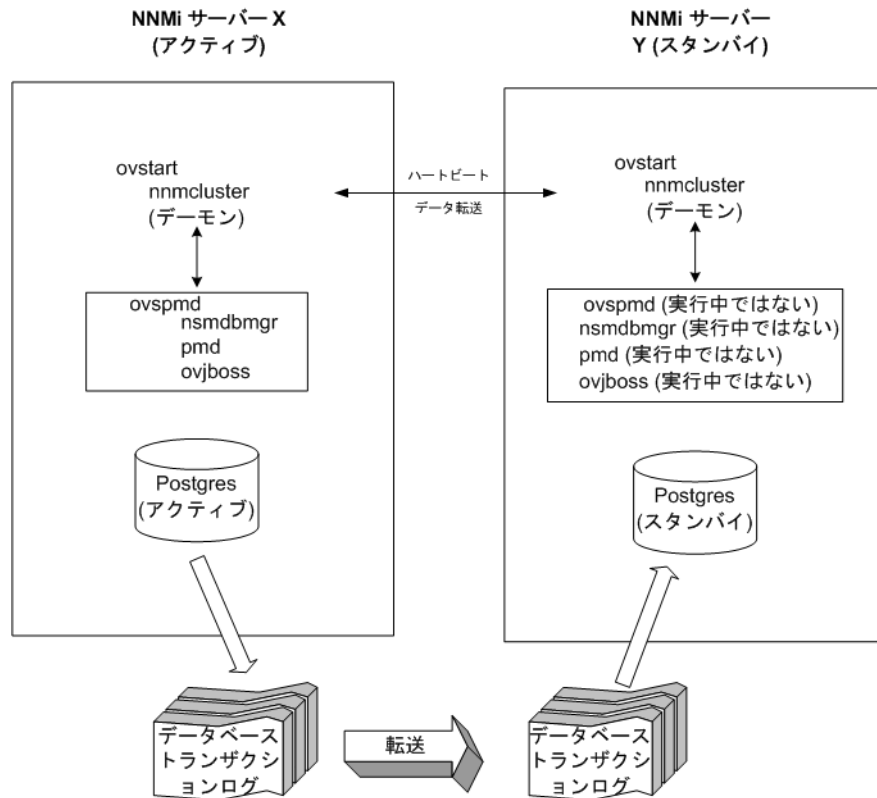
- **デーモンモード:** クラスターマネージャーのプロセスはバックグラウンドで実行し、`ovstop` および `ovstart` コマンドを使用して NNMi サービスを開始および停止します。
- **インタラクティブモード:** クラスターマネージャーは、NNMi 管理者がクラスターの属性を表示および変更できるインタラクティブセッションを実行します。たとえば、NNMi 管理者はこのセッションを使用して、アプリケーションフェイルオーバー機能を有効または無効にしたり、デーモンプロセスをシャットダウンしたりできます。
- **コマンドラインモード:** NNMi 管理者は、コマンドプロンプト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

詳細については、`nnmcluster` リファレンスページ、または UNIX のマンページを参照してください。

## 組み込みデータベースを使用したアプリケーションフェイルオーバーの動作

図24は、組み込み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24 アプリケーションフェイルオーバーの設定（組み込みデータベース）





NNMi 9.20 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。これにより、(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にのみ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。



アクティブノードとスタンバイノードの両方で、ファイアウォールが有効になっている場合、組み込みデータベースに使用しているポート(デフォルトではポート 5432)が開いていることを確認します。このポートは以下のファイルで設定されます。

**Windows:** %NNM\_CONF%\nmm\props\nms-local.properties

**UNIX:** \$NNM\_CONF/nmm/props/nms-local.properties

アクティブノードとスタンバイノードの両方を開始すると、スタンバイノードはアクティブノードを検知してアクティブノードにデータベースのバックアップをリクエストしますが、NNMi サービスは開始しません。このデータベースのバックアップは 1 つの **Java-ZIP** ファイルとして保存されます。すでにスタンバイノードに以前のクラスター接続から得た ZIP ファイルがあり、NNMi が、そのファイルとアクティブサーバーの同期が確認された場合は、ファイルは再送されません。

アクティブノードとスタンバイノードの両方が実行している間、アクティブノード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ノードに送信します。nms-cluster.properties ファイルの com.hp.ov.nms.cluster.timeout.archive パラメーターの値を変更すると、このデータの転送頻度を変更できます。これらのトランザクションログはスタンバイノードに蓄積されるため、スタンバイからアクティブになったときにすぐに利用することができます。

スタンバイノードがアクティブノードからデータベースの完全バックアップを受信すると、その情報を組み込み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い他のサービスがデータベースを使用できないことを組み込みデータベースに知らせます。

何らかの理由でアクティブノードが利用できなくなると、スタンバイノードは 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 NNMi 管理サーバーは、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。

アクティブ NNMi システムに障害が発生すると、スタンバイシステムは、ディスクバリエーションとポーリングアクティビティを開始します。このトランジションによって、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。



以下の点に注意してください。

- NNMi ではアプリケーション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。
- この再同期中に以下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、アップグレード、アプリケーションフェイルオーバー、バックアップの復元または手動による再同期の後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。

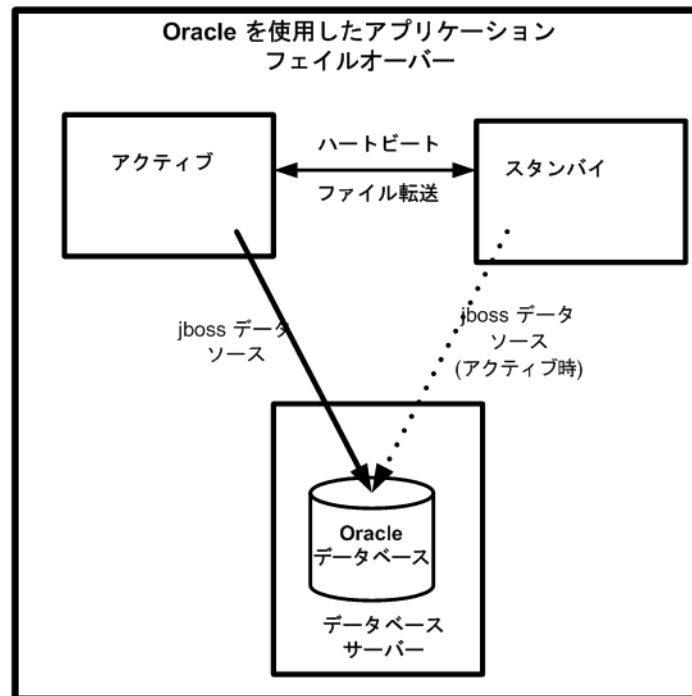
- この再同期中に NNMi を停止しないでください。再同期を確実にを行うには、アプリケーションフェイルオーバー後に数時間 NNMi が実行されている必要があります。



## Oracle データベースを使用したアプリケーションフェイルオーバーの動作

図 25 は、Oracle 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の設定を示します。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 25 アプリケーションフェイルオーバーの設定 (Oracle データベース)



何らかの理由でアクティブノードが利用できなくなると、スタンバイノードは 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。

アクティブ NNMi システムに障害が発生すると、スタンバイシステムは、ディスクバリとポーリングアクティビティを開始します。このトランジションによって、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。



以下の点に注意してください。

- NNMi ではアプリケーション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る。
- この再同期中に以下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、アップグレード、アプリケーションフェイルオーバー、バックアップの復元または手動による再同期の後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。

- この再同期中に NNMi を停止しないでください。再同期を確実に行うには、アプリケーションフェイルオーバー後に数時間 NNMi が実行されている必要があります。

## アプリケーションフェイルオーバーの例

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いくつかあります。

- 例 1: アクティブ NNMi 管理サーバーに障害が発生した。
- 例 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。
- 例 3: NNMi 管理者がクラスターをシャットダウンした。
- 例 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。

例 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ノードとして稼働するサーバーを決定します。

## その他の ovstart および ovstop オプション

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、NNMi は以下のコマンドを実行します。

- ovstart: **nnmcluster -daemon**
- ovstop: **nnmcluster -disable -shutdown**



**ovstop** コマンドを実行すると、NNMi はスタンバイノード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** リファレンスページ、または UNIX のマンページを参照してください。

**ovstop** コマンドに使用する以下のオプションは、アプリケーションフェイルオーバークラスターに構成された 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ープロセスを停止し、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ー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**: アクティブノードとスタンバイノードを停止し、これらをクラスターから削除します。このコマンドは **nnmcluster -halt** と同等です。



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 **shutdown** コマンドを実行すると、**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。これが最適な設定ではない場合もあります。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、**shutdown** 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ノードとスタンバイノードを目的の動作に設定します。詳細については、**nnmcluster** リファレンスページ、または UNIX のマンページ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 ではそのたびに以下のいずれかのインシデントが生成されます。

- **NnmClusterStartup:** NNMi クラスタは、アクティブノードがない状態で開始されました。したがって、このノード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ノードの障害が検出されました。そのため、スタンバイノードがアクティブノードになり、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。

---

## フェイルオーバー後、元の設定に戻る

アクティブノードで障害が発生し、スタンバイノードがアクティブノードとして機能しているとします。以前のアクティブノードで問題を解決したら、目的のアクティブノードでコマンド **`nnmcluster -acquire`** を実行し、元の設定に戻ります。詳細については、`nnmcluster` リファレンスページ、または UNIX のマンページを参照してください。

## NNM iSPI およびアプリケーションフェイルオーバー

NNMi と一緒に Smart Plug-in (iSPI) を導入する場合、以下の要件を満たすと iSPI 用のアプリケーションフェイルオーバー機能を使用できます。

- NNM iSPI は NNMi 管理サーバーで動作する。
- NNM iSPI は、NNMi と同じ組み込みデータベースインスタンスを使用する。

NNM iSPI Performance for Metrics および NNM iSPI Performance for Traffic には、この説明は該当しません。NNMi アプリケーションフェイルオーバー機能を設定する場合は、これらの iSPI を専用サーバーにインストールする必要があります。この場合、iSPI は、フェイルオーバーが発生すると、新しい NNMi 管理サーバーに自動的に接続します。NNMi アプリケーションフェイルオーバー設定の一環として、クラスターの各 NNMi 管理サーバーに、NNM iSPI Performance for Metrics または NNM iSPI Performance for Traffic 用のイネーブルメントスクリプトを実行します。

詳細については、NNM iSPI Performance for Metrics、NNM iSPI Performance for QA、または NNM iSPI Performance for Traffic ヘルプの「アプリケーションフェイルオーバーのサポート」を参照してください。

### NNM iSPI のインストールに関する情報

アプリケーションフェイルオーバークラスターのすでに一部である NNMi 管理サーバーに NNM iSPI をインストールするには、以下の手順を実行します。

- 1 万に備えて、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** スクリプトを実行します。詳細については、「[ベストプラクティス：既存の設定を保存](#)」(34 ページ)を参照してください。
- 2 万に備えて、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、「[バックアップ領域](#)」(395 ページ)を参照してください。
- 3 組み込みデータベースのみ：万に備えて、アクティブ NNMi 管理サーバーで、**nnmcluster -dbsync** コマンドを実行し、コマンドが完了するまで待ちます。
- 4 スタンバイ NNMi 管理サーバーで、以下のコマンドを実行します。
 

```
nnmcluster -shutdown
```
- 5 スタンバイ NNMi 管理サーバーの以下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
```
- 6 **com.hp.ov.nms.cluster.name** オプションをコメントアウトし、ファイルを保存します。
- 7 スタンバイ NNMi 管理サーバーで **ovstart** コマンドを実行します。すると、スタンダアロン (クラスターに属しない) 状態の NNMi サービスが表示されます。
- 8 『iSPI インストールガイド』で説明されているとおりに、スタンバイ NNMi 管理サーバーに NNM iSPI をインストールします。
- 9 アクティブ NNMi 管理サーバーで **nnmcluster -halt** コマンドを実行します。

- 10 アクティブ NNMi 管理サーバーの以下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
- 11 `com.hp.ov.nms.cluster.name` オプションをコメントアウトし、ファイルを保存します。
- 12 アクティブ NNMi 管理サーバーで `ovstart` コマンドを実行します。すると、スタンバイ (クラスターに属しない) 状態の NNMi サービスが表示されます。
- 13 『iSPI インストールガイド』で説明されているとおりに、アクティブ NNMi 管理サーバーに NNMi iSPI をインストールします。
- 14 アクティブおよびスタンバイ NNMi 管理サーバーの両方で、以下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
- 15 `com.hp.ov.nms.cluster.name` オプションをコメント解除し、各ファイルを保存します。
- 16 アクティブ NNMi 管理サーバーで `ovstart` コマンドを実行します。
- 17 アクティブ NNMi 管理サーバーがクラスターの最初のアクティブノードになるまで数分待ちます。アクティブ NNMi 管理サーバーで `nnmcluster -display` コマンドを実行し、表示された結果で、`ACTIVE_NNM_STARTING` または `ACTIVE_SomeOtherState` の「ACTIVE」という語を検索します。アクティブ NNMi 管理サーバーがアクティブノードであることを確認するまで手順 18 に進まないでください。
- 18 スタンバイ NNMi 管理サーバーで `ovstart` コマンドを実行します。

---

## 統合アプリケーション

HP ソフトウェア製品または第三者の製品が NNMi に統合された場合、統合に対する NNMi アプリケーションフェイルオーバー機能の影響は、製品が NNMi と通信する方法によって異なります。詳細については、適切な統合ドキュメントを参照してください。

統合製品の設定に NNMi 管理サーバーに関する情報が必要な場合は、以下の情報が適用されます。

- 将来的に必要であれば、統合する製品の設定で NNMi 管理サーバーの情報を更新できます。詳細については、適切な統合ドキュメントを参照してください。
  - 機能停止が一時的なものである場合、サーバー X が復旧した後に統合する製品の使用を再開できます。サーバー X のサービスを復旧するには、以下の手順に従います。
- 1 サーバー X で以下のコマンドを実行します。

```
nnmcluster -daemon
```

サーバー X がスタンバイ状態でクラスターに参加します。

- 2 サーバー X で以下のコマンドを実行します。

```
nnmcluster -acquire
```

サーバー X はアクティブ状態になります。

元のサーバー X がより長期に渡って機能停止となる可能性がある場合は、統合する製品内で、NNMi 管理サーバーの IP アドレスを更新できます。[IP アドレス] フィールドの変更方法については、統合する製品のドキュメ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の無効化

アプリケーションフェイルオーバーを設定し、数日間使用した後に、完全に無効化するとします。以下の情報は、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ーに構成された、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、以下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** コマンドを実行します。
- 2 アクティブ NNMi 管理サーバーで **nnmcluster -shutdown** コマンドを実行します。
- 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ちます。
- 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** コマンドを実行します。
- 5 表示された結果で、ACTIVE\_NNM\_RUNNING ステータスを検索します。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** コマンドを実行します。
- 7 DAEMON プロセスがなくなるまで、新しいアクティブ (以前のスタンバイ) で **nnmcluster -display** コマンドを繰り返し実行します。
- 8 クラスターに構成されている両方の NNMi 管理サーバーで、以下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
```
- 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アウトし、各ファイルを保存します。
- 10 両方の NNMi 管理サーバーの以下のファイルを編集します。
  - Windows:
 

```
%NnmDataDir%\shared\nnm\databases\Postgres\postgresql.conf
```
  - UNIX: 

```
$NnmDataDir/shared/nnm/databases/Postgres/postgresql.conf
```

- 11 以下の行を削除します。これらは、アプリケーションフェイルオーバーにより自動的に追加されたものです。これらの行の例を以下に示します。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 = ...
archive_timeout = 900
max_wal_senders = 4
archive_mode = 'on'
wal_level = 'hot_standby'
hot_standby = 'on'
wal_keep_segments = 500
listen_addresses = 'localhost,16.78.61.68'
```

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、各サーバーで以下の手順を実行します。
- a HP NNM Cluster Manager の [スタートアップの種類] を [無効] に設定します。
  - b HP Openview 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。

```
NnmDataDir/tmp/postgresTriggerFile
```

- 14 以前のアクティブ **NNMi** 管理サーバーのみに **ovstart** コマンドを実行します。アプリケーションフェイルオーバー構成では、このサーバーは恒久 **NNMi** ライセンスを取得している **NNMi** 管理サーバーです。
- 15 以前のスタンバイサーバーで非商用ライセンスを使用している場合は、その **NNMi** 管理サーバーで **ovstart** コマンドを実行しないでください。アプリケーションフェイルオーバー構成では、このサーバーは、非商用ライセンスを取得している **NNMi** 管理サーバーです。この **NNMi** 管理サーバーをスタンドアロンサーバーとして実行するには、恒久ライセンスを購入し、インストールする必要があります。詳細については、「**NNMi** のライセンス」(123 ページ) を参照してください。
- 16 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** 管理サーバーから以下のディレクトリを削除します。

- **Windows:** %NnmDataDir%\shared\%nnm%\databases\Postgres\_standby
- **UNIX:** \$NnmDataDir/shared/nnm/databases/Postgres\_standby



このディレクトリはデフォルトのディレクトリで、nms-cluster.properties ファイルにある com.hp.ov.nms.cluster.archivedir パラメーターの値です。この手順では、この値が変更されていないことを前提としています。nms-cluster.properties ファイルの com.hp.ov.nms.cluster.archivedir パラメーターの値を変更した場合は、変更後の新しい値に相当するディレクトリを削除します。

- 17 スタンバイおよびアクティブ **NNMi** 管理サーバーから以下のディレクトリを削除します。
- **Windows:** %NnmDataDir%\shared\%nnm%\databases\Postgres.OLD
  - **UNIX:** \$NnmDataDir/shared/nnm/databases/Postgres.OLD



## 管理タスクおよびアプリケーションフェイルオーバー

以下は、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ています。

### アプリケーションフェイルオーバーおよび NNMi 9.20 へのアップグレード

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旧バージョンの NNMi をアップグレードする場合、使用しているデータベースに応じて後出の適切なセクションの手順に従ってください。

### 組み込みデータベース

アプリケーションフェイルオーバーと組み込みデータベースの使用を設定している NNMi 管理サーバーをアップグレードするには、以下の手順を実行します。

- 1 万に備えて、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` スクリプトを実行します。詳細については、「[ベストプラクティス: 既存の設定を保存](#)」(34 ページ)を参照してください。

万に備えて、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、「[バックアップ領域](#)」(395 ページ)を参照してください。

- 2 アクティブ NNMi 管理サーバーで以下の手順を実行します。以下の `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、305 ページの[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。表示される情報に以下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` コマンドを実行します。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 管理サーバーで以下の手順を実行します。

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ことがあります。
  - 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 以下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します。
- a 以下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` パラメーターをコメントアウトします。
  - 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します。
- 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、[RUNNING] 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 『*HP Network Node Manager i Software インタラクティブインストールガイド*』の指示に従い、スタンバイ **NNMi** 管理サーバーを **NNMi 9.20** にアップグレードします。



スタンバイ **NNMi** 管理サーバーにインストールしたすべての **iSPI** を、**NNMi 9.20** をサポートする **iSPI** バージョンにアップグレードする必要があります。

以前のアクティブ **NNMi** 管理サーバーが **NNMi 9.0x** または **9.1x** を実行し、以前のスタンバイ **NNMi** 管理サーバーが **NNMi 9.20** 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。これらの **NNMi** 管理サーバーを数時間以上この設定のままにしないでください。この設定は、以前のスタンバイノードにインストールした非商用ライセンスの違反になります。

アップグレードを完了してこの状況を解決するには、以前のアクティブノードを **NNMi 9.20** にアップグレードする時間を選択します。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ー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** コマンドを実行します。
- 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** 管理サーバーで以下の手順を実行します。
  - 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ことがあります。
  - 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 以下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します。
- a 以下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` パラメーターをコメントアウトします。

『*HP Network Node Manager i Software インタラクティブ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 9.20** にアップグレードします。



以前のアクティブ **NNMi** 管理サーバーにインストールしたすべての **iSPI** を、**NNMi 9.20** をサポートする **iSPI** バージョンにアップグレードする必要があります。

2 つのサーバーで **NNMi 9.20** 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。

- 11 以前のアクティブ **NNMi** 管理サーバーで以下の手順を実行します。
- a **ovstop** コマンドを実行します。
  - b 以下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 `com.hp.ov.nms.cluster.name` パラメーターの値を入力します。



**NNMi** のアップグレード手順では、コメントアウトされたプロパティは保持されません。したがって、クラスター名は再入力する必要があります。

- d `com.hp.ov.nms.cluster.name` パラメーターのコメントを解除します。
- e 変更を保存します。

12 **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します。これがアクティブノードになりました。

13 アクティブノードを使用してネットワークを監視するように、オペレーターに指示します。



以前のスタンバイ **NNMi** 管理サーバーは、305 ページの **手順 8** から 306 ページの **手順 12** 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。

- 14 以前のスタンバイ **NNMi** 管理サーバーで以下の手順を実行します。
- a **ovstop** コマンドを実行します。
  - b 以下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 `com.hp.ov.nms.cluster.name` パラメーターの値を入力します。
  - d `com.hp.ov.nms.cluster.name` パラメーターのコメントを解除します。

e 変更を保存します。

- 15 **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します。

この NNMi 管理サーバーはスタンバイノードになり、アクティブノードからデータベースのコピーを受信します。

- 16 NNM iSPI Performance for QA、NNM iSPI Performance for Metrics、または NNM iSPI Performance for Traffic をインストールし、アプリケーションフェイルオーバー機能を使用しており、さらに上記のアップグレードプロセスを完了した場合は、アクティブおよびスタンバイ NNMi 管理サーバーの各 NNM iSPI で NNM iSPI イネーブルメントスクリプトを実行します。NNM iSPI イネーブルメントスクリプトへのパスは次のとおりです。

- Windows: %NNMInstallDir%\bin\%nmenableperfspi.ovpl
- UNIX: /opt/OV/bin/nmenableperfspi.ovpl

## Oracle データベース



2 つの NNMi 管理サーバーを同一の Oracle データベースに同時に接続することはできないため、NNMi 管理サーバーは個別にアップグレード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と Oracle データベースの使用を設定している NNMi 管理サーバーをアップグレードするには、以下の手順を実行します。

- 1 万に備えて、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** スクリプトを実行します。詳細については、「[ベストプラクティス: 既存の設定を保存](#)」(34 ページ)を参照してください。
- 2 万に備えて、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、「[バックアップ領域](#)」(395 ページ)を参照してください。
- 3 スタンバイ NNMi 管理サーバーで **nnmcluster -halt** コマンドを実行します。アクティブおよびスタンバイ NNMi 管理サーバーの両方で、すべての **nnmcluster** プロセスをシャットダウンします。
- 4 アクティブまたはスタンバイ NNMi 管理サーバーのどちらでも **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 管理サーバーで以下の手順を実行します。
  - 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。
- 5 以下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します。
  - a 以下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 パラメーターをコメントアウトします。
  - c 変更を保存します。

- 6 『*HP Network Node Manager i Software インタラクティブインストールガイド*』の指示に従い、スタンバイ NNMi 管理サーバーを NNMi 9.20 にアップグレードします。



スタンバイ NNMi 管理サーバーにインストールしたすべての iSPI を、NNMi 9.20 をサポートする iSPI バージョンにアップグレードする必要があります。

以前のアクティブ NNMi 管理サーバーに NNMi 9.0x または 9.1x がインストールされ、以前のスタンバイ NNMi 管理サーバーに NNMi 9.20 がインストールされています。

- 7 以前のスタンバイ NNMi 管理サーバーで **ovstop** コマンドを実行し、NNMi 管理サーバーを Oracle データベースから切断します。
- 8 以下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します。
- a 以下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 パラメーターをコメントアウトします。
- 9 『*HP Network Node Manager i Software インタラクティブ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 9.20 にアップグレードします。



以前のアクティブ NNMi 管理サーバーにインストールしたすべての iSPI を、NNMi 9.20 をサポートする iSPI バージョンにアップグレードする必要があります。

2 つのサーバーに NNMi 9.20 がインストールされています。

- 10 以前のアクティブ NNMi 管理サーバーで以下の手順を実行します。
- a **ovstop** コマンドを実行します。
  - b 以下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 com.hp.ov.nms.cluster.name パラメーターの値を入力します。
- 11 以前のアクティブ NNMi 管理サーバーで **ovstart** コマンドまたは **nnmcluster-daemon** を実行します。これがアクティブノードになりました。
- 12 以前のスタンバイ NNMi 管理サーバーで以下の手順を実行します。
- f 以下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
  - g com.hp.ov.nms.cluster.name パラメーターの値を入力します。
  - h com.hp.ov.nms.cluster.name パラメーターのコメントを解除します。



NNMi のアップグレード手順では、コメントアウトされたプロパティは保持されません。したがって、クラスター名は再入力する必要があります。

- d com.hp.ov.nms.cluster.name パラメーターのコメントを解除します。
- e 変更を保存します。

i 変更を保存します。

- 13 **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します。

NNMi 管理サーバーがスタンバイノードになります。

- 14 NNM iSPI Performance for QA、NNM iSPI Performance for Metrics、または NNM iSPI Performance for Traffic をインストールし、アプリケーションフェイルオーバー機能を使用しており、さらに上記のアップグレードプロセスを完了した場合は、アクティブおよびスタンバイ NNMi 管理サーバーの各 NNM iSPI で NNM iSPI イネーブルメントスクリプトを実行します。NNM iSPI イネーブルメントスクリプトへのパスは次のとおりです。

- Windows: %NNMInstallDir%\bin\%nnmenableperfspi.ovpl
- UNIX: /opt/OV/bin/nnmenableperfspi.ovpl

## アプリケーションフェイルオーバーおよび NNMi パッチ

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、以下のいずれ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。

### 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、以下の手順を実行します。

- 1 万に備えて、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** スクリプトを実行します。詳細については、「ベストプラクティス: 既存の設定を保存」(34 ページ) を参照してください。
- 2 万に備えて、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、「バックアップ領域」(395 ページ) を参照してください。
- 3 万に備えて、アクティブ NNMi 管理サーバーで、以下の手順を実行します。
  - a **nnmcluster** コマンドを実行します。
  - b 組み込みデータベースのみ: NNMi に求められたら、「**dbsync**」と入力し、[Enter] キーを押します。表示される情報に以下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。

- 4 アクティブ NNMi 管理サーバーで **nnmcluster -halt** コマンドを実行します。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以下の手順を実行します。
  - a **nnmcluster** コマンドを実行します。
  - b (SELF) とマークされているもの以外に **nnmcluster** ノードが存在しないことを確認します。
  - c **exit** または **quit** を実行して、手順 a で開始したインタラクティブ **nnmcluster** プロセスを停止します。
- 6 アクティブ NNMi 管理サーバーで、**nms-cluster.properties** ファイルの **com.hp.ov.nms.cluster.name** パラメーターをコメントアウトします。
  - a 以下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** パラメーターをコメントアウトします。
  - c 変更を保存します。
- 7 パッチとともに提供された指示に従って、アクティブ NNMi 管理サーバーに NNMi パッチを適用します。
- 8 アクティブ NNMi 管理サーバーで、**nms-cluster.properties** ファイルの **com.hp.ov.nms.cluster.name** パラメーターをコメント解除します。
  - a 以下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** パラメーターのコメントを解除します。
  - c 変更を保存します。
- 9 アクティブ NNMi 管理サーバーで **ovstart** コマンドを実行します。
- 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します。
- 11 **nnmcluster -dbsync** コマンドを実行して、新しいバックアップを作成します。
- 12 310 ページの手順 a から 310 ページの手順 c に示されているように、スタンバイ NNMi 管理サーバーで、**nms-cluster.properties** ファイルの **com.hp.ov.nms.cluster.name** パラメーターをコメントアウトします。
- 13 NNMi パッチをスタンバイ NNMi 管理サーバーに適用します。
- 14 310 ページの手順 a から 310 ページの手順 c に示されているように、スタンバイ NNMi 管理サーバーで、**nms-cluster.properties** ファイルの **com.hp.ov.nms.cluster.name** パラメーターをコメント解除します。
- 15 スタンバイ NNMi 管理サーバーで **ovstart** コマンドを実行します。



- 16 NNM iSPI Performance for QA、NNM iSPI Performance for Metrics、または NNM iSPI Performance for Traffic をインストールし、アプリケーションフェイルオーバー機能を使用しており、さらに上記のパッチプロセスを完了した場合は、アクティブおよびスタンバイ NNMi 管理サーバーの各 NNM iSPI に NNM iSPI イネーブルメントスクリプトを実行します。

## 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 (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、以下の手順を実行します。

- 1 万に備えて、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** スクリプトを実行します。詳細については、「[ベストプラクティス: 既存の設定を保存](#)」(34 ページ)を参照してください。
- 2 万に備えて、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、「[バックアップ領域](#)」(395 ページ)を参照してください。
- 3 ノードのいずれかで **nnmcluster** コマンドを実行します。
- 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** を入力します。



**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。Oracle データベースを使用するように設定された NNMi 管理サーバーで、**dbsync** オプションを使用しないでください。

- 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、次に進んでください。
- 6 **nnmcluster** を終了または中断させます。
- 7 以下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ーを停止します。  
**nnmcluster -shutdown**
- 8 以下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、次に進みます。
  - postgres
  - ovjboss
- 9 **nnmcluster** プロセスが終了しているのを確認してから、次に進みます。**nnmcluster** プロセスが終了していない場合、他に方法がなければ、**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、以下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

- 11 行の先頭に **#** を入れてクラスター名をコメントアウトし、変更を保存します。  
**#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します。
- 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す。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 管理サーバーで以下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ーをシャットダウンします。  
**nnmcluster -halt**
- 15 `nnmcluster` プロセスの終了を確認します。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ー名をコメント解除します。
- 17 以下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ーを起動します。  
**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します。
- 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、オフラインです。以下の手順を実行して、(スタンバイ NNMi 管理サーバーとして) クラスターに復帰させます。
  - 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をコメント解除します。
  - b 以下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 管理サーバーで以下のコマンドを実行します。  
**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 管理サーバーで以下のコマンドを実行します。  
**nnmcluster -acquire**
- 22 NNM iSPI Performance for QA、NNM iSPI Performance for Metrics、または NNM iSPI Performance for Traffic をインストールし、アプリケーションフェイルオーバー機能を使用しており、さらに上記のパッチプロセスを完了した場合は、アクティブおよびスタンバイ NNMi 管理サーバーの各 NNM iSPI に NNM iSPI イネーブルメントスクリプトを実行します。



## アプリケーションフェイルオーバーおよび NNMi 管理サーバーの再起動

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。スタンバイとアクティブの両方の NNMi 管理サーバーを再起動する場合は、アクティブ NNMi 管理サーバーを先に再起動してください。

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、以下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します。
- 2 NNMi 管理サーバーを再起動します。
  - 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します。

### 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つのリモートノード間の通信障害が解決すると、JGroups は、最も小さい IP アドレスに基づいてコントローラーになるクラスターメンバーを決定します。コントローラーは、アクティブメンバーになるノード(このノードは、必ずコントローラーが実行されるノードになります)を決定します。NNMi は、アクティブメンバーで起動します。この機能は、今後のリリースで変更される可能性があります。

## アプリケーションフェイルオーバーおよび以前のデータベースバックアップから復旧 (組み込みデータベースのみ)

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、元のバックアップから NNMi データベースを復旧するには、以下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 管理サーバーの以下のディレクトリを削除または移動します。
  - Windows: %NnmDataDir%\shared\%nnm%\databases\Postgres\_standby
  - UNIX: \$NnmDataDir/shared/nnm/databases/Postgres\_standby
- 3 アクティブ NNMi 管理サーバーでデータベースを復元します。
  - a 以下のファイルのクラスター名をコメントアウトして変更します。
    - Windows:
 

```
%NnmDataDir%\shared\%nnm%\conf\%props%\nms-cluster.properties
```
    - UNIX: \$NnmDataDir/shared/nnm/conf/props\nms-cluster.properties
  - b 通常どおり、データベースを復旧します。「NNMi データのリストア」(398 ページ)を参照してください。
  - 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。

- d 以下のファイルでクラスター名をコメント解除して変更します。
  - 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** コマンドを実行します。
- 5 アクティブ NNMi 管理サーバーが新しいバックアップを生成するまで待ちます。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、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** コマンドを実行します。スタンバイ NNMi 管理サーバーは新しいバックアップ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。

## 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、クラスター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 組み込みデータベース、データベーストランザクションログ、その他の 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。HP は、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 組み込みデータベースは必ず圧縮されていますが、非常に容量が大きくなり、1GB 以上に増大することがあります。また、NNMi は、ビルトインバックアップインターバル (設定パラメーター、デフォルトは 6 時間) の間に膨大な数のトランザクションログを生成します。各トランザクションログのサイズは数メガバイトから、最大 16MB になることもあります。(これらのファイルは圧縮されています)。以下は、HP の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すぎます。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、スタンバイノード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。たとえば、ハートビート信号が直ちに応答しない場合に、スタンバイノードは、アクティブノード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いくつかあります。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。

マルチサブネット NNMi アプリケーションフェイルオーバーに関する HP の検証では、アクティブサーバーおよびスタンバイサーバーは、それぞれ米国のコロラド州とヒューストンにあり、許容できる帯域幅とレイテンシにより、偽のフェイルオーバーは発生しませんでした。

### アプリケーションフェイルオーバーと NNMi 組み込みデータベース

アプリケーションフェイルオーバーは、NNMi 9.20 の組み込みデータベースと Oracle データベースの両方で動作します。ところが Oracle ではデータベースが NNMi 管理サーバーとは別のサーバーに存在します。Oracle データベースと連動するように NNMi を設定すると、データベースのレプリケーションは行われません。このため、Oracle データベースを使用すると、アプリケーションフェイルオーバーのネットワーク要求が減少します。Oracle でアプリケーションフェイルオーバーを使用しているとき、組み込みデータベースのアプリケーションフェイルオーバーを使用しているときと比較すると、ネットワークではネットワーク要求の 1% 未満しか使用されません。このセクションでは、組み込みデータベースを使用するアプリケーションフェイルオーバーに関連する NNMi トラフィック情報について説明します。

アプリケーションフェイルオーバーに組み込みデータベースを使用するように NNMi を設定すると、NNMi は以下のように動作します。

- 1 アクティブノードがデータベース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します。
- 2 NNMi は、ネットワークを通してこの ZIP ファイルをスタンバイノードに送信します。
- 3 スタンバイノードは ZIP ファイルを展開し、組み込み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します。
- 4 アクティブノードの組み込みデータベースは、データベースアクティビティにより、トランザクションログを生成します。
- 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ノードに送信され、ディスクに蓄積されます。
- 6 スタンバイノードがアクティブになると、NNMi が起動して、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します。これにかかる時間は、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 (サイズが同程度でも、一部のファイルのインポートには別のファイルより時間がかかります)。
- 7 スタンバイノードがすべてのトランザクションログをインポートすると、データベースが使用可能になり、スタンバイノードは残りの NNMi プロセスを開始します。
- 8 元のスタンバイノードがアクティブになり、手順 1 の手順がやり直しされます。

## 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 はアクティブノードからスタンバイノードにネットワークを介して多く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 ファイルとして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ノードのものと同期していることをスタンバイノードが確認できるようにするファイル比較リスト。
- パラメーターの変更 (フェイルオーバーやその他の有効 / 無効) およびクラスターでのノードの追加や除外などの、その他のイベント。

最初の 2 つの項目により、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 99% が生成されます。このセクションでは、この 2 つの項目について詳しく説明します。

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。このアクティビティには以下のデータベースアクティビティが含まれますが、その他にも含まれるものがあります。

- 新しいノードを検出する。
- ノード、インタフェース、VLAN、その他の管理対象オブジェクトに関する属性を検出する。
- 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ーのアクション。

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。たとえば、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により、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り、データベースのエントリが更新されます。

組み込み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：組み込みデータベースは、空の **16 MB** 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、**16 MB** のデータがファイルに書き込まれた時点のいず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可能にします。つまり、完全にアイドル状態のデータベースにより、**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 **16 MB** のファイルは **1 MB** 未満に圧縮されます。満杯の **16 MB** のファイルは約 **8 MB** 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。

## 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

以下のテストでは、**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 **7 MB** 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 **5000** 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 **1.1 GB** で安定し（バックアップ **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ました（スタンバイノードがアクティブになり、以前のアクティブノード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ました（以前のアクティブノードに戻りました）。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、以下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 **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。アクティブノードで `ovstop` コマンドを最初に実行してから、スタンバイノードが完全にアクティブになって NNMi が動作するまでの時間。

結果は表 27 のとおりです。

表 27 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 (MB)	トランザク ションログ の数	の数 (GB)	ノード	インタ フェース	フェイル オーバーの 時間(分)
4	6.5	50	.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	1100	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 がアクティブノードからスタンバイノード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 350 KB/s (1 秒あたりのキロバイト数) または 2.8 MB/s (1 秒あたりのメガビット数) になっています。

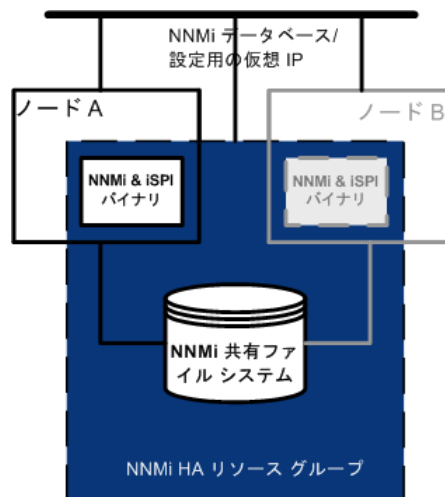
- ▶ このデータには、ハートビート、ファイル整合性チェック、その他のアプリケーションフェイルオーバー通信など、その他のアプリケーションフェイルオーバートラフィックは含まれていません。このデータでは、パケットヘッダーなどのネットワーク I/O のオーバーヘッドも除外されています。このデータには、ネットワークで移動する各ファイルの内容の実ネットワークペイロードのみが含まれます。
- ▶ NNMi のアプリケーションフェイルオーバー環境で生成されるトラフィックは非常に爆発的です。アプリケーションフェイルオーバーでは、5 分ごとにアクティブノードで新しいトランザクションログが識別され、スタンバイノードに送信されます。ネットワークの速度により、スタンバイノード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、ネットワークは比較的アイドル状態となることが多くなります。

アクティブノードとスタンバイノードがロールを切り替えるたびに (スタンバイノードがアクティブになり、アクティブノードがスタンバイになる)、新しいアクティブノード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ノードに送信します。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ノードに送信します。この新しいバックアップがスタンバイノード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。

前述の情報により、組み込み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。



# 高可用性クラスターに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。

NNMi では、別途購入が必要な HA 製品を使って構成される HA クラスタ内で NNMi を実行する設定をサポートするようになりました。ほとんどの NNM Smart Plug-ins (iSPI) も、NNM iSPI NET 診断サーバーを除いて、HA で実行できるようになります。

この章では、HA 環境で実行するように 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ラッパーとなります。HA 製品固有のコマンドの代わりに、以下の手順で説明している NNMi のコマンドを使用できます。

NNMi 管理サーバーにいずれかの NNM iSPI をインストールする場合は、その NNM iSPI のマニュアルも参照してください。

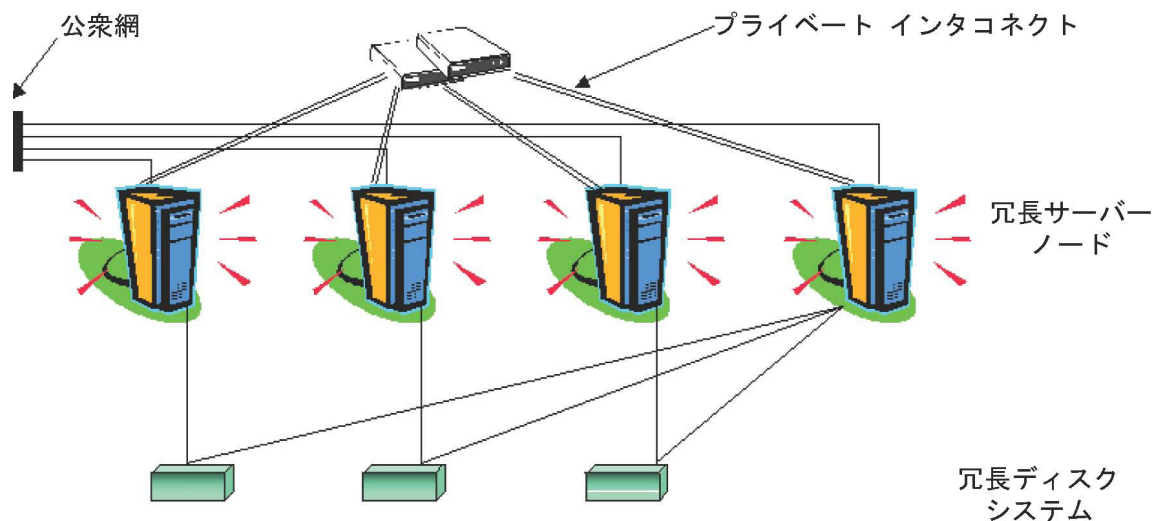
この章には、以下のトピックがあります。

- 「HA の概念」 (320 ページ)
- 「HA 用 NNMi を設定するための前提条件の検証」 (326 ページ)
- 「高可用性の設定」 (328 ページ)
- 「共有 NNMi データ」 (340 ページ)
- 「HA クラスタ内の NNMi のライセンス契約」 (344 ページ)
- 「HA 設定のメンテナンス」 (345 ページ)
- 「HA クラスタ内の NNMi の設定を解除する」 (349 ページ)
- 「HA 下の NNMi のパッチ」 (353 ページ)
- 「HA 下の NNMi を NNMi 9.0x/9.1x から NNMi 9.20 にアップグレードする」 (354 ページ)
- 「HA 設定のトラブルシューティング」 (359 ページ)
- 「HA 設定リファレンス」 (370 ページ)

## HA の概念

クラスターアーキテクチャーには、クラスター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首尾一貫した管理ビューが備わっています。図 26 に、クラスターアーキテクチャーの例を示します。

図 26 高可用性クラスターのアーキテクチャー

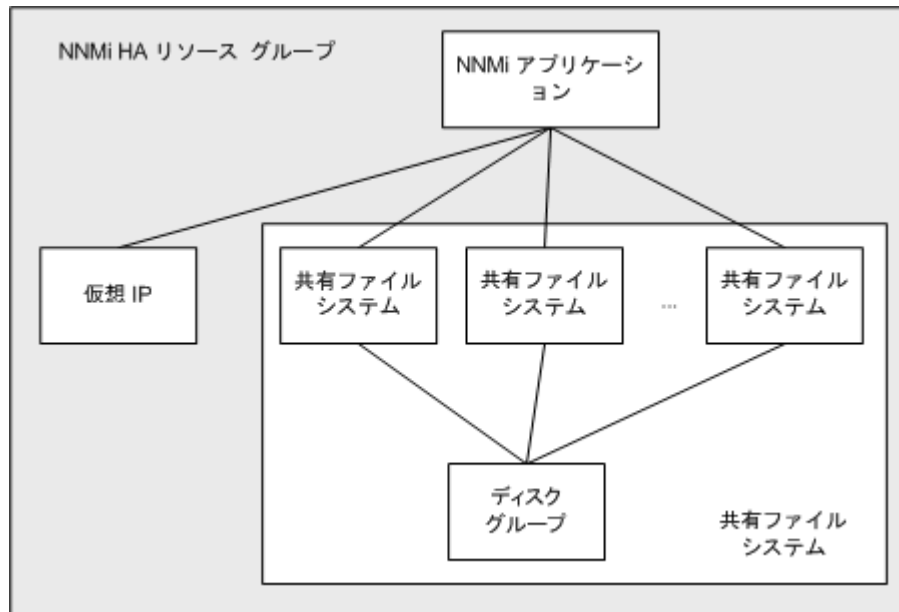


クラスター内の各ノードは、1つ以上の公衆網と1つのプライベートインタコネクト(クラスターノード間のデータ伝送用の通信チャンネル)に接続されます。

HP Serviceguard、Veritas Cluster Server、Microsoft フェールオーバークラスタリング、Microsoft クラスターサービスなどの最新のクラスター環境では、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ー環境で実行することができます。リソースは、クラスター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。図 27 に、HA リソースグループの例を示します。



図 27 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、各種のクラスター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。表 28 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リストします（各 HA 製品のサポート対象バージョンについては、NNMi システムおよびデバイス対応マトリックスを参照してください）。

表 28 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
Microsoft フェールオーバークラスタリング	MSFC	リソースグループ
HP Serviceguard	SG	パッケージ
Veritas Cluster Server	VCS	サービスグループ
Red Hat Cluster Suite	RHCS	サービス

## HA 用語集

表 29 に、一般的な HA 用語の定義を示します。

表 29 一般的な HA 用語

用語	説明
HA リソースグループ	クラスター環境内で (HA 製品下で) 動作するアプリケーションです。HA リソースグループは、同時に、クラスター内のアプリケーションを表すクラスターオブジェクトでもあります。
ボリュームグループ	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ーノード	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ーノードにマウントされます。 プライマリクラスターノードは、通常、最初のアクティブなクラスター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、他のノードをプライマリクラスターノードにできます。
セカンダリクラスターノード	プライマリクラスターノードでの 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ーノード	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ーノード	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ー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ー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ーノードになります。

## NNMi HA クラスターのシナリオ

- ▶ NNMi では、アプリケーションが複数のクラスターノードで実行できるクラスタをサポートしています。詳細については、*nms-ha* のマンページおよび *nnmdatareplicator.ovpl* のリファレンスページ、または UNIX のマンページを参照してください。

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ーノードである 1つのシステムだけです。)

このアプローチは、組み込み型のデータベースと他社製データベースソリューションの場合に有効です。

- ▶ 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ーノードだけです。

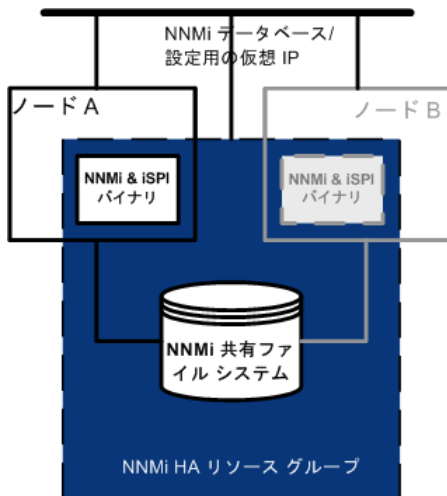
## NNMi のみのシナリオ

図 28 に、NNMi HA クラスターのシナリオを図示します。この図では、NNMi HA リソースグループは、NNMi HA クラスターと同義語です。

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 プログラムと NNM iSPI がすべて含まれています。アクティブなクラスターノードが、共有ディスクのランタイムデータにアクセスします。他の製品は、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。

クラスターに 3 つ以上の NNMi ノードがある場合は、追加ノードには図 28 のノード B と同様の設定を行います。

図 28 NNMi HA クラスタ用の基本的なシナリオ

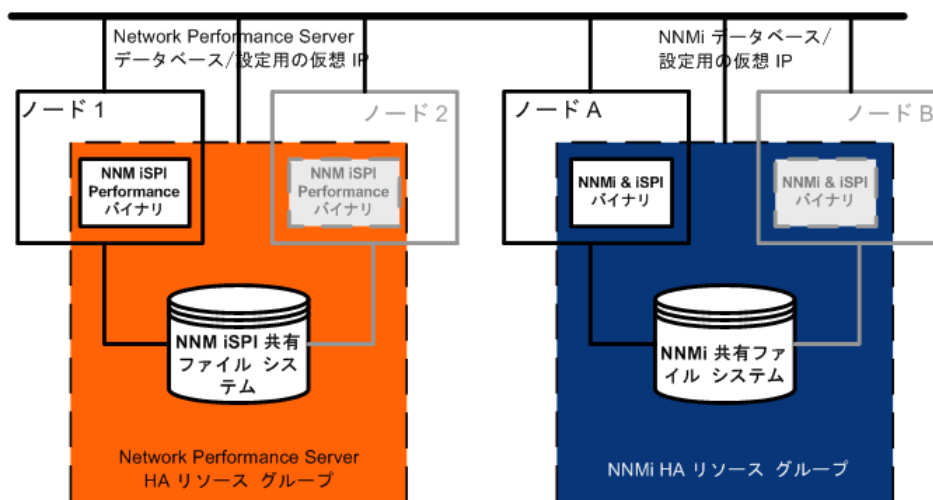


このシナリオの実装方法については、「[HA 用の NNMi の設定](#)」(329 ページ)と「[HA 用の NNM iSPIs の設定](#)」(336 ページ)を参照してください。

### スタンドアロンサーバーシナリオでの NNMi および NNM Performance iSPI

いずれかの NNM Performance iSPI 製品をスタンドアロンサーバーで実行する場合は、[図 29](#) に示すように、NNMi HA クラスタ内で別個の HA リソースグループとして実行されるようこの NNM iSPI を設定できます。NNMi HA リソースグループは、NNMi のみのシナリオで説明したものと同じです。

図 29 スタンドアロンサーバーで NNMi と NNM Performance iSPI を実行する場合の HA



このシナリオの実装方法については、「[HA 用の NNMi の設定](#)」(329 ページ)と「[HA 用の NNM iSPIs の設定](#)」(336 ページ)を参照してください。

スタンドアロンサーバーで実行される NNM Performance iSPI のその他の選択肢は以下のとおりです。

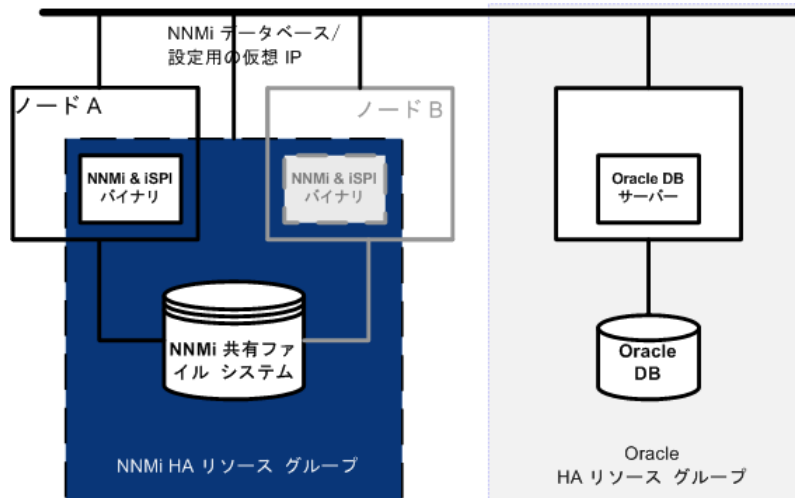
- NNM Performance iSPI を HA を設定していない単一システムで実行します。このアプローチは、NNM iSPI を評価する場合、あるいは、パフォーマンスデータが必ずしも必要ではない環境で使用します。
- NNM Performance iSPI を NNMi 用とは異なる HA クラスタ下で実行するように設定します。この場合は、NNM Performance iSPI の NNMi への依存関係を手動で管理する必要があります。

**NNMiでOracleデータベースを使う場合のシナリオ**

NNMi実装でOracleをメインNNMiデータベースとして使う場合は、Oracleデータベースは、パフォーマンス上の理由で、**図30**のように、独立したサーバーにインストールする必要があります。そのため、NNMi HA クラスターでは、次の2つのHAリソースグループを設定する必要があります。

- NNMi HA リソースグループは、NNMi ノードと、Oracle データベースに格納されないNNMi データ用の共有ディスクで構成します。
- Oracle HA リソースグループは、Oracle データベースサーバーとデータベースディスクで構成します。

**図30 Oracle データベースを使っている NNMi 用の HA**

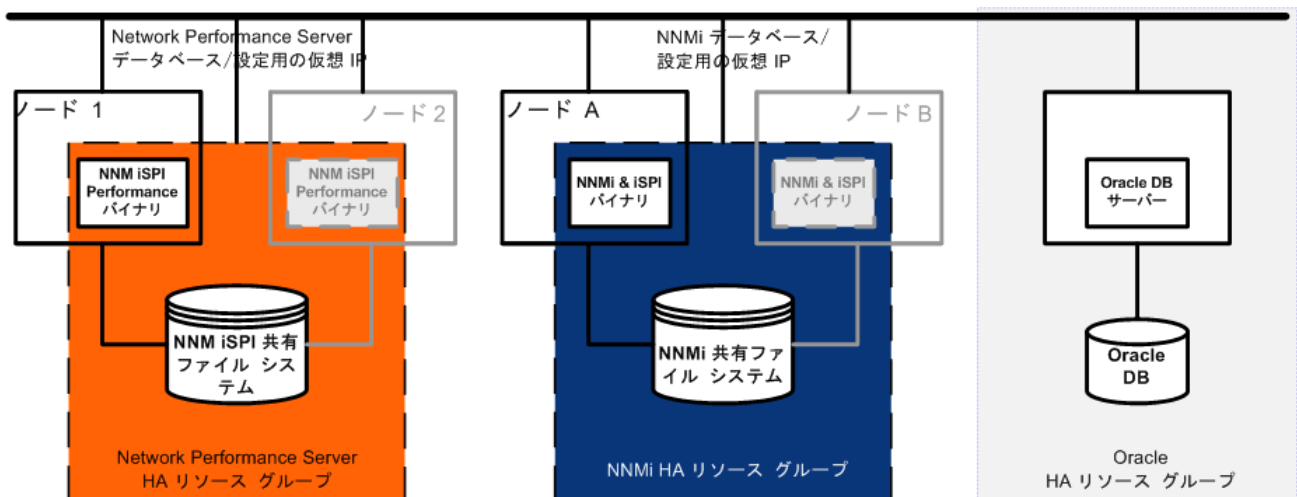


このシナリオの実装方法については、「Oracle 環境での HA 用の NNMi の設定」(338 ページ)と「HA 用の NNM iSPIs の設定」(336 ページ)を参照してください。

**NNMiでOracleデータベースを使用し、NNM Performance iSPIをスタンドアロンサーバーで実行する場合のシナリオ**

NNMi実装でOracleをメインNNMiデータベースとして使用し、いずれかのNNM Performance iSPI製品をスタンドアロンサーバーで実行する場合は、**図31**に示すように、NNMi HA クラスター内に3つのHAリソースグループを設定できます。

**図31 NNMiでOracleデータベースを使用し、NNM Performance iSPIをスタンドアロンサーバーで実行する場合のHA**



このシナリオの実装方法については、「[Oracle 環境での HA 用の NNMi の設定](#)」(338 ページ)と「[HA 用の NNM iSPIs の設定](#)」(336 ページ)を参照してください。

## マンページ

NNMi マンページには、HA 設定について、以下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、これらのマンページはテキストファイルで提供されます。

---

## HA 用 NNMi を設定するための前提条件の検証

HA 用 NNMi を正常に設定できるかどうかは、以下のいくつかの要因に依存します。

- 適切なハードウェア
- HA 製品についての理解
- 系統的な設定方法

HA 用 NNMi の設定を開始する前に、以下の準備手順を実行してください。

- 1 NNMi システムおよびデバイス対応マトリックス の情報を調べて、使用する HA 製品が NNMi でサポートされているかを確認します。
- 2 HA 製品のマニュアルを読み、その製品の機能に精通してから設計上の決定を行います。



HA 製品のマニュアルは頻繁に変更されます。必ず最新版のマニュアルを入手してください。

- 3 NNMi HA クラスターのノードとして含める各システムが以下の要件を満たすことを確認します。
  - HA 製品のマニュアルに記載されているすべての要件に適合する。
  - 少なくとも 2 つのネットワークインタフェースカード (NIC カード) が組み込まれている。



HA 製品、オペレーティングシステム、および NIC カードのマニュアルで調べて、これらの製品を一緒に使用できるかどうか確認してください。

- HA リソースグループの仮想 IP アドレスの使用をサポートする。この IP アドレスは、NNMi ライセンスで使用される IP アドレスです。



MSFC では複数の仮想 IP アドレスが必要であり、1 つは HA クラスタ用で、もう 1 つは各 HA リソースグループ用です。この場合、NNMi HA リソースグループの仮想 IP アドレスは、NNMi ライセンスで使用される IP アドレスです。

- 共有ディスクまたはディスクアレイの使用をサポートする



HA 製品、オペレーティングシステム、およびディスク製造業者のマニュアルで調べて、関連する SCSI カードを含め、これらの製品を一緒に使用できるかどうか確認してください。

- 「NNMi システムおよびデバイス対応マトリックス」で説明される、NNMi のすべての要件に適合する。

4 NNMi HA クラスタでいずれかの NNM iSPI を実行する場合は、HA 設定の追加の前提条件について、該当する NNM iSPI のマニュアルをお読みください。

5 以下の仮想 IP アドレスとホスト名を割り当てます。

- HA クラスタに 1 つの仮想 IP アドレス (MSFC のみ)
- 設定する各 HA リソースグループに 1 つの仮想 IP アドレス

6 任意のシステムから、nslookup コマンドを使用して、手順 5 で割り当てたすべての IP アドレスとホスト名に対して DNS が正しく応答することを確認します。

7 各システムのオペレーティングシステムが、HA 製品と NNMi に適切なバージョンとパッチレベルになっていることを確認します。

8 必要な場合は、HA 製品をインストールします。



Solaris ゾーン環境の場合、HA 製品をグローバルゾーンにインストールします。

9 「共有ディスクの手動準備」(341 ページ)に従って、共有ディスクを準備します。

10 HA 製品用のコマンドを使用して、HA クラスタを設定 (必要な場合) およびテストします。

HA クラスタには、アプリケーションハートビートのチェックやフェイルオーバー起動などの機能が用意されています。HA クラスタ設定には、少なくとも、以下の項目を含める必要があります。

- (UNIX のみ) ssh、remsh、または両方
- (Windows のみ) DNS で解決可能な、HA クラスタ用の仮想 IP アドレス
- DNS で解決可能な、HA クラスタ用の仮想ホスト名
- NNMi 固有の一意のリソースグループ。



NNMi では、必要なすべてのリソースが NNMi HA リソースグループに含まれているが期待されます。不足がある場合は、HA 製品の機能を使用して、NNMi HA リソースグループとその他の HA リソースグループの間の依存関係を管理してください。たとえば、Oracle が別個の HA リソースグループ内で実行されている場合は、HA 製品が NNMi HA リソースグループを起動する前に Oracle HA リソースグループが完全に起動されるように HA 製品を設定します。

- **MSFC: Failover Cluster Management for Windows Server 2008** のクラスター作成ウィザードを使用します。
- **ServiceGuard** の場合 :
  - ノードの `.rhosts` エントリーまたは `.ssh` エントリーを追加します。
  - **HA 製品** (`cmgetconf`、`cmcheckconf`、`cmapplyconf`) を設定します。クラスターの設定の詳細については、**HA 製品**の最新のマニュアルを参照してください。
- **VCS: 不要**。製品のインストールにより **HA** クラスターが作成されました。
- **RHCS: RHCS** のマニュアルの説明に従って、サービス (`cman`、`rgmanager`) を追加します。

NNMi HA リソースグループに入れるリソースのテストの詳細については、「[HA リソーステスト](#)」(360 ページ)を参照してください。

## 高可用性の設定

このセクションでは、NNMi 用の 新規 HA 設定の設定手順を説明します。内容は以下のとおりです。

- 「[HA 用の NNMi 証明書の設定](#)」(328 ページ)
- 「[HA 用の NNMi の設定](#)」(329 ページ)
- 「[HA 用の NNM iSPIs の設定](#)」(336 ページ)
- 「[Oracle 環境での HA 用の NNMi の設定](#)」(338 ページ)

- ▶ Solaris ゾーン環境で NNMi を実行している場合は、この章で説明されている設定プロセスに従う必要はありません。「[Solaris ゾーン環境の HA での NNMi の実行](#)」(284 ページ)を参照してください。
- ▶ RHCS の設定では、HA クラスターの各ノード上で、HA クラスターデーモンとすべてのアプリケーションを完全に再起動する必要があります。これを考慮して、設定作業を計画してください。

### HA 用の NNMi 証明書の設定

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、プライマリノードとセカンダリ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、「[新規証明書を使用するように高可用性を設定する](#)」(136 ページ)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。



## HA用のNNMiの設定

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します。
  - 「プライマリクラスターノードでのNNMiの設定」(333ページ)の手順1～手順9に従って、プライマリ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、設定します。
  - 「プライマリクラスターノードでのNNMiの設定」(333ページ)の手順10～手順15に従って、プライマリノードでこの作業を行います。
  - また、「セカンダリクラスターノードでのNNMiの設定」(335ページ)の説明に従って、セカンダリノードでもこの作業を行います。

1つのHAクラスターノードを、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。プライマリノードを設定します。次にHAクラスター内の残りのすべてのノードをセカンダリノードとして設定します。



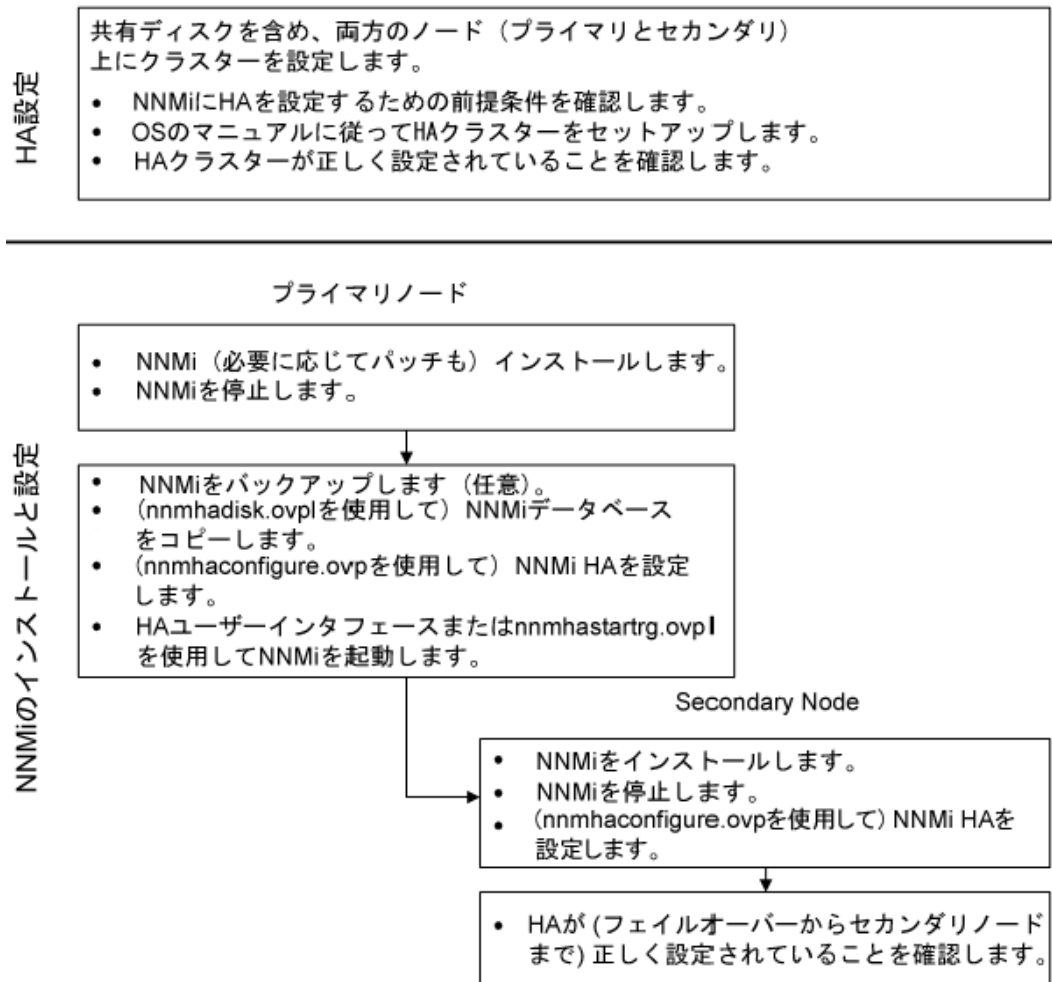
HA用のNNMiの設定は、複数のクラスターノードで同時には行えません。1つのクラスターノードでHA設定プロセスが完了した後、次のクラスターノードでのHA設定プロセスを開始するというように、クラスター環境内のすべてのノードでHA用にNNMiを設定するまで、この作業を繰り返します。



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、ログオンしてNNMiコンソールのセッションを続行する必要があります。

図 32 に、NNMi HA 設定プロセスを示します。

図 32 NNMi HA 設定ワークフロー



▶ HA 設定時にエラーが発生した場合は、以下の手順を実行します。

- 1 nnmhaunconfigure.ovpl コマンドを実行して、HA 環境から NNMi 設定を解除します。
- 2 エラーメッセージが示す状態を修正します。
- 3 nnmhaconfigure.ovpl コマンドを実行して、HA 環境で NNMi 設定を再設定します。

詳細については、[nnmhaunconfigure.ovpl](#) と [nnmhaconfigure.ovpl](#) のリファレンスページ、または UNIX のマンページを参照してください。

## NNMiHA 設定情報

HA 設定スクリプトは、NNMi HA リソースグループに関する情報を収集します。NNMi HA を設定する前に、表 30 にリストされた情報を用意してください。この情報は、使用するオペレーティングシステムまたは HA ソフトウェアに応じて、対話形式で HA スクリプト (nnmhaconfigure.ovpl) を実行するために必要です。

表 30 NNMi HA プライマリノードの設定情報

HA 設定項目	説明
HA リソースグループ	<p>NNMi を含む HA クラスター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。有効な名前の情報については、HA システムプロバイダーの参考資料を参照してください。</p> <p>HA リソースグループ名の入力時に、NNMi は UNIX および Windows システムの次のリソースを生成します。</p> <p>&lt;resource group name&gt;-IP</p> <p>&lt;resource group name&gt;-Mount</p> <p>&lt;resource group name&gt;-App</p> <p>また、Windows システムでは、仮想ホスト名を入力すると次のリソースを生成します。</p> <p>&lt;virtual hostname&gt;</p>
仮想ホストの短い名前	<p>仮想ホストの短い名前です。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。nslookup コマンドで、仮想ホストの短い名前と仮想 IP アドレスを解決できる必要があります。</p> <p><b>注：</b> 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 設定スクリプトのために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、予備のネーミングストラテジ（たとえば、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。</p>
仮想ホストのネットマスク	<p>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。</p>
仮想ホストのネットワークインタフェース	<p>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。次に例を示します。</p> <ul style="list-style-type: none"> <li>• Windows の場合：ローカルエリア接続</li> <li>• HP-UX の場合：lan0</li> <li>• Linux の場合：eth0</li> <li>• Solaris の場合：bge0</li> </ul>
共有ファイルシステムのタイプ	<p>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。使用できる値は次のとおりです。</p> <ul style="list-style-type: none"> <li>• 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 <a href="#">ファイルシステムタイプ</a> の欄を参照してください。</li> <li>• 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 NFS 構成などを使います。HA 設定スクリプトを実行すると、「共有ディスクの手動準備」(341 ページ)のように、共有ディスクが設定されます。</li> </ul>

表 30 NNMi HA プライマリノードの設定情報 ( 続き )

HA 設定項目	説明
ファイルシステムタイプ	<p>(UNIX のみ) 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。</p> <p>以下の共有ディスクフォーマットはテスト済みです。</p> <ul style="list-style-type: none"> <li>• Windows の場合 : 基本型 (「Windows Server での共有ディスク設定についての注記」(343 ページ) を参照); SAN</li> <li>• HP-UX の場合 : vxfs</li> <li>• Linux の場合 : VCS および RHCS には ext2、ext3、および vxfs</li> <li>• Solaris の場合 : vxfs</li> </ul> <p>注 : HA 製品は他のファイルシステムタイプをサポートしています。テストされていない共有ディスクフォーマットを使用する場合は、HA 下で実行するよう NNMi を設定する前にディスクを準備し、次に NNMi HA 設定スクリプトを実行する間に共有ファイルシステムタイプとして none と指定します。</p>
ディスク情報 (使用するオペレーティングシステムに応じて、ディスクグループ、ボリュームグループ、論理ボリュームのいずれか、またはすべて)	<p>NNMi 共有ファイルシステムのディスク情報と関連付けられた名前です。</p> <p>注 : vxfs や lvm などの UNIX プラットフォームのディスクを作成および接続する場合、ディスクグループ、ボリュームグループ、論理ボリュームなどの異なる項目を作成します。これらの項目の名前は、作成時にシステム管理者が割り当てます。NNMi には命名規約はありません。システム管理者に連絡して、会社の命名規約情報を確認してください。</p>
マウントポイント	<p>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。)次に例を示します。</p> <ul style="list-style-type: none"> <li>• Windows: S:¥ 注 : ドライブ名は完全に指定してください。s および s: は受け入れられないフォーマットであり、共有ディスクにアクセスできません。</li> <li>• UNIX: /nnmmount</li> </ul>

## プライマリクラスターノードでの NNMi の設定

プライマリクラスターノードで以下の手順を実行します。

- ▶ メインの NNMi データベースとして Oracle を使用する場合は、まず「Oracle 環境での HA 用の NNMi の設定」(338 ページ)を参照してください。
- ▶ Solaris ゾーン環境で NNMi を実行している場合は、この章で説明されている設定プロセスに従う必要はありません。「Solaris ゾーン環境の HA での NNMi の実行」(284 ページ)を参照してください。

- 1 「HA 用 NNMi を設定するための前提条件の検証」(326 ページ)の作業を完了していない場合は、完了させます。
- 2 NNMi がインストールされていない場合は、NNMi を(最新の統合パッチも含めて)インストールしてから、正しく動作することを確認します。
- 3 この NNMi 管理サーバー上でいずれかの NNM iSPI を実行する場合は、この手順を進める前に「HA 用の NNM iSPIs の設定」(336 ページ)を参照してください。
- 4 `nmbackup.ovpl` コマンド、または別のデータベースコマンドを使用して、NNMi データをすべてバックアップします。次に例を示します。

```
nmbackup.ovpl -type offline -scope all -target nnmi_backups
```

このコマンドの詳細については、「NNMi のバックアップおよびリストアツール」(393 ページ)を参照してください。

- 5 NNMi HA リソースグループ用に、少なくとも 1 つの共有ディスクを含む、ディスクデバイスグループ(および論理ボリューム)を定義します。次に例を示します。
  - MSFC の場合: ディスクの管理を使用して、ディスクのマウントポイントを設定し、ディスクをフォーマットします。
  - Serviceguard の場合:
 

`pvc`、`vg`、および `lv` コマンドを使ってディスクの初期化、ボリュームグループの作成、および論理ボリュームの作成を行います。
  - VCS の場合:
 

`vx`、`vxassist`、および `mkfs` などの VSF コマンドを使用して、ディスクを追加および初期化し、領域ごとにディスクを割り当て、論理ボリュームを作成します。
  - RHCS の場合:
 

`pvc`、`vg`、および `lv` コマンドを使ってディスクの初期化、ボリュームグループの作成、および論理ボリュームの作成を行います。

- ▶ NNMi が正しく開始と停止するために、NNMi では、`/etc/cluster/cluster.conf` ファイルで指定されているクラスターノード名が完全修飾名であるように RHCS クラスターを構成する必要があります。

UNIX オペレーティングシステムの参考 Web サイトは、次のとおりです。

**<http://www.unixguide.net/unixguide.shtml>**

- 6 ディレクトリのマウントポイント（たとえば、s:¥または /nnmmount）を作成し、共有ディスクをマウントします。



設定後、HA 製品はディスクのマウントを管理します。このマウントポイントを使用して、ファイルシステムテーブルを更新しないでください。

- Windows の場合 : Windows Explorer とディスクの管理を使います。
- UNIX:
  - mkdir コマンドおよび mount コマンドを使用します。
  - 共有ディスクのマウントポイントが、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。次に例を示します。

```
ls -l /nnmmount
```

- 7 NNMi を停止します。

```
ovstop -c
```



この HA リソースグループに含めるノードに NNMi がすでにインストールされている場合は、このとき、そのノードで ovstop -c も実行します。

- 8 NNMi データベースを共有ディスクにコピーします。

- Windows:

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

- UNIX:

```
$(NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM ¥
-to <HA_mount_point>
```



データベースの破壊を避けるために、この (-to オプションを指定した) コマンドは 1 回しか実行できません。代替方法については、「すべてのクラスターノードを設定解除した後の HA 用 NNMi の再有効化」(365 ページ) を参照してください。

- 9 (UNIX のみ) 共有ディスクをマウント解除し、ディスクグループを非アクティブ化します。

```
umount <HA_mount_point>
vgchange -a n <disk_group>
```

- 10 NNMi が実行中でないことを確認します。

```
ovstop -c
```

- 11 (RHCS のみ) NNMi カスタムスクリプトを所定の場所にコピーし、HA クラスターデーモンを再起動します。

- a /opt/OV/misc/nnm/ha/NNMscript.sh ファイルを、以下の場所にコピーします。

```
/usr/share/cluster/NNMscript.sh
```

- b /sbin/ccsd プロセスを停止して、再起動します。

## 12 NNMi HA リソースグループを設定します。

- Windows:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaconfigure.ovpl NNM
```

- UNIX:

```
$NnmInstallDir/misc/nnm/ha/nnmhaconfigure.ovpl NNM
```

331 ページの表 30 に、このコマンドに必要な情報を示します。

## 13 (UNIX の場合のみ) NNMi は、デフォルトで、nmhaconfigure.ovpl コマンドを実行したユーザーのロケールで起動します。NNMi のロケールを変更するには、以下のコマンドを実行します。

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl ¥  
-config NNM -set HA_LOCALE <locale>
```

## 14 手順 12 で共有ファイルシステムタイプとして指定した値に応じて、手順が異なります (331 ページの表 30 の共有ファイルシステムのタイプとファイルシステムタイプ)。

- タイプ disk を指定した場合は、nmhaconfigure.ovpl コマンドによって、共有ディスクが設定されています。手順 15 を継続します。
- タイプ none を指定した場合は、「共有ディスクの手動準備」(341 ページ)の説明に従って共有ディスクを準備し、手順 15 に進みます。

## 15 NNMi HA リソースグループを起動します。

- Windows:

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

- UNIX:

```
$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM ¥  
<resource_group>
```

NNMi を正常に起動できなかった場合は、「HA 設定のトラブルシューティング」(359 ページ)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。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。これらのコマンドを使うのは、HA のメンテナンスが目的で、使うことが指示された場合だけです。

## セカンダリクラスターノードでの NNMi の設定

セカンダリクラスターノードでは 1 つのノードごとに順番に以下の手順を実行します。

- 1 「プライマリクラスターノードでの NNMi の設定」(333 ページ)の作業を完了していない場合は、完了させます。
- 2 「HA 用 NNMi を設定するための前提条件の検証」(326 ページ)の作業を完了していない場合は、完了させます。
- 3 NNMi がインストールされていない場合は、NNMi を (最新の統合パッチも含めて) インストールしてから、正しく動作することを確認します。
- 4 「プライマリクラスターノードでの NNMi の設定」(333 ページ)の手順 3 でインストールした NNM iSPI をインストールします。

- 5 NNMi を停止します。

```
ovstop -c
```

- 6 共有ディスクのマウントポイントを作成します (たとえば、S:¥ または /nnmmount)。



このマウントポイントでは、手順 [プライマリクラスターノードでの NNMi の設定の手順 6](#) 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。

- 7 (RHCS のみ) NNMi カスタムスクリプトを所定の場所にコピーし、HA クラスターデーモンを再起動します。

- a /opt/OV/misc/nnm/ha/NNMscript.sh ファイルを、以下の場所にコピーします。

```
/usr/share/cluster/NNMscript.sh
```

- b /sbin/ccsd プロセスを停止して、再起動します。

- 8 NNMi HA リソースグループを設定します。

- Windows: %NnmInstallDir%\misc\%nnm%ha%\nnmhaconfigure.ovpl NNM
- UNIX: \$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。

- 9 設定が正常に行われたことを確認します。

- Windows:

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

- UNIX:

```
$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 リソースグループ用に設定されたノードがすべてリストされます。

- 10 必要に応じて、プライマリノードの NNMi HA リソースグループをオフラインにし、セカンダリノードの NNMi HA リソースグループをオンラインにすることで、設定をテストします。

## HA 用の NNM iSPIs の設定

NNMi 管理サーバー上でいずれかの NNM iSPI を実行する場合は、NNMi を HA 下で実行する設定を行う前に、このセクションをお読みください。

### NNM iSPI Performance for Metrics、NNM iSPI Performance for QA、および NNM iSPI Performance for Traffic

NNM Performance iSPI (NNM iSPI Performance for Metrics、NNM iSPI Performance for QA、および NNM iSPI Performance for Traffic) は、NNMi 管理サーバーかスタンドアロンサーバーにインストールできますが、この 2 つのオプションを組み合わせることはできません。

- NNM Performance iSPI を NNMi 管理サーバー上に配置する場合は、HA 下で実行するように NNMi を設定する前に、この製品をインストールします。



- **NNM Performance iSPI** 製品をスタンドアロンサーバー上に配置する場合は、製品をインストールする前に、**HA** 下で実行するよう **NNMi** を設定します。**NNM iSPI** のインストールプロセス中に、**NNMi HA** リソースグループ仮想ホスト名を **NNMi** 管理サーバー名として指定します。

## **NNM iSPI for MPLS、NNM iSPI for IP Multicast、および NNM iSPI for IP Telephony**

**NNM iSPI for MPLS、NNM iSPI for IP Multicast、および NNM iSPI for IP Telephony** は、**NNMi** 管理サーバーにのみインストールできます。**HA** 下で **NNMi** を実行するよう設定する前に、これらの製品をインストールします。

**HA** 下で実行するよう **NNM iSPI** を設定する場合の詳細については、該当する **NNM iSPI** のマニュアルを参照してください。

## **HA 下で実行中の NNM iSPI ネットワークエンジニアリングツールセットソフトウェアと NNMi**

**NNM iSPI ネットワークエンジニアリングツールセットソフトウェア SNMP** トラップ分析と **Microsoft Visio** エクスポート機能は、**NNMi** と一緒に自動的にインストールされます。これらのツールを **HA** 下で実行するには、これ以上の作業は不要です。

**NNM iSPI NET** 診断サーバーは、**NNMi HA** リソースグループに含めることはできません。このコンポーネントは、**NNMi** 管理サーバーにインストールしないでください。**NNM iSPI NET** 診断サーバーを **NNMi HA** リソースグループ外のシステム上で実行するには、以下の手順を実行します。

- 1 **NNMi HA** リソースグループを完全に設定します。
- 2 **NNM iSPI NET** 診断サーバーを **NNMi HA** リソースグループ外のシステムに、インストールします。**NNM iSPI NET** 診断サーバーのインストールプロセス中に、**NNMi HA** リソースグループ仮想ホスト名を **NNM** サーバーホスト名として指定します。

詳細は、『**NNM iSPI Network Engineering Toolset Software 計画とインストールガイド**』(**NNMiSPI Network Engineering Toolset Software Planning and Installation Guide**) を参照してください。

**NNM iSPI NET** 診断サーバーがすでに **HA** 下で実行される **NNMi** 管理サーバーにインストールされている場合、**HA** 下で実行する **NNMi** を設定する前に **NNM iSPI NET** 診断サーバーをアンインストールします。



**NNM iSPI NET** 診断サーバーをアンインストールすると、既存のレポートがすべて削除されます。



ここで説明するように既存のレポートを保存することもできますが、次の手順はテストされていません。

- 1 **MySQL Workbench** を使って、既存の **nnminet** データベースのバックアップを行う。**MySQL Workbench** は、**dev.mysql.com** のダウンロード領域から入手できます。
- 2 **NNM iSPI NET** 診断サーバーをアンインストールします。
- 3 **HA** 下で **NNMi** を実行するよう、設定します。
- 4 別のシステムに **NNM iSPI NET** 診断サーバーをインストールします。
- 5 フローを実行する前に、**MySQL Workbench** を使って、新しいインストール先にある **nnminet** データベースを復旧します。

## Oracle 環境での HA 用の NNMi の設定

ここでは、Oracle データベースを使っている NNMi を HA 下で実行するための設定作業の概要を説明します。Oracle の設定方法は多数あり、Oracle のリリースによっても異なります。Oracle を HA 下で実行するための設定方法と Oracle HA リソースグループでの NNMi の依存関係の作成方法については、HA 製品マニュアルを参照してください。Oracle の Web サイト ([www.oracle.com](http://www.oracle.com)) でも、HA 製品用の Oracle 設定方法が紹介されています。

### Oracle での NNMi の依存関係

Oracle と NNMi の両方を HA 下で実行する場合は、NNMi HA リソースグループに、Oracle データベースに格納されていない NNMi データ用の共有ディスクを含める必要があります。また、以下の情報を考慮してください。

- HA 製品が依存関係をサポートする場合、各製品を別々の HA リソースグループ内で実行されるように設定するのが推奨される方法です。Oracle HA リソースグループは、NNMi HA リソースグループを起動する前に、完全に起動している必要があります。両方の HA リソースグループが同じ HA クラスタに含まれている場合は、クラスタ設定を変更してリソースグループの起動順序を設定します。それぞれの HA リソースグループが異なる HA クラスタに含まれている場合は、Oracle HA リソースグループに対する NNMi HA リソースグループの依存関係が満たされているかを確認します。
- HA 製品が依存関係をサポートしない場合は、Oracle システムと NNMi システムを NNMi HA リソースグループに含めてください。

### Oracle 環境での HA 用の NNMi の設定

- 1 Oracle を HA 下で実行することを予定している場合は、最初に、以下の手順を実行します。
- 2 NNMi 用の空の Oracle データベースインスタンスを作成します。
- 3 プライマリ NNMi ノードに、(最新の統合パッチも含めて) NNMi をインストールします。インストールの間に、以下を実行します。
  - a [Oracle] データベースタイプを選択してから、[ **プライマリサーバーのインストール** ] を選択します。
  - b Oracle HA リソースグループ用の仮想 IP アドレスまたは仮想ホスト名を指定します。
- 4 プライマリ NNMi ノードで、「**プライマリクラスタードでの NNMi の設定**」(333 ページ)に従って、NNMi を HA 下で実行できるように設定します。
- 5 Oracle HA リソースグループでの NNMi の依存関係を設定します。  
具体的な手順については、HA 製品のマニュアルを参照してください。
- 6 セカンダリ NNMi ノードに、(最新の統合パッチも含めて) NNMi をインストールします。インストールの間に、以下を実行します。
  - [Oracle] データベースタイプを選択してから、[ **セカンダリサーバーのインストール** ] を選択します。

- Oracle HA リソースグループ用の仮想 IP アドレスまたは仮想ホスト名を指定します。
- 7 セカンダリ NNMi ノードで、「セカンダリクラスターノードでの NNMi の設定」(335 ページ)に従って、NNMi を HA 下で実行するように設定します。
  - 8 各セカンダリ NNMi ノードで、手順 6 と手順 7 を繰り返します。

## 共有 NNMi データ

HA 下で実行する NNMi 実装では、HA クラスター内のすべての NNMi ノード間でファイルを共有するために、独立したディスクを使う必要があります。



Oracle をプライマリデータベースとして使っている NNMi の実装でも、共有データ用に独立したディスクを使う必要があります。

### NNMi の共有ディスク内のデータ

ここでは、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 のデータファイルをリストします。

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:**
  - %NnmInstallDir% は、%HA\_MOUNT\_POINT%\NNM\installDir にマッピングされます。
  - %NnmDataDir% は、%HA\_MOUNT\_POINT%\NNM\dataDir にマッピングされます。
- **UNIX:**
  - \$NnmInstallDir は、\$HA\_MOUNT\_POINT/NNM/installDir にマッピングされます。
  - \$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、以下のとおりです。

- **Windows:**
  - %NnmDataDir%\shared\nnm\databases\Postgres  
組み込みデータベース。Oracle データベースを使用する場合は存在しません。
  - %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。
  - %NnmDataDir%\nmsas\NNM\data  
ovjboss で使われるトランザクションストア。
- **UNIX:**
  - \$NnmDataDir/shared/nnm/databases/Postgres  
組み込みデータベース。Oracle データベースを使用する場合は存在しません。
  - \$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。
  - \$NnmDataDir/nmsas/NNM/data  
ovjboss で使われるトランザクションストア。

これらのファイルは、`nnmhadisk.ovpl` コマンドによって、共有ディスク間でコピーされます。この章の手順に従って、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** マンページを参照してください。

## 設定ファイルの複製

NNMi HA の実装では、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 はファイルレプリケーションを管理し、フェイルオーバープロセス中に、アクティブノードからパッシブノードに NNMi 設定ファイルをコピーします。

`nnmdatareplicator.conf` ファイルには、データレプリケーションに含める NNMi のフォルダーとファイルを指定します。

## データレプリケーションの無効化

データレプリケーションは、以下の方法で無効にできます。

- 1 以下のファイルを編集します。
  - Windows: `%NnmDataDir%\$shared¥nnm¥conf¥ov.conf`
  - UNIX: `$NnmDataDir/shared/nnm/conf/ov.conf`
- 2 以下の行を含めます。
 

```
DISABLE_REPLICATION=DoNotReplicate
```
- 3 変更を保存します。
- 4 コマンドプロンプトから `ovstop` を実行します。
- 5 コマンドプロンプトから `ovstart` を実行します。



アクティブノードでファイル(設定ファイルなど)を変更すると、これらのファイルはフェイルオーバーで自動的にスタンバイノードに複製されます。

## 共有ディスクの手動準備

共有ディスクが HP のテスト済みフォーマット (331 ページの表 30 に一覧) である場合は、HA 設定スクリプトによって共有ディスクが準備されるため、以下の手順はスキップしてください。

共有ディスクで、HA 製品によってサポートされているディスクフォーマットなど、未検証の設定が使用されている場合は、共有ディスクを手動で準備する必要があります。HA の設定作業時に、ファイルシステムタイプの値として `none` と入力してから、共有ディスクと NNMi HA リソースグループでの共有ディスクの使用を設定します。



ディスクの設定は、NNMi HA リソースグループを設定する前、または後に実行できます。

共有ディスクを手動で準備するには、以下の手順を実行します。

- 1 「SAN または物理的に接続されたディスクの設定」(342 ページ)の説明に従って、共有ディスクを設定します。
- 2 以下の両方の手順を実行して、ディスクを認識するように NNMi HA リソースグループを設定します。

- 「[ov.conf ファイルへの HA 変数の設定](#)」 (342 ページ)
- 「[NNMi HA リソースグループへの共有ディスクの移動](#)」 (343 ページ)

## SAN または物理的に接続されたディスクの設定

ディスクを、**vxfs** または **ext3** ファイルシステムに接続およびフォーマットします。SAN または物理的に接続されたディスクを設定するには、以下の手順を実行します。

- 1 共有ディスクがシステムブート時にマウントされるように設定されていないことを確認します。  
リソースグループには、共有ディスクをマウントする役割があります。
- 2 以下のように、デバイスを接続します。
  - SAN ディスクの場合、SAN デバイスをネットワークに追加します。  
SAN ディスクの論理ボリュームは、排他モードが使用できる場合には、排他モードである必要があります。
  - 物理的に接続されたディスクの場合、Y ケーブルを使用してディスクを接続します。
- 3 オペレーティングシステムエントリを、すべてのクラスターノード（ディスクグループ、論理ボリューム、ボリュームグループ、およびディスク）に追加します。
  - SAN ディスクの場合、エントリは SAN を参照します。
  - 物理的に接続されたディスクの場合、エントリはディスクハードウェアを参照します。
- 4 331 ページの表 30 にあるディスクフォーマットを使用してディスクをフォーマットします。
- 5 SAN がマウントされていることを確認します。



UNIX システムの参考 Web サイトは、次のとおりです。  
**<http://www.unixguide.net/unixguide.shtml>**

- 6 ディスクをマウント解除してデポートします。
- 7 設定をテストするには、ディスクをリソースグループに追加し、フェイルオーバーを開始します。

## ov.conf ファイルへの HA 変数の設定

NNMi HA リソースグループは、以下の変数を使用して共有ディスクにアクセスします。

- HA\_POSTGRES\_DIR=<HA\_mount\_point>/NNM/dataDir/shared/nnm/databases/Postgres
- HA\_EVENTDB\_DIR=<HA\_mount\_point>/NNM/dataDir/shared/nnm/eventdb
- HA\_NNM\_LOG\_DIR=<HA\_mount\_point>/NNM/dataDir/log
- HA\_JBOSS\_DATA\_DIR=<HA\_mount\_point>/NNM/dataDir/nmsas/NNM/data
- HA\_MOUNT\_POINT=<HA\_mount\_point>

- HA\_CUSTOMPOLLER\_DIR=<HA\_mount\_point>/NNM/dataDir/shared/nnm/databases/custompoller



NNMi HA リソースグループで NNM iSPI を実行する場合は、それらの NNM iSPI ごとに `ov.conf` 変数も設定します。詳細については、該当する NNM iSPI のマニュアルを参照してください。

`ov.conf` ファイルで共有ディスクにアクセスするための製品の変数を設定するには、前述の各変数に対して、以下のコマンドを実行します。

- Windows:

```
%NnmInstallDir%¥misc¥nnm¥ha¥nnmhaclusterinfo.ovpl ¥
-config NNM -set <variable> <value>
```

- UNIX:

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl ¥
-config NNM -set <variable> <value>
```

## NNMi HA リソースグループへの共有ディスクの移動

製品マニュアルに従ってディスク設定ファイルを変更し、共有ディスクを NNMi HA リソースグループに移動します。次に例を示します。



このプロセスを使用して、NIC カードやバックアップディスクなどの他のリソースを NNMi HA リソースグループに追加することもできます。

- MSFC: フェイルオーバー管理を使用して、リソースをリソースグループに追加します。

- ServiceGuard の場合 :

```
/etc/cmcluster/<resource_group>/<resource_group>.cntl
```

- VCS の場合 : ディスクエントリを追加し、  
`/opt/VRTSvcs/bin/hares` コマンドを使って HA 設定ファイルにリンクします。次に例を示します。

- RHCS の場合 :

```
/etc/cluster/cluster.conf
```

## Windows Server での共有ディスク設定についての注記

Microsoft Knowledge Base の文書 237853 によれば、Windows Server 2008 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。正しくディスクを設定するには、以下の Web サイトの情報を参照してください。

- <http://support.microsoft.com/kb/237853>
- [http://www.petri.co.il/difference\\_between\\_basic\\_and\\_dynamic\\_disks\\_in\\_windows\\_xp\\_2000\\_2003.htm](http://www.petri.co.il/difference_between_basic_and_dynamic_disks_in_windows_xp_2000_2003.htm)

## HA クラスター内の NNMi のライセンス契約

HA クラスター内の NNMi を実行するには、NNMi に以下の 2 つのライセンスが必要です。

- 物理クラスターノードのいずれかの IP アドレスが対象になる商用ライセンス
- NNMi HA リソースグループの仮想 IP アドレスが対象になる非商用ライセンス

NNMi ライセンスキーは、共有ディスクで管理されます。このため、各 NNMi HA リソースグループで、別個にライセンス契約された各製品に必要なのは非商用ライセンスキーのみです。

HA クラスターで NNMi のライセンスを設定する場合、アクティブノードのライセンスファイルにある新しい情報で共有ディスクの licenses.txt ファイルを更新する必要があります。HA クラスターで NNMi のライセンスを正しく設定するには、以下の手順を実行します。

HA クラスターで NNMi のライセンスを正しく設定するには、アクティブな NNMi クラスターノードで以下の手順を実行します。

- 1 「**NNMi のライセンス**」(123 ページ)の説明に従って、注文した製品ごとに非商用の恒久ライセンスキーを入手してインストールします。NNMi 管理サーバーの IP ドレスを入力するよう求められたら、NNMi HA リソースグループの仮想 IP アドレスを入力します。
- 2 アクティブノードの licenses.txt ファイルにある新しい情報で、共有ディスクの LicFile.txt ファイルを更新します。以下のいずれかを行います。
  - licenses.txt ファイルが共有ディスクの NNM ディレクトリにある場合は、アクティブノードの LicFile.txt 内の新しいライセンスキーを共有ディスクの licenses.txt に追加します。
  - licenses.txt ファイルが共有ディスクにない場合は、アクティブノードから共有ディスクの NNM ディレクトリ内の licenses.txt に、LicFile.txt をコピーします。

アクティブノードでは、LicFile.txt ファイルが以下の場所にあります。

- **Windows:** <drive>:\Program Files (x86)\HP\HP BTO Software\data\shared\nnm\conf\licensing\LicFile.txt
- **UNIX:** /var/opt/OV/shared/nnm/conf/licensing/LicFile.txt

共有ディスクでは、licenses.txt ファイルの場所は、たとえば以下のとおりです。

- **Windows:** S:\NNM\licenses.txt
- **UNIX:** /nnmount/NNM/licenses.txt



# HA 設定のメンテナンス

## メンテナンスモード

NNMi のパッチまたは更新プログラムを新しいバージョンの NNMi に適用する必要がある場合は、NNMi HA リソースグループをメンテナンスモードにし、処理中のフェイルオーバーを回避します。NNMi HA リソースグループがメンテナンスモードにある場合、ユーザー（またはインストールスクリプト）は必要に応じて、プライマリ（アクティブ）クラスターノード上で `ovstop` コマンドや `ovstart` コマンドを実行できます。



`ovstart` コマンドや `ovstop` コマンドは、セカンダリ（バックアップ）クラスターノードでは絶対に実行しないでください。

## HA リソースグループをメンテナンスモードにする

HA リソースグループをメンテナンスモードにすると、HA リソースグループの監視が無効になります。HA リソースグループ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製品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。

HA リソースグループをメンテナンスモードにするには、アクティブノードで以下のファイルを作成します。

- **Windows:** %NnmDataDir%\%hacluster%\<resource\_group%\maintenance
- **UNIX:** \$NnmDataDir/hacluster/<resource\_group>/maintenance



maintenance ファイルの内容は以下のとおりです。

- HA リソースグループの監視を無効にするには、maintenance ファイルを作成します。このファイルは空にすることもできますし、キーワード `NORESTART` を含めることもできます。
- 設定手順を行っている間に NNMi が開始しないようにするには、maintenance ファイルの最初の行に以下の 1 語のみを記載してください。  
`NORESTART`

## HA リソースグループのメンテナンスモードを解除する

HA リソースグループのメンテナンスモードを解除すると、HA リソースグループの監視が再び有効になります。HA リソースグループの製品を停止すると、HA リソースグループはパッシブなクラスター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を解除するには、以下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します。

```
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クラスターノードであったノードから、maintenance ファイルを削除します。このファイルについては、[HA リソースグループをメンテナンスモードにする](#)を参照してください。

## HA クラスタ内の NNMi のメンテナンス

### 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限り、ovstart コマンドや ovstop コマンドは、使わないでください。通常の実行では、NNMi に用意されている HA コマンドまたは HA 製品の適切なコマンドを使用して、HA リソースグループの起動や停止を行います。

### クラスタ環境で NNMi のホスト名や IP アドレスを変更する

クラスタ環境内のノードは、複数の IP アドレスやホスト名を持つことができます。ノードが別のサブネットのメンバーになった場合は、IP アドレスを変更する必要があります。それにより、IP アドレスや完全修飾ドメイン名が変更されます。

たとえば、UNIX システムでは、IP アドレスと関連ホスト名は、通常、次のいずれかの方法を使って設定されています。

- /etc/hosts
- ドメインネームサービス (DNS)
- ネットワーク情報サービス (HP-UX または Linux では NIS、Solaris では NIS+)

NNMi は、管理対象ノードが参照できるように、NNMi データベース内に管理サーバーのホスト名と IP アドレスを格納します。

ネームサーバーがない環境からネームサーバー (すなわち、DNS や BIND) がある環境に移行した場合は、ネームサーバーが新しい IP アドレスを解決することを確認してください。

ホスト名は、IP ネットワーク内で管理対象ノードを特定するために使われます。ノードには複数の IP アドレスが設定されていることがありますが、ホスト名は特定のノードを指定するために使われます。システムのホスト名は、hostname コマンドを使ったときに返される文字列です。

NNMi HA リソースグループの仮想ホスト名または IP アドレスを変更する場合は、アクティブノードのライセンスファイルにある新しい情報で、共有ディスクの licenses.txt ファイルを更新する必要があります。HA 設定を正しく更新するには、以下の手順を実行します。

NNMi HA リソースグループの仮想ホスト名または IP アドレスを変更するには、アクティブな NNMi クラスタノードで以下の手順を実行します。

- 1 NNMi HA リソースグループの以前の仮想 IP アドレスの非商用恒久ライセンスキーを、NNMi HA リソースグループの新しい仮想 IP アドレスに変換します。



この時点で、新しいライセンスキーをインストールしないでください。

- 2 「HA リソースグループをメンテナンスモードにする」(345 ページ) の説明に従って、NNMi HA リソースグループをメンテナンスモードにします。
- 3 NNMi HA リソースグループを停止します。

- Windows:

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

- UNIX:

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

- 4 NNMi HA リソースグループの IP アドレスまたはノード名を変更します。

- a ov.conf ファイルの NNM\_INTERFACE エントリーを編集して、新しいホスト名または IP アドレスに変更します。
- b ovspmd.auth ファイル内の旧ホスト名を含む行を編集して、新しいホスト名を含むようにします。

ov.conf ファイルと ovspmd.auth ファイルは、以下の場所にあります。

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

- UNIX: \$NnmDataDir/shared/nnm/conf

- 5 NNMi HA リソースグループのノード名を変更した場合、`nnmsetofficialfqdn.ovpl` コマンドを使用して、NNMi HA リソースグループの新しい完全修飾ドメイン名を使用するように、NNMi を設定します。次に例を示します。

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

詳細については、`nnmsetofficialfqdn.ovpl` リファレンスページ、または UNIX のマンページを参照してください。

- 6 新しい IP アドレスを使うように、クラスター設定を変更します。

- MSFC の場合:

Failover Cluster Management で、`<resource_group>` を開きます。

`<resource_group>-ip` をダブルクリックして、[パラメーター] を選択し、新しい IP アドレスを入力します。

- Serviceguard の場合:

アクティブな HA クラスターノードで、`/etc/cmcluster/<resource_group>/<resource_group>.cntl` ファイルを編集して、`IP[0]=<old_IP_address>` を `IP[0]=<new_IP_address>` に置き換えます。(NNMi HA リソースグループを別のサブネットに移動した場合は、`SUBNET[0]=<old_subnet_mask>` も `SUBNET[0]=<new_subnet_mask>` に置き換えます。)そして、`cmapplyconf` を使って残りのシステムをすべてアップデートします。

- VCS の場合:

```
$NnmInstallDir/misc/nnm/ha/nnmhargconfigure.ovpl NNM ¥
<resource_group> -set_value <resource_group>-ip ¥
Address <new_IP_address>
```

- RHCS の場合:

アクティブな HA クラスターノードで、`/etc/cluster/cluster.conf` ファイルを編集して、`ip address="<old_IP_address>"` を `ip address="<new_IP_address>"` に置き換えます。次に、`ccs_tool update /etc/cluster/cluster.conf` を実行して、残りのシステムをすべて更新します。

- 7 「NNMi のライセンス」(123 ページ)の説明に従って、NNMi HA リソースグループの新しい仮想 IP アドレスの非商用恒久ライセンスキーをインストールします。

- 8 アクティブノードの licenses.txt ファイルにある新しい情報で、共有ディスクの LicFile.txt ファイルを更新します。以下のいずれかを行います。
- licenses.txt ファイルが共有ディスクの NNM ディレクトリにある場合は、アクティブノードの LicFile.txt 内の新しいライセンスキーを共有ディスクの licenses.txt に追加します。
  - licenses.txt ファイルが共有ディスクにない場合は、アクティブノードから共有ディスクの NNM ディレクトリ内の licenses.txt に、LicFile.txt をコピーします。

アクティブノードでは、LicFile.txt ファイルが以下の場所にあります。

- **Windows:** <drive>:\Program Files (x86)\HP\HP BTO Software\data\shared\nnm\conf\licensing\LicFile.txt
- **UNIX:** /var/opt/OV/shared/nnm/conf/licensing/LicFile.txt

共有ディスクでは、licenses.txt ファイルの場所は、たとえば以下のとおりです。

- **Windows:** S:\NNM\licenses.txt
- **UNIX:** /nnmount/NNM/licenses.txt

- 9 NNMi HA リソースグループを起動します。

- **Windows:**

```
%NnmInstallDir%\misc\nnm\ha\nnmhastartrg.ovpl NNM ¥
<resource_group>
```
- **UNIX:**

```
$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM ¥
<resource_group>
```

- 10 NNMi を正常に起動できたことを確認します。

```
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。

- 11 「[HA リソースグループのメンテナンスモードを解除する](#)」(345 ページ) の説明に従って、NNMi HA リソースグループのメンテナンスモードを解除します。

## 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、アクティブクラスターノードの NNMi を、パッシブ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。アクティブクラスターノードで以下の手順を実行します。

- 1 「[HA リソースグループをメンテナンスモードにする](#)」(345 ページ) の説明に従って、NNMi HA リソースグループをメンテナンスモードにします。
- 2 NNMi を停止します。

```
ovstop -c
```

## 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、以下の手順を実行して、NNMi と HA 監視を再起動します。

- 1 NNMi を起動します。

```
ovstart -c
```

- 2 NNMi を正常に起動できたことを確認します。

```
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 「[HA リソースグループのメンテナンスモードを解除する](#)」(345 ページ) の説明に従って、NNMi HA リソースグループのメンテナンスモードを解除します。

## NNMi HA クラスタ内のアドオン NNM iSPI のメンテナンス

NNM iSPI は、NNMi に密接にリンクしています。アドオン NNM iSPI を NNMi HA クラスタ内のノードにインストールする場合は、NNMi HA クラスタのメンテナンス手順を使います。

---

## 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、NNMi を実行中の 1 つのノードと、少なくとも、1 つのパッシブ NNMi 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、以下の手順に従います。

- 1 HA クラスタ内のアクティブなノードを特定します。スタンバイで、以下のコマンドを実行します。

- Windows:

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

- UNIX:

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

- 2 各パッシブノードで、HA クラスタから任意のアドオン NNM iSPI の設定を解除します。

詳細については、各 NNM iSPI のマニュアルを参照してください。

- 3 HA クラスター内の任意のノードで、すべてのパッシブノード上のアドオン NNM iSPI が HA クラスターから設定解除されていることを確認します。

- Windows:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl %  
-config NNM -get NNM_ADD_ON_PRODUCTS
```

- UNIX:

```
$NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl %  
-config NNM -get NNM_ADD_ON_PRODUCTS
```

コマンドの出力には、アドオン iSPI の設定が <iSPI\_PM\_Name>[hostname\_list] のフォーマットでリストされます。次に例を示します。

```
PerfSPIHA[hostname1, hostname2]
```

このとき、アクティブノードのホスト名のみが出力に表示されます。パッシブノードのホスト名が出力に表示される場合は、このコマンドの出力にアクティブノードのホスト名のみが表示されるようになるまで、[手順 2](#)を繰り返します。

- 4 各パッシブノードで、HA クラスターから NNMi の設定を解除します。

- Windows:

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM %  
<resource_group>
```

- UNIX:

```
$NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl NNM %  
<resource_group>
```

このコマンドにより、共有ディスクへのアクセスが削除されますが、ディスクグループやボリュームグループは設定解除されません。

- 5 各パッシブノードで、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します。



NNMi HA リソースグループを再設定する予定がない場合は、これら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構いません。

- MSFC の場合 : Windows のエクスプローラーを使って

```
%NnmDataDir%\hacluster%\<resource_group>% フォルダを削除します。
```

- Serviceguard の場合 :

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

- VCS の場合 :

```
rm -rf /var/opt/OV/hacluster/<resource_group>
```

- RHCS の場合 :

```
rm -rf /var/opt/OV/hacluster/<resource_group>
```

- 6 アクティブノードで、HA クラスターからアドオン NNM iSPI の設定を解除します。

詳細については、各 NNM iSPI のマニュアルを参照してください。HA クラスター内の任意のノードで、すべてのノード上のアドオン NNM iSPI が HA クラスターから設定解除されていることを確認します。

- Windows:
 

```
%NnmInstallDir%\misc\nnm\ha\nnmhaclusterinfo.ovpl \  
-config NNM -get NNM_ADD_ON_PRODUCTS
```
- UNIX:
 

```
$(NnmInstallDir/misc/nnm/ha/nnmhaclusterinfo.ovpl) \  
-config NNM -get NNM_ADD_ON_PRODUCTS
```

ホスト名が出力に表示される場合は、このコマンドの出力が iSPI が設定されていないことを示すまで、手順6を繰り返します。

#### 7 アクティブノードで、NNMi HA リソースグループを停止します。

- Windows:
 

```
%NnmInstallDir%\misc\nnm\ha\nnmhastoprg.ovpl NNM \  
<resource_group>
```
- UNIX:
 

```
$(NnmInstallDir/misc/nnm/ha/nnmhastoprg.ovpl) NNM \  
<resource_group>
```

このコマンドでは、共有ディスクへのアクセス権は削除しません。また、ディスクグループやボリュームグループの設定も解除しません。

#### 8 各アクティブノードで、HA クラスタから NNMi の設定を解除します。

- Windows:
 

```
%NnmInstallDir%\misc\nnm\ha\nnmhaunconfigure.ovpl NNM \  
<resource_group>
```
- UNIX:
 

```
$(NnmInstallDir/misc/nnm/ha/nnmhaunconfigure.ovpl) NNM \  
<resource_group>
```

このコマンドにより、共有ディスクへのアクセスが削除されますが、ディスクグループやボリュームグループは設定解除されません。

#### 9 アクティブノードで、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します。



NNMi HA リソースグループを再設定する予定がない場合は、これら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構いません。

- MSFC の場合 : Windows のエクスプローラーを使って  
%NnmDataDir%\hacluster\- Serviceguard の場合 :
 

```
rm -rf /var/opt/OV/hacluster/<resource_group>  
rm -rf /etc/cmcluster/<resource_group>
```
- VCS の場合 :
 

```
rm -rf /var/opt/OV/hacluster/<resource_group>
```
- RHCS の場合 :
 

```
rm -rf /var/opt/OV/hacluster/<resource_group>
```

#### 10 共有ディスクをマウント解除します。



- NNMi HA クラスターの再設定を予定している場合は、ディスクを現状のままに保管できます。
- 共有ディスクを別の目的で使用する場合は、保存するデータをすべてコピーして（「既存データベースを使用した HA 外での NNMi 実行」(352 ページ)の説明を参照）から、HA 製品のコマンドを使用し、ディスクグループとボリュームグループの設定を解除します。

## 既存データベースを使用した HA 外での NNMi 実行

NNMi を HA の外部の任意のノードで既存のデータベースを使って実行する場合は、以下の手順を実行します。

- 1 アクティブノードで（存在する場合）、NNMi が実行中ではないことを確認します。

### ovstop

あるいは、タスクマネージャー (Windows) または ps コマンド (UNIX) を使って、ovspmd プロセスのステータスをチェックします。

- 2 現在のノード (HA の外部で NNMi の実行を予定しているノード) で、NNMi が実行中ではないことを確認します。

### ovstop



データの破壊を避けるために、NNMi のインスタンスが動作中ではないことや、共有ディスクにアクセス中ではないことを確認します。

- 3 (UNIX のみ) ディスクグループをアクティブ化します。たとえば、HP-UX Serviceguard では次を実行します。

```
vgchange -a e <disk_group>
```

- 4 適切なオペレーティングシステムのコマンドを使って、共有ディスクをマウントします。次に例を示します。

- Windows の場合 : [ サーバー マネージャ ] > [ ディスクの管理 ] を使います。
- UNIX: `mount /dev/vgndm/lvndm /nnmmount`

- 5 NNMi のファイルを共有ディスクからローカルディスクにコピーします。

- Windows:

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

- UNIX:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM %  
-from <HA_mount_point>
```

- 6 適切なオペレーティングシステムのコマンドを使って、共有ディスクのマウントを解除します。次に例を示します。

- Windows の場合 : Windows Explorer を使用します。
- UNIX: `umount /nnmmount`

- 7 (UNIX のみ) ディスクグループを非アクティブ化します。たとえば、次を実行します。

```
vgchange -a n <disk_group>
```



- 8 「[NNMi のライセンス](#)」(123 ページ)の説明に従って、この NNMi 管理サーバーの物理 IP アドレスの商用恒久ライセンスキーを取得し、インストールします。
- 9 NNMi を起動します。

```
ovstart -c
```

従来、NNMi HA リソースグループで使われていたデータベースのコピーを使って、NNMi が起動されます。この NNMi 管理サーバーから管理対象としないノードの NNMi 設定を手動で削除します。

---

## HA 下の NNMi のパッチ

パッチを NNMi に適用するには、HA メンテナンスモードで作業します。以下の手順に従ってください。

- 1 HA クラスター内のアクティブなノードを特定します。
  - Windows:
 

```
%NnmInstallDir%\misc\%nm%ha%\nmhaclusterinfo.ovpl %  
-group <resource_group> -activeNode
```
  - UNIX:
 

```
$NnmInstallDir/misc/nm/ha/nmhaclusterinfo.ovpl %  
-group <resource_group> -activeNode
```
- 2 「[HA リソースグループをメンテナンスモードにする](#)」(345 ページ)の説明に従って、アクティブノードで、NNMi HA リソースグループをメンテナンスモードにします。  
NORESTART キーワードを組み込みます。
- 3 「[HA リソースグループをメンテナンスモードにする](#)」(345 ページ)の説明に従って、すべてのパッシブノードで、NNMi HA リソースグループをメンテナンスモードにします。  
NORESTART キーワードを組み込みます。
- 4 アクティブノードで、以下の手順を実行します。
  - a NNMi を停止します。
 

```
ovstop -c
```
  - b ディスクコピーを実行して、共有ディスクをバックアップします。
  - c オプション。nnmbackup.ovpl コマンドまたはその他のデータベースコマンドを使って、NNMi データをすべてバックアップします。次に例を示します。
 

```
nnmbackup.ovpl -type offline -scope all -target nnmi_backups
```

 このコマンドの詳細については、「[NNMi のバックアップおよびリストアツール](#)」(393 ページ)を参照してください。
  - d 該当する NNMi および NNM iSPI のパッチをシステムに適用します。
  - e NNMi を起動します。

```
ovstart -c
```

- f NNMi を正常に起動できたことを確認します。

```
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。

- 5 各パッシブノードで、該当するパッチをシステムに適用します。



**ovstart** コマンドや **ovstop** コマンドは、セカンダリ (バックアップ) クラスターノードでは絶対に実行しないでください。

- 6 「[HA リソースグループのメンテナンスモードを解除する](#)」(345 ページ) の説明に従って、すべてのパッシブノードで、NNMi HA リソースグループをメンテナンスモードから解除します。
- 7 アクティブノードで、「[HA リソースグループのメンテナンスモードを解除する](#)」(345 ページ) の説明に従って、NNMi HA リソースグループをメンテナンスモードから解除します。

## HA 下の NNMi を NNMi 9.0x/9.1x から NNMi 9.20 にアップグレードする

環境に応じて、適切な手順に従ってください。

- 「サポートされるすべてのオペレーティングシステムでの組み込みデータベースを使用した NNMi のアップグレード」(354 ページ)
- 「サポートされるすべてのオペレーティングシステムでの Oracle を使用した NNMi のアップグレード」(358 ページ)

### サポートされるすべてのオペレーティングシステムでの組み込みデータベースを使用した NNMi のアップグレード



NNMi 9.10 では、Linux オペレーティングシステムで、Serviceguard はサポートされなくなりました。NNMi が現在 Serviceguard HA で実行中の場合、このセクションの手順は使用できません。代わりに、「[HA クラスター内の NNMi の設定を解除する](#)」(349 ページ) の説明に従って NNMi の設定を HA から解除し、すべてのノードで NNMi をアップグレードしてから、「[HA 用の NNMi 証明書の設定](#)」(328 ページ) の説明に従って、サポート対象の HA 製品で NNMi を実行するように設定してください。または、「[アプリケーションフェイルオーバー構成の NNMi の設定](#)」(289 ページ) の説明に従って、NNMi アプリケーションフェイルオーバーに対応するように NNMi を設定できます。

NNMi のアップグレードには、Postgres データベースソフトウェアの新しいバージョンへのアップグレードが含まれます。このため、アップグレードプロセスの間、NNMi の操作を停止する必要があります。



このアップグレードプロセスの間、NNMi はおよそ 30 分から 60 分間使用できません。

HA 下の NNMi 9.0x または 9.1x を HA 下の NNMi 9.20 にアップグレードするには、アクティブノードをアップグレードして組み込みデータベースを更新してから、NNMi がまだメンテナンスモードの間にパッシブノードをアップグレードします。以下の手順に従ってください。

- 1 それぞれのパッシブノードで順番にフェイルオーバーを強制的に実行して、すべての HA ノードで NNMi 9.0x または 9.1x の設定が一貫するようにします。
- 2 NNMi 9.0x の場合、すべてのノードで NNMi 9.0x パッチ 5 以降のバージョンが実行されていることを確認します。NNMi 9.1x の場合、パッチ 3 以降を使用します。  
必要に応じて、各システムを適切な統合パッチにアップグレードします。
- 3 両方のシステムで ov.conf ファイルの値が正しいことを確認します。ov.conf ファイルは以下の場所に保存されています。

i Windows: %NnmDataDir%\shared\%nnm%\conf

i UNIX: \$NnmDataDir/shared/nnm/conf

- 4 以下のようにして、NNMi 9.0x または 9.1x HA クラスタでアクティブなノードを判別します。

- Windows:

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

- UNIX:

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

この手順の残りの部分では、現在アクティブなノードをサーバー X とし、現在パッシブなノードをサーバー Y とします。

- 5 HP-UX システムの場合は、サーバー Y で /etc/cmcluster/<resource group>/<resource group>.mon ファイルを次のように編集します。
  - a 以下の行を見つけます。
 

```
if [ ! -f /var/opt/OV/hacluster/$HA_RESOURCE_GROUP/maint_NNM -a  
! -f /var/opt/OV/hacluster/$HA_RESOURCE_GROUP/maint_NNM ]
```
  - b 2 番目の「maint\_NNM」を「maintenance」に変更します。
  - c アプリケーションをフェイルオーバーさせ、リソースグループをもう実行していないノードで手順 a から手順 b を繰り返します。
- 6 Windows システムの場合は、以下を実行します。
  - a サーバー X で、<resource group>-app リソースを停止します。
  - b %NnmDataDir%\%hacluster%\<resource group>%hamscs.vbs ファイルのアクセス制御リスト (ACL) を開き、内容を覚えておきます。
  - c hamscs.vbs ファイルを保存します。
  - d %NnmInstallDir%\%misc%\%nnm%\%ha%\%nnmhamscs.vbs スクリプトを一時ディレクトリにコピーし、ファイルを編集できるようにします。
  - e nnmhamscs.vbs ファイルを開き、product\_name の参照をすべて **NNM** に変更します。値については、元のスクリプトを参考にします。nnmhamscs.vbs ファイルを保存します。

- f 管理者として、更新した `nnmhamscs.vbs` スクリプトを  
`%NnmDataDir%\%hacluster%\<resource group%\hamscs.vbs` にコピーします。
  - g 再度 ACL を開き、以前と同じであることを確認します。
  - h `<resource group>-app` リソースを起動します。
  - i リソースがオンラインになることを確認します。 ならない場合はクラスターログを開き、構文エラーがないかどうかを確認します。クラスターログを生成するには、`cluster log /gen` コマンドを実行します。フォルダーを指定する必要がある場合は、次の構文を使用します。`cluster log /gen /copy:<my folder>`
  - j `ovstop` を実行します。
- 7 サーバー X で、以下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します。
- Windows:
 

```
%NnmDataDir%\%hacluster%\<resource_group%\maintenance
```
- ▶ maintenance ファイルの拡張子が `.txt` になっていないことを確認します。Notepad などのテキストエディタを使って編集すると、この拡張子が付く場合があります。
- UNIX:
 

```
$_NnmDataDir/hacluster/<resource_group>/maintenance
```
- ファイルは空で構いません。
- 8 サーバー X で、NNMi をアップグレードします。
- a HP マニュアル Web サイトから入手できる『HP Network Node Manager i Software アップグレードリファレンス』の説明に従って、NNMi を最新バージョンにアップグレードします。  
 この手順の実行中に、データベースがアップグレードされます。
  - b 以下のコマンドを入力して、アップグレードが正常に完了したことを確認します。  

```
ovstart
```

 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。
  - c すべてのアドオン NNM iSPI をバージョン 9.20 にアップグレードします。  
 詳細については、各 NNM iSPI のマニュアルを参照してください。
- ▶ 使用環境にスタンドアロン NNM iSPI が含まれる場合は、正常に機能させるためにそれらの製品もバージョン 9.20 にアップグレードする必要があります。それらの製品のアップグレードは、この手順の完了後に実行できます。
- 9 Windows システムの場合は、以下を実行します。
- a 更新した `nnmhamscs.vbs` スクリプト (手順 6 の手順 f を参照) をサーバー X からサーバー Y の `%NnmDataDir%\%hacluster%\<resource group%\hamscs.vbs` にコピーします。
  - b ACL を開き、以前と同じであることを確認します。
- 10 サーバー X で次のコマンドを実行します。 `nnmhadisk.ovpl NNM -replicate`
- 11 サーバー Y で、以下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します。

- Windows:

```
%NnmDataDir%¥hacluster¥<resource_group>¥maintenance
```



maintenance ファイルの拡張子が .txt になっていないことを確認します。Notepad などのテキストエディタを使って編集すると、この拡張子が付く場合があります。

- UNIX:

```
$NnmDataDir/hacluster/<resource_group>/maintenance
```

ファイルは空で構いません。

- 12 サーバー Y で、NNMi をアップグレードします。

- HP マニュアル Web サイトから入手できる『HP Network Node Manager i Software アップグレードリファレンス』の説明に従って、NNMi を最新バージョンにアップグレードします。
- エラーを生じずに、アップグレードが完了したことを確認します。
- すべてのアドオン NNM iSPI をバージョン 9.20 にアップグレードします。

詳細については、各 NNM iSPI のマニュアルを参照してください。

- 13 HA クラスタに複数のパッシブノードが含まれている場合、パッシブノードごとに手順 12 を繰り返します。

- 14 HP-UX システムの場合は、リソースグループを実行していないノードで以下のコマンドを実行します。

```
cd /etc/cmcluster/<resource_group>
cp <resource_group>.mon <resource_group>.mon.save
cp /opt/OV/misc/nnm/ha/mcsg/NNM/rg.mon <resource_group>.mon
```

- 15 サーバー X で、メンテナンスファイルを削除します。

- Windows:

```
%NnmDataDir%¥hacluster¥<resource_group>¥maintenance
```

- UNIX:

```
$NnmDataDir/hacluster/<resource_group>/maintenance
```

- 16 インストール後に以下の手順を実行します。

- 以下の変数が設定されていることを確認します。

```
NNM_INTERFACE
```

```
HA_MOUNT_POINT
```

```
NNM_ADD_ON_PRODUCTS
```

```
HA_LOCALE (C で実行する場合は不要)
```

これらの変数は以下の場所で定義します。

HP-UX Serviceguard:

```
/etc/cmcluster/<resource_group>/<resource_group>.public.env
```

Veritas:

```
/opt/VRTSvcs/bin/hagrp -display | grep UserStrGlobal
```

Windows: regedit を使用します。値は以下の場所に格納されています。

```
HKEY_LOCAL_MACHINE¥Cluster¥Groups¥<group>¥Parameters
```

- b 変数が設定されていない場合は、設定されていない値について以下のコマンドを実行します。

```
/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -config NNM -set  
NNM_INTERFACE <value for NNM_INTERFACE>
```

```
/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -config NNM -set  
HA_MOUNT_POINT <value for HA_MOUNT_POINT>
```

```
/opt/OV/misc/nnm/ha/nnmhaclusterinfo.ovpl -config NNM -set  
NNM_ADD_ON_PRODUCTS <value for NNM_ADD_ON_PRODUCTS>
```

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



HA\_LOCALE はローカライズされた言語を使用する場合にのみ必要です。

- 17 Linux HA のすべてのアップグレードで、使用しているシステムに応じて以下のコマンドを実行します。

— RHEL:

```
rm /etc/rc.d/rc*.d/S98netmgt
```

```
rm /etc/rc.d/rc*.d/K01netmgt
```

— SuSE:

```
rm /etc/init.d/rc*.d/S98netmgt
```

```
rm /etc/init.d/rc*.d/K01netmgt
```



Windows Server 2008 R2 を使用するときには、ネットワーク名リソースの名前が「Network Name」の場合があります。この名前は、仮想 IP アドレスの短縮名とする必要があります。必要に応じて、名前を以下の手順で変更します。

- 1 Failover Cluster Management を使用し、[ ネットワーク名 ] リソースを選択します。
- 2 右クリックで [ プロパティ ] を選択します。
- 3 名前を変更します。

## サポートされるすべてのオペレーティングシステムでの Oracle を使用した NNMi のアップグレード

Oracle 環境で HA 用の NNMi をアップグレードするには、「サポートされるすべてのオペレーティングシステムでの組み込みデータベースを使用した NNMi のアップグレード」(354 ページ) で説明されている手順に従います。

## HA 設定のトラブルシューティング

この項では、以下のトピックについて説明します。

- 「一般的な設定の誤り」(359 ページ)
- 「RHCS 6 での設定の問題」(360 ページ)
- 「HA リソーステスト」(360 ページ)
- 「一般的な HA のトラブルシューティング」(361 ページ)
- 「NNMi 固有の HA のトラブルシューティング」(365 ページ)
- 「NNM iSPI 固有の HA のトラブルシューティング」(369 ページ)

### 一般的な設定の誤り

HA 設定における一般的な誤りの一部を以下に示します。

- 正しくないディスク設定
  - VCS: リソースをプローブできない場合は、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場合、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可能性があります。
  - 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 のマニュアルの記載内容と照合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。
- MSFC: 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、NNMi 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確認してください。
- NFS または他のアクセスが 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 クラスターの仮想 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)
- 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。



MSFC: Failover Cluster Management 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。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に作成されました。リソースグループをシステムでオンラインにしようとして、その後すぐにフェイルオーバーする場合、メンテナンスモードを使用してリソースグループのオンラインを維持し、障害のある部分を見つけます。
- 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)。

## RHCS 6 での設定の問題

ricci サービスがダウンしていたり、意図的に無効化されている場合、HA 環境の 2 つのシステム間で /etc/cluster/cluster.conf ファイルのバージョンが異なる可能性があります。そのため、cluster.conf ファイルを定期的に監視して、ファイルのバージョンが同期化されていることを確認します。

cluster.conf ファイルのバージョンが同期化されていない場合は、次のいずれかを実行しようとする場合に問題が発生する可能性があります。

- 変更を cluster.conf に適用する
- リソースグループの設定を解除する
- クラスタを起動する
- clustat コマンドを使用する

## HA リソーステスト

このセクションでは、NNMi HA リソースグループに入れるリソースのテストを行うための一般的な方法を説明します。このテストによって、ハードウェア設定の問題が特定されます。HA の下で実行するように 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完全な設定を行うときに、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についての詳細は、HA 製品の最新マニュアルを参照してください。

HA リソースをテストするには、以下の手順を実行します。

- 1 必要に応じて、HA クラスタを起動します。
- 2 (Windows のみ) HA クラスタに、以下の仮想 IP アドレスが定義されていることを確認します。
  - HA クラスタの仮想 IP アドレス
  - 各 HA リソースグループの仮想 IP アドレス
 これらの各 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 クラスタに追加します。  
この HA リソースグループには、test など、非商用名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します。
  - a 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。  
後で NNMi HA リソースグループに関連付ける値を使用します。
  - b アクティブクラスターノードからパッシブクラスターノードにフェイルオーバーし、HA クラスターが正常にフェイルオーバーすることを確認します。
  - c 新しいアクティブクラスターノードから新しいパッシブクラスター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。
  - d 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、アクセス可能であることを確認します。また、ファイアウォールによって IP address.v がブロックされていないかも確認します。
- 5 「SAN または物理的に接続されたディスクの設定」(342 ページ) の説明に従って、共有ディスクを設定します。
- 6 共有ディスクへの接続をテストします。
  - a 「NNMi HA リソースグループへの共有ディスクの移動」(343 ページ) の説明に従って、共有ディスクをリソースとして HA リソースグループに追加します。
  - b アクティブクラスターノードからパッシブクラスターノードにフェイルオーバーし、HA クラスターが正常にフェイルオーバーすることを確認します。
  - c 新しいアクティブクラスターノードから新しいパッシブクラスター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。
  - d 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、アクティブノードにログオンして、ディスクがマウントされ、使用可能であることを確認します。
- 7 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きます。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。
- 8 各ノードからリソースグループを削除します。
  - a IP アドレスエントリを削除します。
  - b 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 を設定することができます。

## 一般的な HA のトラブルシューティング

このセクションのトピックは、NNMi および NNM iSPI の HA 設定に適用されます。以下の内容が含まれます。

- エラー：引数の数が正しくない
- 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)
- 製品の起動タイムアウト (Solaris)
- アクティブなクラスターノードのログファイルが更新されない

- NNMi HA リソースグループを特定のクラスターノードで起動できない

## エラー：引数の数が正しくない

Perl モジュール製品の名前は、大部分の NNMi HA 設定コマンドで必須パラメーターになりました。

- NNMi では、値として NNM を使用します。
- NNM iSPI で使用する値を調べるには、その NNM iSPI のマニュアルを参照してください。

## 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を実行しているコンピューターで HA クラスター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停止します。

この既知の問題の詳細については、Microsoft サポート Web サイトの記事「Windows Server 2008 R2 では、クラスター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停止します」(<http://support.microsoft.com/kb/978527>) を参照してください。



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ー (rhs.exe) で実行してください。

## 製品の起動タイムアウト (Solaris)

1つ以上の `/var/adm/messages*` ファイルに、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 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 15 分と定義されています。

Veritas タイムアウト値は調整できます。たとえば、Veritas タイムアウト値を 30 分 (1800 秒) に調整するには、以下のコマンドを順番どおりに実行できます。

```
/opt/VRTSvcs/bin/haconf -makerw
/opt/VRTSvcs/bin/hares -makerw <resource_group>-app
OnlineTimeout 1800
/opt/VRTSvcs/bin/haconf -dump -makero
```

## 製品の起動タイムアウト (Windows MSCS 2008)

NNMi 9.0x からのアップグレード後、フェイルオーバークラスターマネージャーのアプリケーションソース (<resource>-app) が「Pending」から「Failed」に変わった場合は、タイムアウトの問題である可能性があります。この場合は、以下を実行します。

- 1 **cluster log /gen** コマンドを使用して、`cluster.log` ファイルを生成します。
- 2 次のディレクトリあるログ (`C:\%Windows%\cluster\reports\cluster.log`) を開きます。
- 3 `cluster.log` ファイルで次のようなエラーが表示される場合は、**DeadlockTimeout** の問題があります。

```
ERR [RHS] Resource <resource-name>-APP handling deadlock. Cleaning
current operation.
```

- 4 **DeadlockTimeout** はエージェントがブロックされた可能性がある場合フェイルオーバーの合計時間です。**PendingTimeout** は、オンライン操作またはオフライン操作のいずれかを示します。**The DeadlockTimeout default value is 45 minutes (2,700,000 milliseconds), and the PendingTimeout default value is 30 minutes (1,800,000 milliseconds). You can change the DeadlockTimeout and the PendingTimeout values.** たとえば、75 分の **DeadlockTimeout** および 60 分の **PendingTimeout** を設定するには、次のコマンドを実行できます。

```
cluster res "<resource group>-APP" /prop DeadlockTimeout=4500000
cluster res "<resource group>-APP" /prop PendingTimeout=3600000
```

詳細については、HA ベンダーのマニュアルを参照してください。

## アクティブなクラスター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。

## NNMi HA リソースグループを特定のクラスターノードで起動できない

nmhastarttrg.ovpl コマンドまたは nnmhastarttrg.ovpl コマンドで、NNMi HA リソースグループを正常に起動、停止、または切り替えできない場合は、以下の事柄を調べてください。

- **MSFC** の場合 :
  - **Failover Cluster Management** で、NNMi HA リソースグループと基盤リソースの状態を調べてください。
  - イベントビューアー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 (たとえば、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。

```
/etc/cmcluster/<resource_group>/<resource_group>.cntl.log
```
- **VCS** の場合 :
  - **/opt/VRTSvcs/bin/hares -state** を実行してリソース状態を確認します。
  - 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。
- **RHCS**:
 

<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 (たとえば、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。

```
/etc/cmcluster/<resource_group>/<resource_group>.cntl.log
```

問題の原因を特定できない場合は、HA 製品のコマンドを使用して NNMi HA リソースグループを手動で起動できます。

- 1 共有ディスクをマウントします。
- 2 仮想ホストをネットワークインタフェースに割り当てます。
  - **MSF** の場合 :
    - **Failover Cluster Management** 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、**[Bring Online]** をクリックします。
  - **Serviceguard** の場合 : `/usr/sbin/cmmodnet` を実行して、IP アドレスを追加します。
  - **VCS** の場合 : `/opt/VRTSvcs/bin/hares -online <resource_group>-ip ¥ -sys <local_hostname>`
  - **RHCS** の場合 : `/usr/sbin/cmmodnet` を実行して、IP アドレスを追加します。
- 3 NNMi HA リソースグループを起動します。次に例を示します。
  - **Windows**:

```
%NnmInstallDir%\misc\nnm\ha\nnmhastarttrg.ovpl NNM ¥
-start <resource_group>
```

- UNIX:

```
$NnmInstallDir/misc/nnm/ha/nnmhastarttrg.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。

## 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 のみの HA 設定です。以下の内容が含まれます。

- すべてのクラスターノードを設定解除した後の HA 用 NNMi の再有効化
- NNMi を HA 下で正常に起動できない
- NNMi データへの変更がフェイルオーバーの後に表示されない
- HA の設定後、nmsdbmgr を起動できない
- HA の設定後、pmd を起動できない
- NNMi が 1 つの HA クラスターノードでのみ正常に実行される (Windows)
- ディスクフェイルオーバーが行われない
- 共有ディスクにアクセスできない (Windows)
- 共有ディスクに最新データが含まれない
- フェイルオーバー後にセカンダリノードが共有ディスクファイルを見つけられない

### すべてのクラスターノードを設定解除した後の HA 用 NNMi の再有効化

すべての NNMi HA クラスター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なおすには、プライマリノードで以下の手順を実行します。

- 1 NNMi が実行中であれば、停止します。

```
ovstop -c
```

- 2 共有ディスクへのリンクを削除します。

- Windows:

```
%NnmInstallDir%\misc\nnm\ha\nnmhadisk.ovpl NNM ¥
-setmount <HA_mount_point>
```

- UNIX:

```
$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ーを確認します。

ov.conf ファイルの場所は、「NNMi HA 設定ファイル」(370 ページ)を参照してください。

## 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、問題が仮想 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、ある種の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、NORESTART キーワードを設定しないで、システムをメンテナンスモードにします。

- 1 HA クラスターのアクティブノードで、以下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します。

- **Windows:** %NnmDataDir%\hacluster\<resource\_group>\maintenance
- **UNIX:** \$NnmDataDir/hacluster/<resource\_group>/maintenance

- 2 NNMi を起動します。

```
ovstart
```

- 3 NNMi を正常に起動できたことを確認します。

```
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します。

- **Windows:** %NnmDataDir%\hacluster\<resource\_group>\maintenance
- **UNIX:** \$NnmDataDir/hacluster/<resource\_group>/maintenance

## NNMi データへの変更がフェイルオーバーの後に表示されない

NNMi の設定で、NNMi を実行中のシステム以外のシステムを指しています。この問題を解決するには、ov.conf ファイルに以下の項目に対応した適切なエントリーがあるか確認します。

- 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/eventdb
- HA\_CUSTOMPOLLER\_DIR=<HA\_mount\_point>/NNM/dataDir/shared/nnm/databases/custompoller
- HA\_NNM\_LOG\_DIR=<HA\_mount\_point>/NNM/dataDir/log
- HA\_JBOSS\_DATA\_DIR=<HA\_mount\_point>/NNM/dataDir/nmsas/NNM/data
- HA\_LOCALE=C

ov.conf ファイルの場所は、「[NNMi HA 設定ファイル](#)」(370 ページ) を参照してください。

## 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` コマンドを実行せずに 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、共有ディスクの組み込みデータベースの場所を指していますが、この場所は NNMi からアクセスできません。

この問題を解決するには、以下の手順を実行します。

- 1 HA クラスターのアクティブノードで、以下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します。

- 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します。

- Windows:
 

```
%NnmInstallDir%\misc\%nnm%\ha\%nnmhadisk.ovpl NNM %
-to <HA_mount_point>
```
- UNIX:
 

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM %
-to <HA_mount_point>NNMi HA リソースグループを起動します。
```



データベースの破壊を避けるために、この (`-to` オプションを指定した) コマンドは 1 回しか実行できません。代替方法については、「すべてのクラスターノードを設定解除した後の HA 用 NNMi の再有効化」(365 ページ) を参照してください。

- Windows:
 

```
%NnmInstallDir%\misc\%nnm%\ha\%nnmhastarttrg.ovpl NNM %
<resource_group>
```
- UNIX:
 

```
$NnmInstallDir/misc/nnm/ha/nnmhastarttrg.ovpl NNM %
<resource_group>
```

- 3 NNMi を起動します。

```
ovstart
```

- 4 NNMi を正常に起動できたことを確認します。

```
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。

- 5 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します。

- Windows: `%NnmDataDir%\hacluster\<resource_group>\maintenance`
- UNIX: `$NnmDataDir/hacluster/<resource_group>/maintenance`

## HA の設定後、pmd を起動できない

この状況は、通常、共有ディスクを正しく設定しなかったなどの設定エラー後に発生します。pmd プロセスの障害は、`ovjboss` プロセスを完全に起動できなかった場合に発生します。



以下のログファイルを調べてください。

- **Windows:** %HA\_MOUNT\_POINT%\NNMi\dataDir\log\%nnm%\jbossServer.log
- **UNIX:** \$HA\_MOUNT\_POINT/NNM/dataDir/log/nnm/jbossServer.log

## NNMi が 1 つの HA クラスターノードでのみ正常に実行される (Windows)

**Windows** オペレーティングシステムには、2 つの異なる仮想 IP アドレス (HA クラスター用に 1 つと、HA リソースグループ用に 1 つ) が必要です。HA クラスターの仮想 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 クラスターの IP アドレスと関連付けられているノードでのみ正常に実行されます。

この問題を修正するには、HA クラスターの仮想 IP アドレスをネットワークで一意的な値に変更します。

## 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で調べて、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。

## 共有ディスクにアクセスできない (Windows)

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 設定時に完全に指定する必要があります (たとえば、S:¥)。

この問題を修正するには、HA クラスターの各ノードで nnmhaconfigure.ovpl コマンドを実行します。共有ディスクのマウントポイントのドライブを完全に指定します。

## 共有ディスクに最新データが含まれない

ディスクタイプについての nnmhaconfigure.ovpl コマンドの質問にテキスト **none** で応答すると、ov.conf ファイルでディスク関連の変数を設定するコードがバイパスされます。この状況を修正するには、「共有ディスクの手動準備」(341 ページ) の手順に従います。

## フェイルオーバー後にセカンダリ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。この場合には、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、以下の手順を実行します。

- 1 HA クラスターのアクティブノードで、以下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します。



- Windows: %NnmDataDir%\hacluster\<resource\_group>\maintenance
  - UNIX: \$NnmDataDir/hacluster/<resource\_group>/maintenance
- 2 アクティブノードにログオンして、ディスクがマウントされ、使用可能であることを確認します。
  - 3 NNMi を停止します。

**ovstop**

- 4 NNMi データベースを共有ディスクにコピーします。

- Windows:

```
%NnmInstallDir%\misc\%nnm%ha%\nnmhadisk.ovpl NNM ¥
-to <HA_mount_point>
```

- UNIX:

```
$NnmInstallDir/misc/nnm/ha/nnmhadisk.ovpl NNM ¥
-to <HA_mount_point>
```



データベースの破壊を避けるために、この (-to オプションを指定した) コマンドは 1 回しか実行できません。代替方法については、「すべてのクラスターノードを設定解除した後の HA 用 NNMi の再有効化」(365 ページ) を参照してください。

- 5 NNMi HA リソースグループを起動します。

- Windows:

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

- UNIX:

```
$NnmInstallDir/misc/nnm/ha/nnmhastartrg.ovpl NNM ¥
<resource_group>
```

- 6 NNMi を起動します。

**ovstart**

- 7 NNMi を正常に起動できたことを確認します。

**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します。

- Windows: %NnmDataDir%\hacluster\<resource\_group>\maintenance

- UNIX: \$NnmDataDir/hacluster/<resource\_group>/maintenance

## NNM iSPI 固有の HA のトラブルシューティング

HA 下で実行中の NNM iSPI のトラブルシューティングについては、その NNM iSPI のマニュアルを参照してください。

## HA 設定リファレンス

### NNMi HA 設定ファイル

表 31 に、NNMi HA 設定ファイルを示します。これらのファイルは、NNMi 管理サーバー上の NNMi とアドオン NNM iSPI に適用されます。これらのファイルは、以下の場所にインストールされます。

- **Windows:** %NnmDataDir%\shared\nnm\conf
- **UNIX:** \$NnmDataDir/shared/nnm/conf

表 31 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、nmhaclusterinfo.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。
nmdatareplicator.conf	このファイルは、nmdatareplicator.ovpl コマンドで、アクティブノードからパッシブノードへのデータレプリケーションを含む NNMi のフォルダー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。

### NNMi に付属している HA 設定スクリプト

表 32 と表 33 に、NNMi に付属している HA 設定スクリプトを示します。表 32 に示した NNMi 付属のスクリプトは、カスタマー 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うことが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、以下の場所にインストールされます。

- **Windows:** %NnmInstallDir%\misc\nnm\ha
- **UNIX:** \$NnmInstallDir/misc/nnm/ha

表 32 NNMi HA 設定スクリプト

スクリプト名	説明
nnmhaconfigure.ovpl	NNMi または NNM iSPI 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 または NNM iSPI 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 および NNM iSPI 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、このスクリプトはプライマリノードで実行します。

表 33 に示した NNMi 付属のスクリプトは、371 ページの表 32 に示したスクリプトで使  
用します。表 33 に示したスクリプトは直接実行しないでください。

表 33 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 設定では、VCS の起動、停止、および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	MSFC HA クラスタで、NNMi プロセスを起動、停止、および監視するスクリプ 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 MSFC によって使 用され、次の場所に保存されます： %NnmDataDir%\%hacluster%\<resource_group%\hamscs.vbs

## NNMi HA 設定のログファイル

以下のログファイルは、NNMi 管理サーバー上の NNMi とアドオン NNM iSPI 用の 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 :**
  - イベントビューアーのログ
  - %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
  - %HA\_MOUNT\_POINT%\NNM\dataDir\log\nnm\jbossServer.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、他の設定上の問題点、起動 / 停止上の問題点を含む、クラスター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
  - \$HA\_MOUNT\_POINT/NNM/dataDir/log/nnm/jbossServer.log

- VCS用のLinuxまたはSolarisの場合：

表 34 VCS用のLinuxまたはSolarisの場合

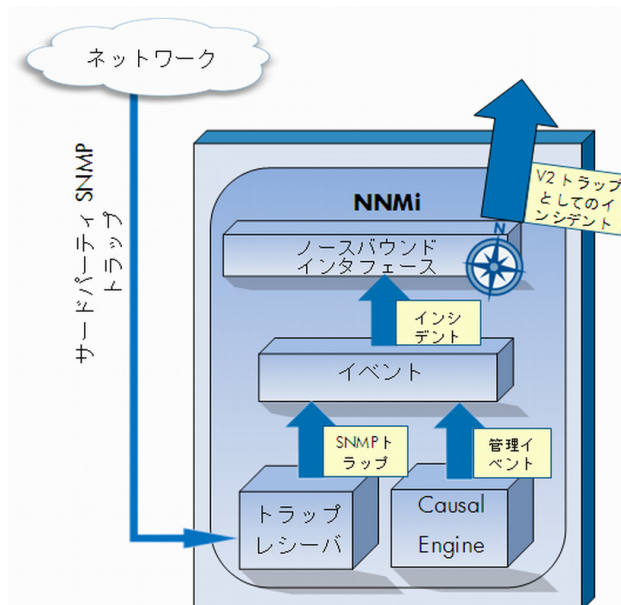
リソース	ログファイル
<resource_group>-app	<ul style="list-style-type: none"> <li>• /var/VRTSvcs/log/Application_A.log</li> <li>• \$SHA_MOUNT_POINT/NNM/dataDir/log/nnm/ovspmd.log</li> <li>• \$SHA_MOUNT_POINT/NNM/dataDir/log/nnm/public/postgres.log</li> <li>• \$SHA_MOUNT_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log</li> <li>• \$SHA_MOUNT_POINT/NNM/dataDir/log/nnm/jbossServer.log</li> <li>• /var/adm/messages*</li> </ul>
<resource_group>-dg <resource_group>-volume <resource_group>-mount	<ul style="list-style-type: none"> <li>• /var/VRTSvcs/log/DiskGroup_A.log</li> <li>• /var/VRTSvcs/log/Volume_A.log</li> <li>• /var/VRTSvcs/log/Mount_A.log</li> <li>• /var/adm/messages*</li> </ul>
<resource_group>-ip	<ul style="list-style-type: none"> <li>• /var/VRTSvcs/log/IP_A.log</li> <li>• /var/adm/messages*</li> </ul>

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\* 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。

- RCHS用のLinux実行時：
  - /var/adm/syslog/syslog.log
  - \$SHA\_MOUNT\_POINT/NNM/dataDir/log/nnm/ovspmd.log
  - \$SHA\_MOUNT\_POINT/NNM/dataDir/log/nnm/public/postgres.log
  - \$SHA\_MOUNT\_POINT/NNM/dataDir/log/nnm/public/nmsdbmgr.log
  - \$SHA\_MOUNT\_POINT/NNM/dataDir/log/nnm/jbossServer.log



# NNMi Northbound イン タフェース



HP Network Node Manager i Software (NNMi) には、NNMi Northbound インタフェースが用意されており、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することができます。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。

NNMi には、NNMi Northbound インタフェースを使用して以下の製品との統合をサポートする機能も組み込まれています。

- HP Business Service Management (BSM) プラットフォームの Operations Management 機能。
- HP Operations Manager (HPOM) アクティブメッセージブラウザー。
- IBM Tivoli Netcool/OMNIBus。
- HP ArcSight Logger。

異なる Northbound アプリケーションと統合するには、この章の指示に従ってください。

この章には、以下のトピックがあります。

- [NNMi Northbound インタフェース](#)
- [NNMi ノースバウンドインタフェースの有効化](#)
- [NNMi ノースバウンドインタフェースの使用法](#)
- [NNMi ノースバウンドインタフェースの変更](#)
- [NNMi ノースバウンドインタフェースの無効化](#)
- [NNMi ノースバウンドインタフェースのトラブルシューティング](#)
- [アプリケーションフェールオーバーと NNMi ノースバウンドインタフェース](#)
- [\[NNMi Northbound Interface デスティネーション\] フォームのリファレンス](#)

## NNMi Northbound インタフェース

NNMi Northbound インタフェースは、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。

NNMi Northbound インタフェースは、インシデントライフサイクルの状態変更通知、インシデント関連処理通知、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することができます。

NNMi Northbound インタフェースは、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。NNMi Northbound インタフェースは、NNM 6.x または 7.x 管理ステーションによって生成されたイベントは Northbound アプリケーションに転送しません。

### 値

NNMi ノースバウンドインタフェースにより、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することができます。NNMi Northbound インタフェースは、その他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。

### サポートされるバージョン

この章の情報は、NNMi バージョン 9.00 以降に適用されます。

サポートされているハードウェアプラットフォームおよびオペレーティングシステムの最新情報については、『NNMi システムおよびデバイス対応マトリックス』を参照してください。

### 用語

この章では、以下の用語を使用します。

- Northbound アプリケーション 祐 NMPv2c 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。
- トラップ受信コンポーネント 祐 NMP トラップを受信する、ノースバウンドアプリケーションの一部分。
  - 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 ó 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。



- **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。

## ドキュメント

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、そのアプリケーションのマニュアルを参照してください。

## NNMi ノースバウンド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。

NNMi ノースバウンドインタフェースを有効にするには、以下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します。
- 2 NNMi 管理サーバーで、NNMi インシデント転送を設定します。
  - a NNMi コンソールで、**[HP NNMi-Northbound Interface 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。  
(使用可能な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]** ボタンを使用可能にしてください。)
  - 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可能にします。
  - 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、「**Northbound アプリケーションの接続パラメーター**」(385 ページ) 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、「**NNMi Northbound インタフェース統合の内容**」(386 ページ) 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 オプション。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。

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。

## NNMi ノースバウンド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-registration.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、統合で送信可能なトラップのタイプを説明します。コンテンツ設定の設定詳細については、「[NNMi Northbound インタフェース統合の内容](#)」(386 ページ)を参照してください。

### インシデント転送

#### 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。

#### 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 値をパディングします。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、以下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 IncidentNodeHostname (1.3.6.1.4.1.11.2.17.19.2.2.21) と、(n + 24) 番目の varbind の値 IncidentNodeMgmtAddr (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いずれかが 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、『NNMi デプロイメントリファレンス』の「トラップおよびインシデント転送」を参照してください。

## 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、[HP NNMi-Northbound Interface デスティネーション] ページの [送信オプション] で行った選択によって異なります。

### 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 EventLifecycleStateClosed (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。EventLifecycleStateClosed 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。EventLifecycleStateClosed トラップは、6 番目の varbind である IncidentUuid (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。

### 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 LifecycleStateChangeEvent (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、LifecycleStateChangeEvent と元のインシデントを関連付けできます。

LifecycleStateChangeEvent トラップは、以下の 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。

- IncidentUuid、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。

- IncidentLifecycleStatePreviousValue、7 番目の varbind (1.3.6.1.4.1.11.2.17.19.2.2.200)
- IncidentLifecycleStateCurrentValue、8 番目の varbind (1.3.6.1.4.1.11.2.17.19.2.2.201)

以下の表は、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

## インシデント関連処理通知

Northbound 転送先にインシデント関連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関連処理されると、NNMi はインシデント関連処理トラップを 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関連変更を複製することができます。

### 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、以下の相関トラップを送信します。

- EventDedupCorrelation (1.3.6.1.4.1.11.2.17.19.2.0.1100)
- EventImpactCorrelation (1.3.6.1.4.1.11.2.17.19.2.0.1101)
- EventPairwiseCorrelation (1.3.6.1.4.1.11.2.17.19.2.0.1102)
- EventRateCorrelation (1.3.6.1.4.1.11.2.17.19.2.0.1103)
- EventApaCorrelation (1.3.6.1.4.1.11.2.17.19.2.0.1104)
- EventCustomCorrelation (1.3.6.1.4.1.11.2.17.19.2.0.1105)

各トラップは、以下の varbind において、1 つの親子インシデント相関関係を示します。

- IncidentCorrelationIndicatorParentUuid、6 番目の varbind (1.3.6.1.4.1.11.2.17.19.2.2.6)
- IncidentCorrelationIndicatorChildUuid、7 番目の varbind (1.3.6.1.4.1.11.2.17.19.2.2.300)

### 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、以下の相関トラップを送信します。

- EventDedupCorrelationGroup (1.3.6.1.4.1.11.2.17.19.2.0.2100)
- EventImpactCorrelationGroup (1.3.6.1.4.1.11.2.17.19.2.0.2101)
- EventPairwiseCorrelationGroup (1.3.6.1.4.1.11.2.17.19.2.0.2102)
- EventRateCorrelationGroup (1.3.6.1.4.1.11.2.17.19.2.0.2103)
- EventApaCorrelationGroup (1.3.6.1.4.1.11.2.17.19.2.0.2104)
- EventCustomCorrelationGroup (1.3.6.1.4.1.11.2.17.19.2.0.2105)

各トラップは、以下の varbind において、親子インシデント相関関係を示します。

- IncidentCorrelationIndicatorParentUuid、6 番目の varbind (1.3.6.1.4.1.11.2.17.19.2.2.6)
- IncidentCorrelationIndicatorChildCount、7 番目の varbind (1.3.6.1.4.1.11.2.17.19.2.2.301)
- IncidentCorrelationIndicatorChildUuidCsv、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 のカンマ区切りリストです。

## 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 **EventDeleted (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。EventDeleted トラップは、6 番目の varbind である **IncidentUuid (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。

## 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、以下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
- EventLifecycleStateClosed トラップ
- LifecycleStateChangeEvent トラップ
- EventDeleted トラップ
- 相関関係通知トラップ

以下の注は、相関関係通知トラップに適用されます。

- 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。(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、EventDedupCorrelation または EventDedupCorrelationGroup 相関関係通知トラップとは無関係に転送されません。同様に、RateCorrelation 管理イベントは EventRateCorrelation または EventRateCorrelationGroup 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いずれかを転送しない場合でも、NNMi により関連管理イベントが転送される場合があります。

---

## NNMi ノースバウンドインタフェースの変更

NNMi ノースバウンドインタフェースの設定パラメーターを変更するには、以下の手順を実行します。

- 1 NNMi コンソールで、**[HP NNMi-Northbound Interface 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。
- 2 転送先を選択し、**[編集]** をクリックします。
- 3 該当するように値を変更します。

このフォームのフィールドの詳細は、「[NNMi Northbound Interface デスティネーション] フォームのリファレンス」(384 ページ) 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] をクリックします。

変更はただちに有効になります。

---

## NNMi ノースバウンド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、以下の手順を実行します。

- 1 NNMi コンソールで、[HP NNMi-Northbound Interface 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。
- 2 転送先を選択し、[編集] をクリックします。  
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します。
- 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] をクリックします。  
変更はただちに有効になります。


---

## NNMi ノースバウンドインタフェースのトラブルシューティング

NNMi ノースバウンドインタフェースが正常に機能しない場合は、以下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します。  
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します。
  - a NNMi コンソールで、[HP NNMi-Northbound Interface 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、この機能を確認します。
  - 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 EventLifecycleStateClosed トラップ (または LifecycleStateChangeEvent 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、HP サポートにご連絡ください。

4 Northbound 転送先に SNMP トラップが含まれる場合は、この機能を確認します。

a NNMi 管理サーバーで以下のコマンドを入力することにより、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 -u username -p password -a ¥
discovered_node NNMi_node linkDown
```

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、NNMi\_node は NNMi 管理サーバーのホスト名または IP アドレスです。

b 30秒後に、Northbound アプリケーションが転送されたトラップを受信したかどうかを確認します。

— Northbound アプリケーションがトラップを受信した場合、NNMi Northbound インタフェースは正常に機能しています。

— Northbound アプリケーションがトラップを受信しなかった場合は、異なる Northbound アプリケーションに接続する新規 Northbound 転送先を設定してから、手順 a からこのテストを繰り返します。

再テストに合格した場合、問題は最初の Northbound アプリケーションにあります。アプリケーションのドキュメントでトラブルシューティング情報を参照してください。

再テストに不合格になった場合は、HP サポートにご連絡ください。

---

## アプリケーションフェールオーバーと NNMi ノースバウンドインタフェース

NNMi 管理サーバーが NNMi アプリケーションフェー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 (たとえば、NNMi コンソールの起動) は失敗します。

## 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、以下の考慮事項が 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 管理サーバーでのみ、NNMi ノースバウンドインタフェースを設定します。

[HP NNMi-Northbound Interface 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 ノースバウンドインタフェースは、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。

## 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 管理サーバーにのみ設定します。[HP NNMi-Northbound Interface 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。

---

## [NNMi Northbound Interface デスティネーション] フォームのリファレンス

[HP NNMi-Northbound Interface 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。([HP NNMi-Northbound Interface 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 ユーザーのみが [HP NNMi-Northbound Interface デスティネーション] フォームにアクセスできます。

[HP NNMi-Northbound Northbound Interface デスティネーション] フォームには、以下の領域の情報が表示されます。



- 「Northbound アプリケーションの接続パラメーター」(385 ページ)
- 「NNMi Northbound インタフェース統合の内容」(386 ページ)
- 「NNMi Northbound インタフェース転送先のステータス情報」(388 ページ)

統合設定に変更を適用するには、[HP NNMi-Northbound Interface 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。

## Northbound アプリケーションの接続パラメーター

表 35 は、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 35 Northbound アプリケーションの接続情報

フィールド	説明
ホスト	<p>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。</p> <p>統合では、以下のサーバーの識別方法がサポートされています。</p> <ul style="list-style-type: none"> <li>• <b>NNMi FQDN</b> 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。</li> <li>• <b>ループバックを使用</b> 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。</li> <li>• <b>その他</b> 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。 これがデフォルト設定です。</li> </ul> <p>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、「アプリケーションフェイルオーバーと NNMi ノースバウンドインタフェース」(383 ページ) を参照してください。</p>
ポート	<p>Northbound アプリケーションが SNMP トラップを受信する UDP ポート。</p> <p>Northbound アプリケーション固有のポート番号を入力します。</p> <p>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。</p>
コミュニティ文字列	<p>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。</p> <p>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。</p> <p>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。</p>

## NNMi Northbound インタフェース統合の内容

表 36 に、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示します。

表 36 NNMi ノースバウンド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	<p>インシデント転送の指定。</p> <ul style="list-style-type: none"> <li> <b>管理</b>            NNMi は、NNMi が生成した管理イベントのみを Northbound アプリケーションに転送します。         </li> <li> <b>サードパーティ SNMP トラップ</b>            NNMi は、NNMi が管理対象デバイスから受信する SNMP トラップのみを Northbound アプリケーションに転送します。         </li> <li> <b>Syslog</b>            NNMi は、NNMi が管理対象デバイスから受信する ArcSight Syslog メッセージのみを 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。         </li> </ul> <p>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、「<a href="#">インシデント転送</a>」(378 ページ)を参照してください。</p>
ライフサイクル状態の変化	<p>インシデント変更通知の仕様。</p> <ul style="list-style-type: none"> <li> <b>エンハンスド解決済み</b>            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。         </li> <li> <b>変化した状態</b>            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。         </li> <li> <b>両方</b>            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。  <b>注:</b> 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。         </li> </ul> <p>詳細については、「<a href="#">インシデントライフサイクル状態変化通知</a>」(379 ページ)を参照してください。</p>

表 36 NNMi ノースバウンドインタフェースの内容設定情報 (続き)

フィールド	説明
<p>関連処理</p>	<p>インシデント関連処理通知の仕様。</p> <ul style="list-style-type: none"> <li>● <b>なし</b> NNMi は、NNMi 因果関係分析によるインシデント関連処理結果を 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。</li> <li>● <b>単一</b> NNMi は、NNMi 因果関係分析で判明した親子インシデント関連関係ごとにトラップを 1 つ送信します。</li> <li>● <b>グループ</b> 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関連処理ごとに、トラップを 1 つ送信します。</li> </ul> <p>詳細については、「<a href="#">インシデント関連処理通知</a>」(380 ページ)を参照してください。</p>
<p>削除</p>	<p>インシデント削除の仕様。このセクションは、[ <b>インシデント</b> ] 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。</p> <ul style="list-style-type: none"> <li>● <b>送信しない</b> 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。</li> <li>● <b>送信</b> 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。</li> </ul> <p>詳細については、「<a href="#">インシデント削除通知</a>」(381 ページ)を参照してください。</p>
<p>NNMi コンソールアクセス</p>	<p>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。</p> <p>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。</p> <p>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。たとえば、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、[ <b>HTTP</b> ] オプションまたは [ <b>HTTPS</b> ] オプションを選択します。</p>

表 36 NNMi ノースバウンドインタフェースの内容設定情報 (続き)

フィールド	説明
Incident Filter( インシデントフィルター)	<p>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 (たとえば、.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 (たとえば、.1.3.6.1.6.3.1.1.5.*) にすることができます。</p> <p>以下のオプションの 1 つを選択します。</p> <ul style="list-style-type: none"> <li>• <b>なし</b> 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。</li> <li>• <b>含む</b> 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のみを送信します。</li> <li>• <b>除外する</b> 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。</li> </ul> <p>インシデントフィルターを指定します。</p> <ul style="list-style-type: none"> <li>• 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。</li> <li>• 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。</li> </ul> <p>詳細については、「イベント転送フィルター」(381 ページ) を参照してください。</p>

## NNMi Northbound インタフェース転送先のステータス情報

表 37 に、ノースバウンド転送先の読み取り専用ステータス情報を示します。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 37 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	<p>転送先ホスト名の解決先となる IP アドレス。</p> <p>この値は、このノースバウンド転送先に固有です。</p>
アップタイム (秒)	<p>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。</p> <p>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。</p>
NNMi URL	<p>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。</p> <p>この値は、このノースバウンド転送先に固有です。</p>

## 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、以下の手順を実行します。

- 1 コマンドプロンプトで、**nnmloadmib.ovpl -load hp-nnmi.mib** コマンドを実行して hp-nnmi.mib ファイルをロードします。
- 2 コマンドプロンプトで、**nnmloadmib.ovpl -load p-nnmi-registration.mib** コマンドを実行して hp-nnmi-registration.mib ファイルをロードします。
- 3 コマンドプロンプトで、**nnmloadmib.ovpl -load hp-nnmi-nbi.mib** コマンドを実行して hp-nnmi-nbi.mib ファイルをロードします。
- 4 オプション手順: コマンドプロンプトで、**nnmloadmib.ovpl -load hp-nnmi-ispiref-nbi.mib** コマンドを実行して hp-nnmi-ispiref-nbi.mib ファイルをロードします。
- 5 NNMi コンソールから、[ **設定** ] ワークスペースを開きます。
- 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。



# NNMi のメンテナンス

この項では以下の章について説明します。

- NNMi のバックアップおよびリストアツール
- NNMi の保守
- NNMi ログイン
- Xen 仮想化環境での NNMi の実行





# NNMiのバックアップおよびリストアーツール

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して行うには、バックアップおよびリストアーに関して優れた方針を持つことが重要です。HP Network Node Manager i Software (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、以下の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（組み込みまたは外部）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ーするために NNMi で装備しているツールについて説明しています。

この章には、以下のトピックがあります。

- [バックアップコマンドとリストアーコマンド](#)
- [NNMi データのバックアップ](#)
- [NNMi データのリストアー](#)
- [バックアップとリストアーの方針](#)
- [組み込みデータベースのみをバックアップおよびリストアーする](#)

## バックアップコマンドとリストアーコマンド

NNMiには、NNMiデータをバックアップおよびリストアーするために以下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** 組み込み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ーします。
- `nnmbakupembdb.ovpl`— **NNMi** 組み込みデータベース（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`—`nnmbakupembdb.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ーします。
- `nnmresetembdb.ovpl`—**NNMi** 組み込み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または **UNIX** のマンページを参照してください。

## NNMi データのバックアップ

**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。**NNMi** バックアップコマンドにより、バックアップデータの **tar** アーカイブを作成したり、独自のツールを使用してバックアップファイルを圧縮したりできます。これで、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。



**NNMi** 実装で **Oracle** をメイン **NNMi** データベースとして使用する場合は、**NNMi** ファイルシステムデータでのみ **NNMi** バックアップコマンドとリストアーコマンドを使用できます。外部データベースの保守は、既存のデータベースバックアップおよびリストアー手順の一環として扱う必要があります。

バックアップデータとリストアーデータには、ご使用のネットワーク環境にインストールされている **NNM iSPI** すべてのデータが含まれていることも、含まれていないこともあります。詳細については、各 **NNM iSPI** に付属のドキュメントで確認してください。



ファイルをロックするソフトウェア（たとえば、ウイルス対策ソフトウェアやシステムバックアップソフトウェア）は、すべて **NNMi** データベースへの **NNMi** のアクセスを妨害する可能性があります。これにより、ウイルス対策アプリケーションなど、他のプロセスで使用されているファイルに対する読み取りまたは書き込みができなくなるような問題が生じる可能性があります。**NNMi Postgres** データベースの場合は、**NNMi** データベースディレクトリ (**Windows** の `%NNM_DB%`、**UNIX** の `$NNM_DB`) を除外するようにアプリケーションを設定してください。**NNMi** データベースを定期的にバックアップするには、`nnmbackup.ovpl` を使用します。

## 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、オペレーターは制約を受けることなく NNMi コンソールを使用することができ、他のプロセスは NNMi データベースとやりとりできます。オンラインバックアップを実行することにより、**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のみをバックアップできます。組み込み 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。外部データベースの場合、このバックアップには NNMi ファイルシステムデータが含まれます。外部データベースをバックアップするために、NNMi プロセスが実行されている必要はありません。
- 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のみ適用されます。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。組み込み 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。外部データベースの場合、このバックアップには NNMi ファイルシステムデータのみが含まれます。

## 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いくつか指定できます。

### 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。

設定領域には以下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組み込みデータベーステーブルのみ。
- オフラインバックアップの場合は、組み込みデータベース全体。
- 全バックアップの場合は、**表 38** 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。

トポロジ領域には以下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組み込みデータベーステーブルのみ。
- オフラインバックアップの場合は、組み込みデータベース全体。
- 全バックアップの場合は、**表 38** 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付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 event) 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。

イベント領域には以下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組み込みデータベーステーブルのみ。
- オフラインバックアップの場合は、組み込みデータベース全体。
- 全バックアップの場合は、表 38 のリストに示すファイルシステム内の NNMi 設定情報と、表 39 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組み込みデータベース全体が含まれます。

表 38 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 のみ)	設定情報
%NnmInstallDir%¥misc¥nms¥lic \$NnmInstallDir/misc/nms/lic	その他のライセンス情報
%NnmInstallDir%¥nmsas¥server¥nms¥conf \$NnmInstallDir/nmsas/server/nms/conf	jboss の設定
%NnmDataDir%¥conf \$NnmDataDir/conf	他の HP 製品が共有する設定
%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
<ul style="list-style-type: none"> <li>• Windows Server 2008: &lt;drive&gt;:¥Program Files (x86)¥HP¥HP BTO Software¥data¥shared¥nnm¥conf¥licensing¥LicFile.txt</li> <li>• UNIX: /var/opt/OV/shared/nnm/conf/licensing/LicFile.txt</li> </ul>	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-snmplib \$NnmDataDir/shared/nnm/user-snmplib	共有されるユーザー追加の SNMP MIB 情報
%NnmDataDir%¥shared¥nnm¥actions \$NnmDataDir/shared/nnm/actions	共有される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

表 38 設定領域ファイルとディレクトリ (続き)

ディレクトリまたはファイル名	説明
%NnmDataDir%\shared\nnm\lrf \$NnmDataDir/shared/nnm/lrf	共有される NNMi コンポーネント登録ファイル
%NnmDataDir%\shared\nnm\conf\props \$NnmDataDir/shared/nnm/conf/props	共有される 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	共有される NNMi ノードグループマップの背景イメージ

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 39 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.0.0.log	NNMi コンソールサインインログ

## NNMi データのリストアー

NNMi リストアースクリプト (`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り、NNMi でリストア可能なバックアップデータが決まります。

▶ `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。

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。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。上書きするのは、バックアップのリストア以後に削除されたオブジェクトと、バックアップの削除以後に作成されたオブジェクトです。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。組み込み 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。外部データベースの場合、リストアには NNMi ファイルシステムデータのみが含まれ、実行中の NNMi プロセスが存在しないようにする必要があります。
- 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。外部データベースの場合、このバックアップには NNMi ファイルシステムデータのみが含まれます。

`-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 組み込み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 リストアコマンドにより、現在の作業ディレクトリの一時フォルダーに **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記憶領域があるため一時フォルダーを使用でき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。

▶ 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恐れがあるため、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。

▶ 以下の点に注意してください。

- NNMi ではバックアップの復元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る。
- この再同期中に以下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、アップグレード、アプリケーションフェイルオーバー、バックアップの復元または手動による再同期の後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。

- この再同期中に NNMi を停止しないでください。再同期を確実に行うには、バックアップの復元後に数時間 NNMi が実行されている必要があります。

## 同じシステムでのリストアー

1つのシステムでバックアップコマンドとリストアーコマンドを使用することにより、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ーの実行時までの間に、以下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
- キャラクターセット (言語)
- ホスト名
- ドメイン

## 異なるシステムでのリストアー

バックアップコマンドとリストアーコマンドを使用して、NNMi 管理サーバーから他の管理サーバーへデータを転送することができます。異なるシステムでのリストアーの用途には、システム障害からの復旧や、オペレーティングシステムのアップグレード時の NNMi の異なるシステムへの転送などがあります。

### ベストプラクティス

NNMi UUID がデータベースのリストアー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しているようです。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。



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には、`nnmconfigexport.ovpl` および `nnmconfigimport.ovpl` コマンドを使用します。

異なるシステムのリストアーでは、両方のシステムで以下の項目を同じにする必要があります。

- NNMi のバージョン (パッチを含む)
- OS のタイプとバージョン
- キャラクターセット (言語)

以下の項目は、2つのシステム間で異なってもかまいません。

- ホスト名
- ドメイン

異なるシステムでのリストアーの場合、`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、「[NNMi のライセンス](#)」(123 ページ)を参照してください。

## バックアップとリストアーの方針

### すべてのデータを定期的にバックアップする

ディザスターリカバリ計画には、すべての NNMi データの完全バックアップを定期的に行う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。次に例を示します。

```
nmbackup.ovpl -force -type online -scope all -archive
               -target nmi_backups¥periodic
```

ハードウェアに障害が発生したために NNMi データを復旧する必要がある場合は、以下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します。
- 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 をインストールします。
- 3 NNMi データをリストアーします。
  - リカバリ NNMi 管理サーバーが「[同じシステムでのリストアー](#)」(399 ページ)の一覧にある要件を満たす場合は、以下の例のようなコマンドを実行します。

```
nmrestore.ovpl -force -lic
               -source nmi_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ーを行うのに適格ではなくても、「[異なるシステムでのリストアー](#)」(399 ページ)の一覧にある要件を満たす場合は、以下の例に似たコマンドを実行します。

```
nmrestore.ovpl -force
               -source nmi_backups¥periodic¥newest_backup
```

必要に応じてライセンスを更新します。

### 設定変更前のデータのバックアップ

設定変更を開始する前に、領域を限定したバックアップ(「[バックアップ領域](#)」(395 ページ)で説明)を必要に応じて実施してください。このようにすると、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。次に例を示します。

```
nmbackup.ovpl -type online -scope config
               -target nmi_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ーするには、すべての NNMi プロセスを停止してから、以下の例のようなコマンドを実行します。

```
nmrestore.ovpl -force -source nmi_backups¥config¥newest_backup
```



## NNMi またはオペレーティングシステムのアップグレード前のバックアップ

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 データベースに対する変更が何も行われなくするために、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。次に例を示します。

```
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、「異なるシステムでのリストア」(399 ページ) 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、以下の例に似たコマンドを実行します。

```
nnmrestore.ovpl -lic -source nnmi_backups¥offline¥newest_backup
```

## ファイルシステムのファイルのみのリストア

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、以下の例に似たコマンドを実行します。

```
nnmrestore.ovpl -partial
               -source nnmi_backups¥offline¥newest_backup
```

このコマンドは、NNMi 実装のメイン NNMi データベースとして Oracle を使用する場合に役立ちます。

---

## 組み込みデータベースのみをバックアップおよびリストアする

NNMi では、nnmbakupembdb.ovpl コマンドと nnmrestoreembdb.ovpl コマンドにより、NNMi 組み込みデータベースのみをバックアップおよびリストアします。この機能は、NNMi の設定においてデータのスナップショットを作成する場合に便利です。nnmbakupembdb.ovpl コマンドと nnmrestoreembdb.ovpl コマンドは、オンラインバックアップのみを実行します。最低でも、nmsdbmgr サービスが実行されている必要があります。

### ベストプラクティス

nnmresetembdb.ovpl コマンドは、組み込み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。このコマンドにより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可能性がなくなります。組み込みデータベースリセットコマンドの実行については、nnmresetembdb.ovpl リファレンスページか UNIX のマンページを参照してください。

## HA 環境におけるバックアップおよび復元ツールの使用

HA環境でバックアップおよび復元ツールを使用する場合に役立つヒントをいくつか紹介します。

### バックアップ

- アクティブ (プライマリ) システムを使用してバックアップを実行する (設定ファイルが古かったり、共有ディスク情報が含まれていなかったりするため (バックアップノードは共有ディスクにアクセスできないため)、バックアップ (セカンダリ) ノードのバックアップはお勧めできません)。
- 共有ディスクはアクティブノードに接続する。cron ジョブを使用している場合、共有ディスクがマウントされていることを確認します。
- システムをメンテナンスモードにする (フェイルオーバーをトリガーしないように)。
- アクティブノードでのみ `nnmbackup.ovpl` スクリプトを使用してオンラインバックアップを実行する。
- 定期的にオフラインバックアップを実行する。
- 詳細については、`nnmbackup.ovpl` リファレンスページ、または UNIX マンページを参照してください。

### 復元

- 共有ディスクがマウントされていることを確認する。
- システムがメンテナンスモードになっていることを確認する。
- `nnmrestore.ovpl` スクリプトを使用して復元を実行する。
- 詳細については、`nnmrestore.ovpl` のリファレンスページまたは UNIX のマンページを参照してください。

HA 環境で NNMi を使用方法の詳細については、「高可用性クラスターに NNMi を設定する」(319 ページ)を参照してください。

# 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することができます。

本章には、以下のトピックがあります。

- NNMi フォルダーのアクセス制御リストの管理
- カスタムポーラー収集エクスポートの管理
- インシデントアクションの管理
- `hosted-on-trapstorm.conf` ファイルによるトラップストームのブロック
- `trapFilter.conf` ファイルによるインシデントのブロック
- NNMi の文字セットエンコードの設定
- 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
- 最も古い SNMP トラップインシデントの自動トリム機能の設定
- SNMP MIB 変数名を表示するための NNMi ゲージタイトルの変更
- NNMi 正規化プロパティの変更
- 同時 SNMP 要求の変更
- 組み込みデータベースポートの変更
- MIB ブラウザーパラメーターの変更
- NNMi 自己監視
- 特定ノードの検出プロトコルの使用を抑える
- 大規模スイッチの VLAN インデックス付けの使用を抑制する
- ノードコンポーネントステータスの設定

## NNMi フォルダーのアクセス制御リストの管理

「アクションサーバー名のパラメーターの設定」(408 ページ)に示されているように、HP NNM Action Server を実行するユーザー名の変更が必要な状況が発生する場合があります。ユーザー名の権限を変更せずに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、HP 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可能性があります。本項では、この発生を防ぐ方法について説明します。

NNMi (Everest) には、以下のディレクトリ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 Everest の /var/opt/OV/log/nnm/public フォルダー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 Everest 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 フォルダーの権限が変更され、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 フォルダーに ACL を設定できない場合、スクリプトは /var/opt/OV/log/nnm/public フォルダーをワールド (その他のユーザー) により書き込み可能にし、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 フォルダー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 フォルダー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 4 つのすべてのプラットフォーム (RedHat、SuSE、HP-UX、および Solaris) でサポートされています。

たとえば、以下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** フォルダー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** フォルダーの権限は 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、SuSE Linux、および Solaris:

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

HP-UX:

```
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 コマンドの使用の詳細については、該当するリファレンスページ、または UNIX のマンページを参照してください。

## 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。『HP Network Node Manager Software ステップバイステップガイド ( カスタムポーラーに関するホワイトペーパー )』(HP Network Node Manager i Software Step-by-Step Guide to Custom Poller White Paper) も参照してください。

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処理して別の場所に保存していないと、ファイルは失われます。

### カスタムポーラー収集のエクスポートディレクトリの変更

NNMi は、ユーザーがエクスポートした収集データを以下のディレクトリに書き込みます。

- Windows: %NNM\_DATA%\\$shared\\$nmm\$databases\$custompoller\$export
- UNIX: \$NNM\_DATA/shared/nmm/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、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\nms-custompoller.properties
  - UNIX: \$NNM\_PROPS/nms-custompoller.properties
- 2 exportdir エントリーを特定します。このエントリーは以下の行のように記述されています。

```
#!com.hp.nnm.custompoller.exportdir=< カスタムポーラーメトリックスを  
エクスポートするためのベースディレクトリ >
```

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、以下のように行を変更します。

```
com.hp.nnm.custompoller.exportdir=C:\$CustomPoller
```

- 3 NNMi 管理サーバーを再起動します。
  - 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。

### 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の変更

**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、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\nms-custompoller.properties
  - UNIX: \$NNM\_PROPS/nms-custompoller.properties

- 2 maxdiskspace エントリーを特定します。このエントリーは以下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 2,000 MB (2 GB) のストレージ容量を確保するように NNMi を設定するには、その行を以下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します。
  - 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。

## カスタムポーラーメトリックスの累積周期の変更

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、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\nms-custompoller.properties
  - UNIX: \$NNM\_PROPS/nms-custompoller.properties
- 2 以下のような行を探します。

```
#!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、その行を以下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します。
  - 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。

## 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。たとえば、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するとします。詳細については、NNMi ヘルプの「インシデントのアクションを設定する」を参照してください。

アクションのパラメーターを調整するには、次の項に示す手順に従ってください。

- ▶ 望まない結果（予期せぬメモリー使用量の増大、イベントアクション処理時間の延長など）を避けるには、イベントアクション処理のデフォルトのプロパティ値を変更しないことをお勧めします。

### 同時アクション数の設定

- ▶ 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、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\shared\%nnmaction.properties
  - UNIX: \$NNM\_PROPS/shared/nnmaction.properties

- 2 以下のような行を探します。

```
#!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、その行を以下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します。
  - 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。

### Jython アクションのスレッド数の設定

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、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\shared\%nnmaction.properties
  - UNIX: \$NNM\_PROPS/shared/nnmaction.properties

- 2 以下のような行を探します。

```
#!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、その行を以下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します。
  - 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。

## アクションサーバー名のパラメーターの設定

Windows オペレーティングシステムで NNMi 管理サーバーを実行している場合、HP NNM Action Server は Local System アカウントの Windows サービスとして実行されます。つまり、アクションサーバーでアクションを実行するには、Local System アカウントを使用する必要があります。

Windows NNMi 管理サーバーで HP NNM Action Server Windows サービスを実行するユーザー名を変更するには、HP NNM Action Server サービスの LogOn プロパティを変更します。

HP-UX、Solaris、または Linux オペレーティングシステムで NNMi 管理サーバーを実行している場合、アクションサーバーは `bin` ユーザー名で実行されます。これらのオペレーティングシステム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、以下の手順を実行します。

- 1 以下のファイルを編集します。  
`$NNM_PROPS/nnmaction.properties`

- 2 以下のような行を探します。

```
#!com.hp.ov.nms.events.action.userName=bin
```

デフォルト値ではなく、`root` 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、その行を以下のように変更します。

```
com.hp.ov.nms.events.action.userName=root
```



行の始めにある `#!` 文字を必ず削除してください。

- 3 変更を保存します。
- 4 アクションサーバーを再起動します。
  - a NNMi 管理サーバーで `ovstop nnmaction` コマンドを実行します。
  - b NNMi 管理サーバーで `ovstart nnmaction` コマンドを実行します。



## アクションサーバーのキューサイズを変更する

トラップストームへの応答など、高実行率で **Long** アクションコマンド文字列を使用するアクションの場合、アクションサーバーは多くのメモリーを使用する可能性があります。アクションサーバーのパフォーマンスを上げるために、**HP** ではアクションサーバーで利用可能なメモリーサイズが制限されています。



**Solaris NNMi** 管理サーバーの場合、**NNMi** の稼動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ーサイズを削減します。

これらの制限を変更するには、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - %NNM\_PROPS%\shared\nnmaction.properties
  - \$NNM\_PROPS/shared/nnmaction.properties
- 2 以下のような 2 行を探します。
 

```
com.hp.ov.nms.events.action.jvmargs.minMemsize=-Xms6m
com.hp.ov.nms.events.action.jvmargs.maxMemsize=-Xmx30m
```
- 3 上記のパラメーターでは、最小メモリーサイズが **6MB** に、最大が **30MB** に設定されていることがわかります。これらのパラメーターをニーズに合わせて調整します。
- 4 変更を保存します。
- 5 **NNMi** 管理サーバーを再起動します。
  - 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。

## インシデントアクションログ

アクションを実行すると、関連付けられたインシデントアクションログファイルに出力が記録されます。選択したインシデントのログの内容を表示するには、[ツール]>[インシデントアクションログ]メニューオプションを使用します。ログに含まれる項目を以下に説明します。

表 40 インシデントアクションログ項目

項目	説明
コマンド	インシデントの発生時に実行するスクリプト
インシデント名	インシデント設定で定義されたインシデント名
インシデント UUID	インシデントの <b>UUID</b> ([登録] タブ)
コマンドタイプ	コマンドのタイプ ([Jython] または [ScriptOrExecutable])
ライフサイクル状態	インシデントのライフサイクル状態 ([登録済み]、[進行中]、[完了]、または [解決済み])
終了コード	コマンドのリターンコード (エラーコードと同様)
標準出力	アクションの標準出力
標準エラー	標準エラー出力
実行ステータス	アクションごとに判別されるステータス

## hosted-on-trapstorm.conf ファイルによるトラップストームのブロック

NNMi には、ホスト元デバイス (インタフェースを含む) からのトラップストームをブロックする方法があるため、トラップストームを検出 / 抑制するために iSPI-Net ライセンスを使用する必要はありません。

- 1 nnmtrapconfig.ovpl スクリプトを実行します。nnmtrapconfig.ovpl リファレンスページまたは UNIX のマンページの説明に従って適切な `-hostedOnTrapstorm` および `-hostedOnThreshold` の値を指定し、トラップサービスを設定します。プロパティの変更を反映させるようにトラップサーバーを再設定するには、`-setProp` パラメーターを使用します。
- 2 必要に応じて既定の設定を変更するには、以下のファイルを編集します。
  - **Windows:** %NnmDataDir%\\$shared¥nnm¥conf¥hosted-on-trapstorm.conf
  - **UNIX:** \$NnmDataDir/shared/nnm/conf/hosted-on-trapstorm.conf

hosted-on-trapstorm.conf リファレンスページまたは UNIX のマンページで示された形式に従って変更します。

- 3 hosted-on-trapstorm.conf ファイルを変更した場合、nnmtrapconfig.ovpl `-stop` に続いて `nnmtrapconfig.ovpl -start` を実行することでトラップサービスを再起動する必要があります。詳細については、nnmtrapconfig.ovpl リファレンスページまたは UNIX のマンページを参照してください。

## trapFilter.conf ファイルによるインシデントのブロック

NNMi 管理サーバーに流れる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。

これが発生すると、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。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。

これを解決するには、nnmtrapd.conf ファイルを使用し、インシデントが NNMi に入るのをブロックしてインシデントトラフィックを減らしてみてください。ただし nnmtrapd.conf ファイルによる方法を使用すると、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ーに書き込みます。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はできません。詳細については、nnmtrapd.conf リファレンスページまたは UNIX のマンページを参照してください。

この問題を解決する方法には、nnmtrapd.conf ファイルを使用する方法より適切な方法があります。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ー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、インシデントのボリュームの問題を回避できます。詳細については、trapFilter.conf リファレンスページまたは UNIX のマンページを参照してください。

## 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` ファイルを以下のように編集します。

- 1 以下のファイルを編集します。
  - Windows: `%NNM_PROPS%\nms-jboss.properties`
  - UNIX: `$NNM_PROPS/nms-jboss.properties`
- 2 以下の行を含むテキストブロックを探します。  
`#!com.hp.nnm.sourceEncoding=UTF-8`
- 3 この行をコメント解除し、以下のように編集します。  
`com.hp.nnm.sourceEncoding=UTF-8`
- 4 `nms-jboss.properties` ファイルの指示と例に従って、手順3で示されたUTF-8プロパティ値を変更します。
- 5 作業内容を保存します。
- 6 NNMiを再起動します。
  - a `ovstop`
  - b `ovstart`

## 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 やその他の非暗号化アクセスを無効にできます。

- ▶ 暗号化リモートアクセスのみを許可するように NNMi を設定する前に、グローバルネットワーク管理、NNM iSPI、およびその他の統合が 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のみを許可するように NNMi を設定する前に、これらを SSL 用に設定します。

ネットワークから NNMi への HTTP やその他の非暗号化アクセスを無効にするには、`server.properties` ファイルを以下のように編集します。

- 1 以下のファイルを編集します (ファイルが存在しない場合は作成が必要な場合があります)。
  - Windows: `%NnmDataDir%\nmsas\NNM\server.properties`
  - UNIX: `$NnmDataDir/nmsas/NNM/server.properties`
- 2 `server.properties` ファイルに以下の4行を追加します。
 

```
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します。
  - 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。

上記の変更によって、NNMi はリモートシステムからの HTTP 要求を「待機」しなくなりますが、ローカルホストアクセスによる HTTP 要求はそのままサポートされます。

## 最も古い SNMP トラップインシデントの自動トリム機能の設定

NNMi が常に高いパフォーマンスを発揮するように、NNMi はデータベース内に一定数の SNMP トラップを保存した後に着信 SNMP トラップ (syslog メッセージを含む) をドロップします。最も古い SNMP トラップインシデントの自動トリム機能を使用して、NNMi データベース内に保存する SNMP トラップ数を制御し、重要な着信 SNMP トラップを保持できます。

最も古い 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。最も古い SNMP トラップインシデントの自動トリム機能を有効にすると、NNMi は NNMi データベースから最も古い SNMP トラップインシデントを削除します。



SNMP トラップインシデントを NNMi データベースから手動でトリムするには、`nnmtrimincidents.ovpl` スクリプトを使用します。詳細については、`nnmtrimincidents.ovpl` リファレンスページ、または UNIX のマンページを参照してください。

### 最も古い SNMP トラップインシデントの自動トリム機能の有効化 (インシデントアーカイブなし)

最も古い SNMP トラップインシデントの自動トリム機能を使用して、NNMi データベース内の SNMP トラップインシデント数が 60,000 個を超えた場合は 30,000 個の SNMP トラップインシデント (syslog メッセージを含む) をトリムするとします。この例では、NNMi で SNMP トラップインシデントをトリムする前にアーカイブしません。以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: `%NNM_PROPS%nms-jboss.properties`
  - UNIX: `$NNM_PROPS/nms-jboss.properties`
- 2 以下の行を含むテキストブロックを探します。  
**#!com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=50**
- 3 この行をコメント解除し、以下のように編集します。  
**com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=60**
- 4 以下の行を含むテキストブロックを探します。  
**#!com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=25**
- 5 この行をコメント解除し、以下のように編集します。  
**com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=50**
- 6 以下の行を含むテキストブロックを探します。  
**#!com.hp.nnm.events.snmpTrapAutoTrimSetting=Disabled**
- 7 この行をコメント解除し、以下のように編集します。  
**com.hp.nnm.events.snmpTrapsAutoTrimSetting=TrimOnly**
- 8 NNMi を再起動します。
  - 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。

**com.hp.nnm.events.snmpTrapMaxStoreLimit** のデフォルト値は100,000です。この設定で以下の数式を使用することで、NNMi はNNMi データベースに 60,000 個の SNMP トラップインシデント (syslog メッセージを含む) を保存した後に、NNMi データベースから 30,000 個の SNMP トラップインシデントをトリムします。

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

## 最も古いSNMPトラップインシデントの自動トリム機能の有効化 (インシデントアーカイブ有効)

最も古い SNMP トラップインシデントの自動トリム機能を使用して、NNMi データベース内の SNMP トラップインシデント数が 80,000 個を超えた場合は 60,000 個の SNMP トラップインシデント (syslog メッセージを含む) をトリムするとします。この例では、NNMi で SNMP トラップインシデントをトリムする前にアーカイブします。以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%nms-jboss.properties
  - UNIX: \$NNM\_PROPS/nms-jboss.properties
- 2 以下の行を含むテキストブロックを探します。  
**#!com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=50**
- 3 この行をコメント解除し、以下のように編集します。  
**com.hp.nnm.events.snmpTrapAutoTrimStartPercentage=80**
- 4 以下の行を含むテキストブロックを探します。  
**#!com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=25**
- 5 この行をコメント解除し、以下のように編集します。  
**com.hp.nnm.events.snmpTrapAutoTrimPercentageToDelete=75**
- 6 以下の行を含むテキストブロックを探します。  
**#!com.hp.nnm.events.snmpTrapAutoTrimSetting=Disabled**
- 7 この行を以下のように編集します。  
**com.hp.nnm.events.snmpTrapsAutoTrimSetting=TrimAndArchive**
- 8 NNMi を再起動します。
  - 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。

**com.hp.nnm.events.snmpTrapMaxStoreLimit** のデフォルト値は100,000です。この設定で以下の数式を使用することで、NNMi はNNMi データベースに 80,000 個の SNMP トラップインシデント (syslog メッセージを含む) を保存した後にアーカイブし、NNMi データベースから 60,000 個の SNMP トラップインシデントをトリムします。

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

## 保存する SNMP トラップインシデント数の削減

NNMi で長期間 SNMP トラップインシデントを保持する必要がない場合、NNMi データベースに保存する SNMP トラップインシデント数を削減できます。



NNMi は、データベース内の SNMP トラップインシデント数が 100,000 個に達すると、SNMP トラップ (syslog メッセージを含む) のドロップを開始します。この制限値をより高く設定すると NNMi のパフォーマンスが低下するため、制限値を高くすることはできません。

保存する SNMP トラップインシデント (syslog メッセージを含む) の最大数を 50,000 SNMP トラップインシデントに削減するとします。これを行うには、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\nms-jboss.properties
  - UNIX: \$NNM\_PROPS/nms-jboss.properties
- 2 以下の行を含むテキストブロックを探します。  
**#!com.hp.nnm.events.snmpTrapMaxStoreLimit=100000**
- 3 この行をコメント解除し、以下のように編集します。  
**com.hp.nnm.events.snmpTrapMaxStoreLimit=50000**
- 4 NNMi を再起動します。
  - 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。

## 最も古い SNMP トラップインシデントの自動トリム機能の監視

最も古い SNMP トラップインシデントの自動トリム機能の稼動状態を確認するには、NNMi コンソールから [ヘルプ]>[システム情報]>[ヘルス] をクリックします。NNMi は、最も古い SNMP トラップインシデントの自動トリム機能に関する以下のアラームも生成します。

- NNMi は、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) の数が **com.hp.nnm.events.snmpTrapMaxStoreLimit** 値の 100% に達したときに危険域アラームを生成します。
- NNMi は、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) の数が **com.hp.nnm.events.snmpTrapMaxStoreLimit** 値の 95% に達したときに **snmpTrapLimitMajorAlarm** アラームを生成します。
- NNMi は、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) の数が **com.hp.nnm.events.snmpTrapMaxStoreLimit** 値の 90% に達したときに **snmpTrapLimitWarningAlarm** アラームを生成します。

## 最も古い SNMP トラップインシデントの自動トリム機能の無効化

最も古い SNMP トラップインシデントの自動トリム機能を無効にするには、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\nms-jboss.properties
  - UNIX: \$NNM\_PROPS/nms-jboss.properties



- 2 以下を含むテキストブロックを探します。  
`com.hp.nnm.events.snmpTrapAutoTrimSetting`
- 3 この行をコメント解除し、以下のように編集します。  
`com.hp.nnm.events.snmpTrapAutoTrimSetting=Disabled`
- 4 NNMi を再起動します。
  - 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。

---

## SNMP MIB 変数名を表示するための NNMi ゲージタイトルの変更

NNMi 分析ペインの [ノードコンポーネント] タブには、MIB OID がポーリングされるときに NNMi コンポーネント名を表示するゲージが含まれています。これにより、コンポーネントに属するゲージを判別できます。ノードコンポーネント名は、NNMi でノードに多数のゲージが表示される場合にゲージを区別するのに役立ちます。たとえば、ノードに多数の CPU が含まれる場合、NNMi には CPU ごとに異なる名前が表示されます。この機能を無効にすると、NNMi にはすべての CPU に同じ SNMP MIB 変数名が表示されます。

NNMi コンポーネント名ではなく SNMP MIB 変数名としてゲージタイトルを表示するようにこのプロパティを変更する必要がある場合は、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\nms-ui.properties
  - UNIX: \$NNM\_PROPS/nms-ui.properties
- 2 以下の行を含むテキストブロックを探します。  
`com.hp.nnm.ui.analysisGaugeTitleIsNodeComponentName = true`
- 3 この行を以下のように編集します。  
`com.hp.nnm.ui.analysisGaugeTitleIsNodeComponentName = false`
- 4 変更を保存します。
- 5 NNMi を再起動します。
  - a `ovstop`
  - b `ovstart`

---

## NNMi 正規化プロパティの変更

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、およびフィルター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お勧めします。HP は、デプロイ中の初期検出を実行する前に、本項の設定を調整することをお勧めします。

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 **nnmnode rediscover.ovpl -all** スクリプトを実行できます。詳細については、**nnmnode rediscover.ovpl** のリファレンスページまたは UNIX のマンページを参照してください。

以下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、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - **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、以下のような行を探します。
 

```
#!com.hp.ov.nms.topo.NAME_NORMALIZATION=OFF
```

  - a プロパティをコメント解除します。
 

```
com.hp.ov.nms.topo.NAME_NORMALIZATION=OFF
```

 プロパティをコメント解除するには、行の先頭から **#!** 文字を削除します。
  - 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、以下のような行を探します。
 

```
#!com.hp.ov.nms.topo.HOSTNAME_NORMALIZATION=OFF
```

  - a プロパティをコメント解除します。
 

```
com.hp.ov.nms.topo.HOSTNAME_NORMALIZATION=OFF
```
  - 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します。
  - 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。

## 初期検出後の正規化プロパティの変更

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、**nnmnode rediscover.ovpl -all** スクリプトを実行して完全検出を開始します。



NNMi が完全検出を完了したら、以下の動作が正常に戻ります。以下はすべての例ではなく、NNMi 正規化プロパティを変更する場合に考慮する必要のある項目の一部を挙げています。

## 同時 SNMP 要求の変更

NNMi では、1つのノードに対して同時 SNMP 要求が3個に制限されています。これにより、ノードの SNMP エージェントが応答をドロップするリスクが減ります。

この値をより高く調整し、検出速度を高めることができます。ただしこの値を高く設定しすぎると、応答がドロップされるリスクが増して、検出精度が落ちます。

この制限を変更する必要がある場合は、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - **Windows:** %NNM\_PROPS%\nms-communication.properties
  - **UNIX:** \$NNM\_PROPS/nms-communication.properties
- 2 1つのノードに対する同時 SNMP 要求の現在の数を増やすには、以下の手順を実行します。
  - a 以下のような行を探します。  
#!com.hp.ov.nms.comm.snmp.maxConcurrentRequests=3
  - b プロパティをコメント解除します。  
com.hp.ov.nms.comm.snmp.maxConcurrentRequests=3
  - ▶ プロパティをコメント解除するには、行の先頭から #! 文字を削除します。
  - c 1つのノードに対する同時 SNMP 要求の目的の数に、既存の値を変更します。
  - d 変更を保存します。
- 3 NNMi 管理サーバーを再起動します。
  - 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。

## 組み込みデータベースポートの変更

組み込み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、以下の手順を実行します。

- 1 以下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
- 2 以下のような行を探します。  
#!com.hp.ov.nms.postgres.port=5432

- 3 プロパティをコメント解除します。  
com.hp.ov.nms.postgres.port=5432



プロパティをコメント解除するには、行の先頭から #! 文字を削除します。

- 4 既存の値を新しいポート番号に変更します。
- 5 変更を保存します。
- 6 NNMi 管理サーバーを再起動します。
  - 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。

## MIB ブラウザーパラメーターの変更

NNMi MIB ブラウザー ([アクション]>[MIB 情報]>[MIB を参照]メニュー)を使用して、ノードの情報を取得し、オプションの SNMP コミュニティ文字列をそのノードに指定する場合は、NNMi MIB ブラウザーは、MIB ブラウザー SNMP 通信用の `nms-ui.properties` ファイルにある MIB ブラウザーパラメーターを使用します。



MIB ブラウザーを使用するときにコミュニティ文字列を指定しない場合は、NNMi ではノードで確立されている [通信の設定] 設定 (ある場合) を使用します。これらの設定は、[設定] ワークスペースの [通信の設定] ビューを使用して NNMi コンソールで設定されます。詳細については、NNMi ヘルプの「通信プロトコルを設定する」を参照してください。

`nms-ui.properties` ファイルの MIB ブラウザーパラメーターを変更するには、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%nms-ui.properties
  - UNIX: \$NNM\_PROPS/nms-ui.properties
- 2 以下の行を含むテキストブロックを探します。  
# MIB Browser Parameters
- 3 次のテキストを含む行を検索し、# MIB Browser Parameters の下にある MIB ブラウザーパラメーターを探します。  
mibbrowser
- 4 `nms-ui.properties` ファイル内の手順に従って、MIB ブラウザーパラメーターを変更します。
- 5 変更を保存します。
- 6 NNMi を再起動します。
  - a **ovstop**
  - b **ovstart**

## NNMi 自己監視

NNMi では、メモリー、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 の稼動状態情報を表示するには、以下のいずれかの方法を使用します。

- NNMi コンソールで、[表示]>[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。
- 自己監視の詳細レポートについては、[ツール]>[NNMi システムヘルスレポート]を選択します。
- `nmmhealth.ovpl` スクリプトを実行します。

NNMi が自己監視稼動状態の例外を検出すると、NNMi により 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。以下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 以下のファイルを編集します。
  - Windows: `%NNM_PROPS%nms-ui.properties`
  - UNIX: `$NNM_PROPS/nms-ui.properties`
- 2 以下の行を含むテキストブロックを探します。
 

```
#!com.hp.nms.ui.health.disablewarning=false
```
- 3 この行をコメント解除し、以下のように編集します。
 

```
com.hp.nms.ui.health.disablewarning==true
```
- 4 変更を保存します。
- 5 NNMi を再起動します。
  - a `ovstop`
  - b `ovstart`

## 特定ノードの検出プロトコルの使用を抑える

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接続を検出します。定義されている検出プロトコルは多数あります。たとえば **Link Layer Discovery Protocol (LLDP)** は業界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収集を抑制するように NNMi を設定できます。検出プロトコル収集を抑制することで解決できる、特別な状況があります。

以下に例を挙げます。

- **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 でメモリーが不足するという問題が発生することがあります。このようなデバイスで

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、これを防止できます。これを実行するには、[検出プロトコル収集の使用の抑制](#)に示すように、デバイスの管理アドレスを `disco.SkipXdpProcessing` ファイルに追加します。



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` ファイルにリスト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、誤った接続に関与しているデバイスに対して、レイヤー 2 接続を引き出すことを推奨します。これを実行するには、[検出プロトコル収集の使用の抑制](#)で示すように、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。

## 検出プロトコル収集の使用の抑制

この収集を抑制する必要がある場合は、以下の手順を実行します。

- 1 以下のファイルを作成します。
  - **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 プロトコル収集を抑制するすべてのデバイスで、デバイスの IP アドレスを `disco.SkipXdpProcessing` ファイルに追加します。**disco.SkipXdpProcessing** リファレンスページ、または UNIX のマンページの指示に従ってください。
- 3 NNMi 管理サーバーを再起動します。
  - 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。



1 つまたは複数のノードの検出プロトコル処理を抑制すると、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。HP はこの精度低下の責任を負いません。



ovjboss サービスは起動時に `disco.SkipXdpProcessing` ファイルを読み取ります。NNMi の起動後に変更を行った場合は、この手順で示すように NNMi 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 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` リファレンスページまたは UNIX のマンページを参照してください。

## 大規模スイッチの VLAN インデックス付けの使用を抑制する

NNMi が管理対象ネットワークのスイッチデバイス間でレイヤー 2 接続を認識する方法の 1 つは、dot1dTpFdbTable (FDB) をスイッチから取得することです。ただし Cisco スイッチの場合、NNMi は VLAN-indexing 方法を使用して FDB 全体を取得する必要があります。各デバイスで設定されている VLAN の数が多い場合、VLAN-indexing による FDB の取得の完了には数時間かかることがあります。

Cisco スイッチは、多くの場合、Cisco Discovery Protocol (CDP) を使用するよう設定されています。CDP は、レイヤー 2 接続を認識するための優れた方法であるとみなされています。ネットワークのコアに配置されている大規模スイッチには、多くの VLAN が含まれていることがあります。このスイッチには一般的に、直接接続されているエンドノードがありません。管理するスイッチに直接接続されているエンドノードがない場合は、この大規模スイッチで FDB の収集を抑制するとよいでしょう。NNMi は、CDP から収集したデータを使用してレイヤー 2 検出を完了します。この大規模スイッチは、VLAN-indexing の抑制の主な候補となります。多くのエンドノードが接続している、ネットワークのエッジにある小規模スイッチ (アクセススイッチと呼ばれる) では、VLAN-indexing を抑制しないでください。

VLAN-indexing を抑制するように NNMi を設定できます。これを実行するには、「[VLAN インデックス付けの使用を抑制する](#)」(421 ページ) で示すように、NNMi 管理者が大規模スイッチの管理アドレスまたはアドレス範囲を作成して disco.NoVLANIndexing ファイルに追加する必要があります。ovjboss サービスは起動時に disco.NoVLANIndexing ファイルを読み取ります。ovjboss サービスの起動後、NNMi 管理者が disco.NoVLANIndexing ファイルを変更した場合、その変更内容は、ovjboss サービスを次回起動するまで有効になりません。デフォルトでは、disco.NoVLANIndexing ファイルは存在しません。disco.NoVLANIndexing が存在しない場合、この機能は無効になり、NNMi は VLAN-indexing を使用して、すべてのデバイスで FDB テーブル全体を収集しようとします。

## VLAN インデックス付けの使用を抑制する

この vlan-indexing を無効にする必要がある場合は、以下の手順を実行します。

- 1 以下のファイルを作成します。
  - **Windows:** %NnmDataDir%\shared\nnm\conf\disco\disco.NoVLANIndexing
  - **UNIX:** \$NnmDataDir/shared/nnm/conf/disco/disco.NoVLANIndexing

disco.NoVLANIndexing ファイルでは、大文字と小文字が区別されます。
- 2 vlan-indexing を無効にするすべてのデバイスの IP アドレスまたはアドレス範囲を disco.NoVLANIndexing ファイルに追加します。disco.NoVLANIndexing リファレンスページ、または UNIX のマンページの指示に従ってください。
- 3 NNMi 管理サーバーを再起動します。
  - 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。



1 つまたは複数のノードの vlan-indexing を抑制すると、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。HP はこの精度低下の責任を負いません。



ovjboss サービスは起動時に `disco.NoVLANIndexing` ファイルを読み取ります。NNMiの起動後に変更を行った場合は、この手順で示すようにNNMiを再起動してください。

詳細については、`disco.Disco.NoVLANIndexing` リファレンスページまたは UNIX のマンページを参照してください。

## ノードコンポーネントステータスの設定

NNMi には、ノードのステータス判別用に監視できる以下のノードコンポーネントが含まれています。

- 1 CPU
- 2 BUFFERS
- 3 VOLTAGE
- 4 TEMPERATURE
- 5 DISK\_SPACE
- 6 FAN
- 7 POWER\_SUPPLY
- 8 BACK\_PLANE
- 9 MEMORY

デフォルトでは、上記リストの最後の 4 つのノードコンポーネント (**FAN**、**POWER\_SUPPLY**、**BACK\_PLANE**、および **MEMORY**) は、ステータスをノードレベルに伝達します。たとえば、ファンが赤色のステータスインジケータを示している場合、対応するノードは黄色のステータスインジケータを受け取ります。この場合、ノードのステータスを表示しているユーザーには、ノードのコンポーネントに何らかの障害があることが警告されます。

デフォルトでは、上記リストの最初の 5 つのノードコンポーネントは、ステータスをノードレベルに伝達しません。

### ノードへのノードコンポーネントステータスの伝達

ステータスをノードレベルに伝達するようにノードコンポーネントを設定するには、以下の手順を実行します。

- 1 以下のディレクトリに `nmm-apa.properties` という名前の新しいプロパティファイルを作成します (このファイルが存在しない場合)。
  - Windows: `%NnmDataDir%\shared\%nmm%\conf\props`
  - UNIX: `$NnmDataDir/shared/nmm/conf/props`
- 2 テキストエディターを使用して、プロパティファイル内に以下のテキストを挿入します。
 

```
com.hp.ov.nms.apa.NodeComponentPropagateToNodeStatus_<Type>: true
```

 ここで、**<Type>** は「ノードコンポーネントステータスの設定」(422 ページ) の最初のリストで示されたノードコンポーネントです。
- 3 プロパティファイルを保存します。
- 4 NNMi 管理サーバーで以下の一連のコマンドを実行します。

- a ovstop
- b ovstart

## ステータスをノードに伝達しないようにノードコンポーネントを設定する

ステータスをノードレベルに伝達しないようにノードコンポーネントを設定するには、以下の手順を実行します。

- 1 以下のディレクトリに `nmn-apa.properties` という名前の新しいプロパティファイルを作成します(このファイルが存在しない場合)。

— **Windows:** `%NnmDataDir%\shared\%nmn%\conf\props`

— **UNIX:** `$NnmDataDir/shared/nmn/conf/props`

- 2 テキストエディターを使用して、プロパティファイル内に以下のテキストを挿入します。

```
com.hp.ov.nms.apa.NodeComponentNoPropagateToNodeStatus_<Type>:  
true
```

ここで、**<Type>**は「ノードコンポーネントステータスの設定」(422ページ)の最初のリストで示されたノードコンポーネントです。

- 3 プロパティファイルを保存します。
- 4 NNMi 管理サーバーで以下の一連のコマンドを実行します。

- a ovstop
- b ovstart



## ノードコンポーネントのステータス値の上書き

デフォルトでは、3つのノードコンポーネントのステータス値([なし]、[注意域]、および[利用不可])は、**Causal Engine**によって正常域ステータスにマッピングされます。以下の方法を使用することで、[なし]、[注意域]、および[利用不可]が危険域にマッピングされるように、これらのデフォルトの状態マッピングを上書きできます。

- `com.hp.ov.nms.apa.NodeComponentValueRemappedToDown_None`
- `com.hp.ov.nms.apa.NodeComponentValueRemappedToDown_Warning`
- `com.hp.ov.nms.apa.NodeComponentValueRemappedToDown_Unavailable`

ノードコンポーネントのステータス値を上書きするには、以下の手順を実行します。

- 1 以下のディレクトリに `nm-apa.properties` という名前の新しいプロパティファイルを作成します(このファイルが存在しない場合)。

— **Windows:** `%NnmDataDir%\shared\%nm%\conf\props`

— **UNIX:** `$NnmDataDir/shared/nm/conf/props`

- 2 テキストエディターを使用して、プロパティファイル内に必要に応じて以下の行の1つ、2つ、または3つすべてを挿入します。

```
com.hp.ov.nms.apa.NodeComponentValueRemappedToDown_None: true
```

```
com.hp.ov.nms.apa.NodeComponentValueRemappedToDown_Warning: true
```

```
com.hp.ov.nms.apa.NodeComponentValueRemappedToDown_Unavailable: true
```

- 3 プロパティファイルを保存します。
- 4 **NNMi** 管理サーバーで以下の一連のコマンドを実行します。
  - a `ovstop`
  - b `ovstart`



[利用不可]の状態を[未ポーリング]状態にマッピングできます([利用不可]は測定機能が利用できないことを指すため)。この状態は、多くの場合コンポーネントの機能不全ではなくセンサーの機能不全で発生します。[利用不可]を[未ポーリング]にマッピングするには、**手順 2**で以下のテキストを使用する以外は上記と同じ手順を実行します。

```
com.hp.ov.nms.apa.NodeComponentValueRemappedToUnpolled_Unavailable: true
```



# NNMi ロギング

---

## NNMi ログファイル

HP Network Node Manager i Software (NNMi) のパフォーマンスを調べる、または NNMiのプロセスとサービスがどのように動作しているかを観察するには、プロセスとサービスアクティビティの履歴を表示するログファイルを確認できます。これらのファイルは、以下の場所にあります。

- **Windows:** %NnmDataDir%\log\%nm%
- **UNIX:** \$NnmDataDir/log/nnm

NNMiでは、name.logという形式のファイル名でログファイルが保存されます。アーカイブされたログファイルには、name.log.%gという形式で番号が追加されます。

- name は、ログファイルのベース名です。
- %g は、アーカイブされたログファイルのアーカイブ番号です。最も高いアーカイブ番号は最も古いファイルを示します。

ログファイルは、そのサイズが設定した制限を超えたときにアーカイブされる可能性があります。ログファイルのサイズが設定した制限を超えると、最後のアクティブなログファイルがアーカイブされます。たとえば、nnm.log ファイルを nnm.log.1 ファイルとしてアーカイブした後に、NNMi は新しい nnm.log ファイルへの記録を開始します。

NNMi では、以下のロギングレベルでメッセージがログに記録されます。

- **SEVERE:** NNMi の異常動作に関連するイベント。
- **WARNING:** 潜在的な問題を示すイベント、および **SEVERE** ロギングレベルに含まれるすべてのメッセージ。
- **INFO:** NNMiコンソール(またはそれと同等のもの)書き込まれるメッセージ、および **WARNING** ロギングレベルに含まれるすべてのメッセージ。

## ロギングファイルのプロパティの変更

NNMi には、NNMi ロギングを変更できるいくつかの機能があります。このセクションでは、これらの機能の調整方法について説明します。

### ロギングのサインインおよびサインアウト

NNMi 9.20は、各ユーザーによるNNMiコンソールへのサインインまたはサインアウトのログを生成するように設定されていません。サインインおよびサインアウトアクティビティを記録するように NNMi を設定するには、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - **Windows:** %NNM\_DATA%\shared\nnm\conf\props\nnm-logging.properties
  - **UNIX:** \$NNM\_DATA/shared/nnm/conf/props/nnm-logging.properties
- 2 以下の行を含むテキストブロックを探します。

```
com.hp.ov.nnm.log.signin.level = OFF
```
- 3 この行を以下のように変更します。

```
com.hp.ov.nnm.log.signin.level = INFO
```
- 4 変更を保存します。
- 5 NNMi を再起動します。
  - a **ovstop** を実行
  - b **ovstart** を実行

# NNMi セキュリティ

本章には、以下のトピックがあります。

- Web アクセスおよび RMI 通信に SSL 通信を設定する
- 非ルート UNIX ユーザーに NNMi の開始と停止を許可する
- 組み込みデータベースツールのパスワードの入力

---

## Web アクセスおよび RMI 通信に SSL 通信を設定する

NNMi には、Web アクセスおよび Java Remote Method Invocation (RMI) 通信で Secure Sockets Layer (SSL) を設定するのに使用される一連のデフォルト暗号が含まれています。暗号は `nms-jboss.properties` ファイルにリストされています。



HP の承認なしに暗号リストから暗号を追加または削除しないでください。これを行うと、製品に障害が発生したり、製品が動作しなくなる可能性があります。

---

## 非ルート UNIX ユーザーに NNMi の開始と停止を許可する



`/opt/ov` ディレクトリが `nosuid` オプションセットを含むパーティション上にある場合は、非ルートユーザー機能を利用できません。パーティションが `nosuid` オプションセットを使用して設定されているかどうかを判別するには、`/etc/fstab` を参照してください。

NNMi には、非ルート UNIX ユーザーに NNMi の開始と停止を許可する方法があります。以下の手順を実行します。

- 1 ルートとして、以下のファイルを編集します。  
`$NnmDataDir/shared/nnm/conf/ovstart.allow`
- 2 NNMi の開始と停止を許可する非ルートユーザーを含めます (1 行に 1 ユーザー)。
- 3 変更を保存します。

---

## 組み込みデータベースツールのパスワードの入力

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、または **UNIX** のマンページを参照してください。

**HA**環境のプライマリクラスターノードのみで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ー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。

# 追加情報

この項では以下の付録について説明します。

- アプリケーションフェイルオーバー構成の NNMi の手動設定
- NNMi 環境変数
- NNMi 9.20 およびウェルノウンポート
- NNMi 9.20 iSPI のウェルノウンポート
- 設定変更の提案



# アプリケーションフェイルオーバー構成の NNMiの手動設定

この付録の手順では、NNMi クラスター設定ウィザードを使用しないでアプリケーションフェイルオーバーを設定する方法を説明します。

▶ **Oracle** データベースでアプリケーションフェイルオーバーを使用している場合、この付録の設定手順を実行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を手動で設定するには、以下の手順を実行します。

- 1 両方のノードで **ovstop** を実行します。
- 2 `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します。以下の手順を実行します。

▶ 以下の手順では、ファイルのテキストブロックの行のコメントを解除し、テキストを変更することを**編集**と呼びます。

- a 以下のファイルを編集します。

— **Windows:** `%NnmDataDir%\¥shared¥nmm¥conf¥props¥nms-cluster.properties`

— **UNIX:** `$NnmDataDir/shared/nmm/conf/props/nms-cluster.properties`

- b NNMi クラスター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。

```
com.hp.ov.nms.cluster.name=MyCluster
```

- c `nms-cluster.properties` ファイルの `com.hp.ov.nms.cluster.member.hostnames` パラメーターに、クラスター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
```

▶ NNMi 9.0x では、アプリケーションフェイルオーバー機能で UDP ソリューションがサポートされ、クラスターホストはネットワークで自動的に検出されました。NNMi 9.1x からは UDP ソリューションが排除され、TCP ソリューションのみがサポートされます。NNMi 9.0x から移行する場合は、アプリケーションフェイルオーバーを機能させるために**手順c**を完了してクラスターホスト名を定義する必要があります。

- d オプション。`nms-cluster.properties` ファイル内のその他の `com.hp.ov.nms.cluster*` パラメーターを定義します。各パラメーターの変更方法については、`nms-cluster.properties` ファイル内の指示に従ってください。

▶ **Oracle** データベースでアプリケーションフェイルオーバーを使用している場合、NNMi では `nms-cluster.properties` ファイルに含まれるデータベースパラメーターが無視されます。

- 3 選択した方法によって、「自己署名証明書を使用するようにアプリケーションフェイルオーバーを設定する」(132 ページ) に示した指示、または、「認証機関を使用するようにアプリケーションフェイルオーバーを設定する」(134 ページ) に示した指示を実行します。



アプリケーションフェイルオーバー機能を設定するときには、両方のノードの `nnm.keystore` および `nnm.truststore` ファイルの内容をマージして、`nnm.keystore` および `nnm.truststore` を 1 つのファイルにする必要があります。方法を選択し、手順 3 の 1 セットの指示を完了する必要があります。

- 4 以下のファイルをサーバー X からサーバー Y にコピーします。

- **Windows:**  
`%NnmDataDir%\shared\%nnm%\conf\%nnmcluster%\cluster.keystore`
- **UNIX:**  
`$NnmDataDir/shared/nnm/conf/nnmcluster/cluster.keystore`

- 5 次のコマンドをサーバー X とサーバー Y の両方で実行します：**nnmcluster** 各サーバーに、以下のように表示されます。

```
===== Current cluster state =====
State ID: 000000001000000005
Date/Time: 15 Mar 2011 - 09:37:58 (GMT-0600)
Cluster name: ThisCluster (key CRC:626,187,650)
Automatic failover: Enabled
NNM database type: Embedded
NNM configured ACTIVE node is: NO_ACTIVE
NNM current ACTIVE node is: NO_ACTIVE
Cluster members are:

  Local?      NodeType  State           OvStatus      Hostname/Address
  -----
* 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
```

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、以下の点を確認して、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、以下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。このエラーが表示されるのは、**nnmcluster** コマンドを実行した後です。
- 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。たとえば、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。このエラーが表示されるのは、**nnmcluster** コマンドを実行した後です。



- 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。たとえば、サーバー X が NNMi 9.20 を実行しており、サーバー Y が NNMi 9.20 パッチ 1 (リリース後) を実行している場合などです。このエラーが表示されるのは、**nnmcluster** コマンドを実行した後です。

6 サーバー X で、NNMi クラスタマネージャーを開始します。

**nnmcluster -daemon**

**nnmcluster -daemon** コマンドを NNMi 管理サーバー X で実行すると、NNMi クラスタマネージャーが以下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** リファレンスページ、または UNIX のマンページを参照してください。

7 サーバー X がクラスタの最初のアクティブノードになるまで数分待ちます。サーバー X で **nnmcluster -display** コマンドを実行し、表示された結果から ACTIVE\_NNM\_STARTING または ACTIVE\_SomeOtherState の「ACTIVE」という語を検索します。サーバー X がアクティブノードであることを確認するまで手順 8 に進まないでください。

8 サーバー Y で NNMi クラスタマネージャーを開始します。

**nnmcluster -daemon**

**nnmcluster -daemon** コマンドを NNMi 管理サーバー Y で実行すると、NNMi クラスタマネージャーが以下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。ディスプレイに 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。ディスプレイに 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。ディスプレイに 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。ディスプレイに STANDBY\_READY と表示されます。

詳細については、**nnmcluster** リファレンスページ、または UNIX のマンページを参照してください。

- 9 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ります。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) にログオンします。NNMi ユーザーに、サーバー X (アクティブ NNMi 管理サーバー) とサーバー Y (スタンバイ NNMi 管理サーバー) 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。
- 10 ネットワークオペレーションセンター (NOC) の担当者に、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デバイスを設定するように指示します。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。

# NNMi 環境変数

HP Network Node Manager i Software (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。

この付録では、以下の内容を記載しています。

- このドキュメントで使用する環境変数
- 他の使用可能な環境変数

## このドキュメントで使用する環境変数

このドキュメントでは、主に以下の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。以下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 Server 2008:**

- %NnmInstallDir%: <drive>%Program Files (x86)%HP%HP BTO Software
- %NnmDataDir%: <drive>%ProgramData%HP%HP BTO Software

▶ 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

▶ 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、「他の使用可能な環境変数」(435 ページ) を参照してください。

## 他の使用可能な環境変数

NNMi 管理者は、いくつかの NNMi ファイルの場所に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さまざまな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)%HP%HP BTO Software%bin%nmn.envvars.bat"
- **UNIX:** . /opt/OV/bin/nnm.envvars.sh

上記の各 OS 用のコマンドを実行した後で、表 41 (Windows) または表 42 (UNIX) で示す NNMi 環境変数を使用して、頻繁に使用する NNMi ファイルの場所に移動できます。

**表 41 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)\HP\HP BTO Software\bin
%NNM_CONF%	C:\ProgramData\HP\HP BTO Software\conf
%NNM_DATA%	C:\ProgramData\HP\HP BTO Software\
%NNM_DB%	C:\ProgramData\HP\HP BTO Software\shared\%nnm%\databases
%NNM_JAVA%	C:\Program Files (x86)\HP\HP BTO Software\nonOV\jdk\%nnm%\bin\java.exe
%NNM_JAVA_DIR%	C:\Program Files (x86)\HP\HP BTO Software\java
%NNM_JAVA_PATH_SEP%	;
%NNM_JBOSS%	C:\Program Files (x86)\HP\HP BTO Software\nmsas
%NNM_JBOSS_DEPLOY%	C:\Program Files (x86)\HP\HP BTO Software\nmsas\server\nms\deploy
%NNM_JBOSS_LOG%	C:\Program Files (x86)\HP\HP BTO Software\nmsas\server\nms\log
%NNM_JBOSS_SERVERCONF%	C:\Program Files (x86)\HP\HP BTO Software\nmsas\server\nms
%NNM_JRE%	C:\Program Files (x86)\HP\HP BTO Software\nonOV\jdk\%nnm
%NNM_LOG%	C:\ProgramData\HP\HP BTO Software\log
%NNM_LRF%	C:\ProgramData\HP\HP BTO Software\shared\%nnm%\lrf
%NNM_PRIV_LOG%	C:\ProgramData\HP\HP BTO Software\log
%NNM_PROPS%	C:\ProgramData\HP\HP BTO Software\shared\%nnm%\conf\props
%NNM_SHARED_CONF%	C:\ProgramData\HP\HP BTO Software\shared\%nnm%\conf
%NNM_SHARE_LOG%	C:\ProgramData\HP\HP BTO Software\log
%NNM_SNMP_MIBS%	C:\Program Files (x86)\HP\HP BTO Software\misc\%nnm%\snmp_mibs
%NNM_SUPPORT%	C:\Program Files (x86)\HP\HP BTO Software\support
%NNM_TMP%	C:\ProgramData\HP\HP BTO Software\tmp
%NNM_USER_SNMP_MIBS%	C:\ProgramData\HP\HP BTO Software\shared\%nnm%\user-snmp-mibs
%NNM_WWW%	C:\ProgramData\HP\HP BTO Software\shared\%nnm%\www

表 42 UNIX OS での環境変数のデフォルトの場所

変数	HP-UX
\$NNM_BIN	/opt/OV/bin
\$NNM_CONF	/var/opt/OV/conf
\$NNM_DATA	/var/opt/OV
\$NNM_DB	/var/opt/OV/shared/nnm/databases
\$NNM_JAVA	/opt/OV/nonOV/jdk/nnm/bin/java
\$NNM_JAVA_DIR	/opt/OV/java
\$NNM_JAVA_PATH_SEP	:
\$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_TMP	/var/opt/OV/tmp
\$NNM_USER_SNMP_MIBS	/var/opt/OV/shared/nnm/user-snmp-mibs
\$NNM_WWW	/var/opt/OV/shared/nnm/www



# NNMi 9.20 およびウェルノウンポート

表 43 に、NNMi が使用する管理サーバーのポートを示します。NNMi はそれらのポートで待機します。ポートの競合が発生した場合は、「設定の変更」列の説明に従ってそのポート番号のほとんどを変更できます。詳細については、[nmm.ports](#) リファレンスページ、または [UNIX](#) マンページ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を正しく機能させるために、TCP ポート 7800-7810 をオープンにしてください。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 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。

表 43 NNMi 管理サーバー で使用されるポート

ポート	タイプ	名前	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 と Web サービスで使用  - 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  インストール時に変更することもできます
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。詳細については、 <a href="#">nmmtrapconfig.ovpl</a> リファレンスページまたは <a href="#">UNIX</a> のマンページを参照してください。
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 サービスで使用	nms-local.properties ファイルを変更します

表 43 NNMi 管理サーバー（続き）で使用されるポート

ポート	タイプ	名前	目的	設定の変更
1098	TCP	nmsas.server.port.naming.rmi	<ul style="list-style-type: none"> <li>- 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。</li> <li>- 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のみに制限することをお勧めします。</li> </ul>	nms-local.properties ファイルを変更します
1099	TCP	nmsas.server.port.naming.port	<ul style="list-style-type: none"> <li>- 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。</li> <li>- 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のみに制限することをお勧めします。</li> </ul>	nms-local.properties ファイルを変更します
3873	TCP	nmsas.server.port.remoting.ejb3	<ul style="list-style-type: none"> <li>- 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。</li> <li>- 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のみに制限することをお勧めします。</li> </ul>	nms-local.properties ファイルを変更します
4444	TCP	nmsas.server.port.jmx.jrmp	<ul style="list-style-type: none"> <li>- 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。</li> <li>- 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のみに制限することをお勧めします。</li> </ul>	nms-local.properties ファイルを変更します



表 43 NNMi 管理サーバー（続き）で使用されるポート

ポート	タイプ	名前	目的	設定の変更
4445	TCP	nmsas.server.port.jmx.rmi	<ul style="list-style-type: none"> <li>-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。</li> <li>- 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のみに制限することをお勧めします。</li> </ul>	nms-local.properties ファイルを変更します
4446	TCP	nmsas.server.port.invoker.unified	<ul style="list-style-type: none"> <li>-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。</li> <li>- 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のみに制限することをお勧めします。</li> </ul>	nms-local.properties ファイルを変更します
4457	TCP	nmsas.server.port.hq	<ul style="list-style-type: none"> <li>- 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。</li> <li>- メッセージングでは、グローバルマネージャからリージョナルマネージャへ通信が行われます。</li> <li>- このポートが開くと、双方向となります。</li> </ul>	nms-local.properties ファイルを変更します
4459	TCP	nmsas.server.port.hq.ssl	<ul style="list-style-type: none"> <li>- 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。</li> <li>- メッセージングでは、グローバルマネージャからリージョナルマネージャへ通信が行われます。</li> <li>- このポートが開くと、双方向となります。</li> </ul>	nms-local.properties ファイルを変更します
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
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

表 43 NNMi 管理サーバー（続き）で使用されるポート

ポート	タイプ	名前	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
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
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
8886	TCP	OVSPMD_MGMT	NNMi ovspmd (プロセスマネージャー) 管理ポート	/etc/services ファイルを変更します
8887	TCP	OVSPMD_REQ	NNMi ovsmppd (プロセスマネージャー) 要求ポート	/etc/services ファイルを変更します

表 44 に、他のシステムとの通信で NNMi が使用するポートの一部を示します。ファイアウォールによって NNMi がこれらのシステムから分断されている場合は、その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。実際のポートセットは、NNMi で使用するように設定した統合セットと、それらの統合の設定方法に応じて異なります。4 列目がクライアントであれば NNMi はそのポートに接続または送信し、4 列目がサーバーであれば NNMi はそのポートで待機します。

表 44 NNMi 管理サーバーと他のシステム の通信で使用されるポート

ポート	タイプ	目的	クライアント、サーバー
80	TCP	NNMi のデフォルト HTTP ポート、Web UI と Web サービスで使用	サーバー
80	TCP	NNMi が他のアプリケーションに接続するときのデフォルト HTTP ポート。実際のポートは NNMi の設定によって異なります。	クライアント
161	UDP	SNMP 要求ポート	クライアント
162	UDP	SNMP トラップポート - NNMi が受信するトラップ	サーバー
162	UDP	SNMP トラップポート。トラップ転送、Northbound インタフェース、または NetCool 統合	クライアント

表 44 NNMi 管理サーバーと他のシステム ( 続き ) の通信で使用されるポート

ポート	タイプ	目的	クライアント、サーバー
389	TCP	デフォルト LDAP ポート	クライアント
395	UDP	nGenius Probe SNMP トラップポート	クライアント
443	TCP	NNMi が他のアプリケーションに接続するときのデフォルトのセキュアーHTTPS ポート、実際のポートは NNMi の設定によって異なります。 HP OM on Windows のデフォルト HTTPS ポート	クライアント
443	TCP	デフォルトのセキュアーHTTPS ポート、Web UI と Web サービスで使用	サーバー
636	TCP	デフォルトのセキュアー LDAP ポート (SSL)	クライアント
1741	TCP	デフォルトの CiscoWorks LMS Web サービスポート	クライアント
4457	TCP	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。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に対して接続を行います。	クライアント、サーバー
4459	TCP	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。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に対して接続を行います。	クライアント、サーバー
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	クライアントとサーバー
8004	TCP	別の Web サーバーがすでにポート 80 を使用している場合の NNMi のデフォルト HTTP ポート。Web UI と Web サービスで使用。NNMi 管理サーバーの実際の HTTP ポートを検証します。	サーバー
8080	TCP	NNMi と同じシステムにインストールされている場合に、NA に接続するときのデフォルト HTTP ポート。 HP UCMDB Web サービスのデフォルト HTTPS ポート	クライアント
8443 または 8444	TCP	HP OM for UNIX に接続するときのデフォルト HTTP ポート	クライアント
9300	TCP	NNM iSPI Performance for Metrics に接続するときのデフォルト HTTP ポート	クライアント
50000	TCP	SIM に接続するときのデフォルト HTTPS ポート	クライアント

- ▶ 検出のために ICMP 障害ポーリングまたは ping スweepを使用するように NNMi を設定する場合は、ICMP パケットを通過させるようにファイアウォールを設定してください。
- ▶ NNMi-HP OM 統合の Web サービス方式は、ファイアウォールを介して機能することはありませんが、Northbound インタフェースを使用する NNMi-HP OM 統合はファイアウォールを介して機能します。

グローバルネットワーク管理機能を使用する場合は、グローバル NNMi 管理サーバーから地域 NNMi 管理サーバーに対して、表 45 に示すウェルノウン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、TCP アクセス用にグローバル 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、これらのポートが開いている必要があります。リージョナル NNMi 管理サーバーが逆に、グローバル NNMi 管理サーバーに対してソケットを開くことはありません。

表 45 グローバルネットワーク管理で必須のアクセス可能ソケット

セキュリティ	パラメーター	TCP ポート
非 SSL	jboss.http.port	80
	jboss.bisocket.port	4457
SSL	jboss.https.port	443
	jboss.sslbisocket.port	4459

# NNMi 9.20 iSPI のウェルノウンポート

表 46 に、HP Network Node Manager iSPI for MPLS Software が使用する管理サーバーのポートを示します。ポートが競合する場合、%NnmDataDir%/nmsas/mpls/server.properties にある server.properties ファイルを使用してこれらのポート番号のほぼすべてを変更できます。

**表 46 HP Network Node Manager iSPI for MPLS Software 管理サーバー で使用されるポート**

ポート	タイプ	名前	目的	設定の変更
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。このポートは、nms-local.properties ファイルで NNMi に設定するポートと同じです。	N/A
24040	TCP	nmsas.server.port.web.http	Web UI で使用される、デフォルトの HTTP ポート。	server.properties ファイルを変更します。インストール時に変更することもできます。
24041	TCP	nmsas.server.port.remoting.ejb3	デフォルトの EJB3 リモートコネクタポート	server.properties ファイルを変更します。
24043	TCP	nmsas.server.port.web.https	Web UI で使用される、デフォルトのセキュア HTTPS ポート (SSL)。	server.properties ファイルを変更します。インストール時に変更することもできます。
24044	TCP	nmsas.server.port.jmx.jrmp	デフォルトの RMI オブジェクトポート (JRMP 呼び出し元)	server.properties ファイルを変更します。
24045	TCP	nmsas.server.port.invoker.unified	デフォルトの RMI リモートサーバーコネクタポート	server.properties ファイルを変更します。
24046	TCP	nmsas.server.port.naming.port	デフォルトのブートストラップ JNP サービスポート (JNDI プロバイダー)	server.properties ファイルを変更します。インストール時に変更することもできます。
24047	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。	server.properties ファイルを変更します。
24048	TCP	nmsas.server.port.jmx.rmi	デフォルトの RMI プール済み呼び出し元ポート	server.properties ファイルを変更します。

表 46 HP Network Node Manager iSPI for MPLS Software 管理サーバー ( 続き ) で使用されるポート

ポート	タイプ	名前	目的	設定の変更
24049	TCP	nmsas.server.port.naming.rmi	RMI ネームサービスのデフォルトポート	server.properties ファイルを変更します。
24092	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。	server.properties ファイルを変更します。
24712	TCP	nmsas.server.port.ts.recovery	トランザクションサービスで使用するデフォルトの復旧ポート。	server.properties ファイルを変更します。
24713	TCP	nmsas.server.port.ts.status	トランザクションサービスで使用するデフォルトのステータスポート。	server.properties ファイルを変更します。
24714	TCP	nmsas.server.port.ts.id	トランザクションサービスで使用するデフォルトポート。	server.properties ファイルを変更します。

表 47 に、NNM iSPI for IP Telephony が使用する管理サーバーのポートを示します。ポートが競合する場合、%NnmDataDir%/nmsas/ipt/server.properties にある server.properties ファイルを使用してこれらのポート番号のほぼすべてを変更できます。

表 47 NNM iSPI for IP Telephony 管理サーバー で使用されるポート

ポート	タイプ	名前	目的	設定の変更
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。このポートは、nms-local.properties ファイルで NNMi に設定するポートと同じです。	N/A
10080	TCP	nmsas.server.port.web.http	Web UI で使用される、デフォルトの HTTP ポート。	server.properties ファイルを変更します。インストール時に変更することもできます。
10083	TCP	nmsas.server.port.naming.rmi	RMI ネームサービスのデフォルトポート	server.properties ファイルを変更します。
10084	TCP	nmsas.server.port.jmx.jrmp	デフォルトの RMI オブジェクトポート (JRMP 呼び出し元)	server.properties ファイルを変更します。
10085	TCP	nmsas.server.port.jmx.rmi	デフォルトの RMI プール済み呼び出し元ポート	server.properties ファイルを変更します。
10086	TCP	nmsas.server.port.invoker.unified	デフォルトの RMI リモートサーバーコネクタポート	server.properties ファイルを変更します。

表 47 NNM iSPI for IP Telephony 管理サーバー（続き）で使用されるポート

ポート	タイプ	名前	目的	設定の変更
10087	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。	server.properties ファイルを変更します。
10089	TCP	nmsas.server.port.remoting.ejb3	デフォルトの EJB3 リモートコネクタポート	server.properties ファイルを変更します。
10092	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。	server.properties ファイルを変更します。
10099	TCP	nmsas.server.port.naming.port	デフォルトのブートストラップ JNP サービスポート (JNDI プロバイダー)	server.properties ファイルを変更します。インストール時に変更することもできます。
10443	TCP	nmsas.server.port.web.https	Web UI で使用される、デフォルトのセキュア HTTPS ポート (SSL)。	server.properties ファイルを変更します。インストール時に変更することもできます。
14712	TCP	nmsas.server.port.ts.recovery	トランザクションサービスで使用するデフォルトの復旧ポート。	server.properties ファイルを変更します。
14713	TCP	nmsas.server.port.ts.status	トランザクションサービスで使用するデフォルトのステータスポート。	server.properties ファイルを変更します。
14714	TCP	nmsas.server.port.ts.id	トランザクションサービスで使用するデフォルトポート。	server.properties ファイルを変更します。

表 48 に、NNM iSPI for IP Multicast が使用する管理サーバーのポートを示します。ポートが競合する場合、%NnmDataDir%/nmsas/multicast/server.properties にある **server.properties** ファイルを使用してこれらのポート番号のほぼすべてを変更できます。

**表 48 NNM iSPI for IP Multicast 管理サーバー で使用されるポート**

ポート	タイプ	名前	目的	設定の変更
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。このポートは、 <b>nms-local.properties</b> ファイルで NNMi に設定するポートと同じです。	N/A
8084	TCP	nmsas.server.port.web.http	Web UI で使用される、デフォルトの HTTP ポート。	<b>server.properties</b> ファイルを変更します。インストール時に変更することもできます。
14083	TCP	nmsas.server.port.naming.rmi	RMI ネームサービスのデフォルトポート	<b>server.properties</b> ファイルを変更します。
14084	TCP	nmsas.server.port.jmx.jrmp	デフォルトの RMI オブジェクトポート (JRMP 呼び出し元)	<b>server.properties</b> ファイルを変更します。
14085	TCP	nmsas.server.port.jmx.rmi	デフォルトの RMI プール済み呼び出し元ポート	<b>server.properties</b> ファイルを変更します。
14086	TCP	nmsas.server.port.invoker.unified	デフォルトの RMI リモートサーバーコネクタポート	<b>server.properties</b> ファイルを変更します。
14087	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。	<b>server.properties</b> ファイルを変更します。
14089	TCP	nmsas.server.port.remoting.ejb3	デフォルトの EJB3 リモートコネクタポート	<b>server.properties</b> ファイルを変更します。
14092	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。	<b>server.properties</b> ファイルを変更します。
14099	TCP	nmsas.server.port.naming.port	デフォルトのブートストラップ JNP サービスポート (JNDI プロバイダー)	<b>server.properties</b> ファイルを変更します。インストール時に変更することもできます。
14102	TCP	nmsas.server.port.ts.id	トランザクションサービスで使用するデフォルトポート。	<b>server.properties</b> ファイルを変更します。



表 48 NNM iSPI for IP Multicast 管理サーバー ( 続き ) で使用されるポート

ポート	タイプ	名前	目的	設定の変更
14103	TCP	nmsas.server.port.ts.recovery	トランザクションサービスで使用するデフォルトの復旧ポート。	server.properties ファイルを変更します。
14104	TCP	nmsas.server.port.ts.status	トランザクションサービスで使用するデフォルトのステータスポート。	server.properties ファイルを変更します。
14443	TCP	nmsas.server.port.web.https	Web UI で使用される、デフォルトのセキュア HTTPS ポート (SSL)。	server.properties ファイルを変更します。インストール時に変更することもできます。

表 49 に、NNM iSPI Performance for Traffic ( トラフィックマスターコンポーネント ) が使用する管理サーバーのポートを示します。ポートが競合する場合、%NnmDataDir%/nmsas/traffic-master/server.properties にある server.properties ファイルを使用してこれらのポート番号のほぼすべてを変更できます。

表 49 NNM iSPI Performance for Traffic 管理サーバー ( トラフィックマスター ) で使用されるポート

ポート	タイプ	名前	目的	設定の変更
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。このポートは、nms-local.properties ファイルで NNMi に設定するポートと同じです。	N/A
12080	TCP	nmsas.server.port.web.http	Web UI で使用される、デフォルトの HTTP ポート。	server.properties ファイルを変更します。インストール時に変更することもできます。
12081	TCP	nmsas.server.port.web.https	Web UI で使用される、デフォルトのセキュア HTTPS ポート (SSL)。	server.properties ファイルを変更します。インストール時に変更することもできます。
12083	TCP	nmsas.server.port.naming.rmi	RMI ネームサービスのデフォルトポート	server.properties ファイルを変更します。
12084	TCP	nmsas.server.port.jmx.jrmp	デフォルトの RMI オブジェクトポート (JRMP 呼び出し元)	server.properties ファイルを変更します。
12085	TCP	nmsas.server.port.jmx.rmi	デフォルトの RMI プール済み呼び出し元ポート	server.properties ファイルを変更します。

表 49 NNM iSPI Performance for Traffic 管理サーバー (トラフィックマスター) で使用されるポート (続き)

ポート	タイプ	名前	目的	設定の変更
12086	TCP	nmsas.server.port.invoker.unified	デフォルトの RMI リモートサーバーコネクターポート	server.properties ファイルを変更します。
12087	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。	server.properties ファイルを変更します。
12089	TCP	nmsas.server.port.remoting.ejb3	デフォルトの EJB3	server.properties ファイルを変更します。
12092	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。	server.properties ファイルを変更します。
12099	TCP	nmsas.server.port.naming.port	デフォルトのブートストラップ JNP サービスポート (JNDI プロバイダー)	server.properties ファイルを変更します。インストール時に変更することもできます。
12712	TCP	nmsas.server.port.ts.recovery	トランザクションサービスで使用するデフォルトの復旧ポート。	server.properties ファイルを変更します。
12713	TCP	nmsas.server.port.ts.status	トランザクションサービスで使用するデフォルトのステータスポート。	server.properties ファイルを変更します。
12714	TCP	nmsas.server.port.ts.id	トランザクションサービスで使用するデフォルトポート。	server.properties ファイルを変更します。

表 50 に、NNM iSPI Performance for Traffic (トラフィックリーフコンポーネント) が使用する管理サーバーのポートを示します。ポートが競合する場合、%NnmDataDir%/nmsas/traffic-leaf/server.properties にある server.properties ファイルを使用してこれらのポート番号のほぼすべてを変更できます。

表 50 NNM iSPI Performance for Traffic 管理サーバー (トラフィックリーフ) で使用されるポート

ポート	タイプ	名前	目的	設定の変更
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。このポートは、nms-local.properties ファイルで NNMi に設定するポートと同じです。	N/A
11080	TCP	nmsas.server.port.web.http	Web UI で使用される、デフォルトの HTTP ポート。	server.properties ファイルを変更します。インストール時に変更することもできます。
11081	TCP	nmsas.server.port.web.https	Web UI で使用される、デフォルトのセキュア HTTPS ポート (SSL)。	server.properties ファイルを変更します。インストール時に変更することもできます。
11083	TCP	nmsas.server.port.naming.rmi	RMI ネームサービスのデフォルトポート	server.properties ファイルを変更します。
11084	TCP	nmsas.server.port.jmx.jrmp	デフォルトの RMI オブジェクトポート (JRMP 呼び出し元)	server.properties ファイルを変更します。
11085	TCP	nmsas.server.port.jmx.rmi	デフォルトの RMI プール済み呼び出し元ポート	server.properties ファイルを変更します。
11086	TCP	nmsas.server.port.invoker.unified	デフォルトの RMI リモートサーバーコネクターポート	server.properties ファイルを変更します。
11087	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。	server.properties ファイルを変更します。
11089	TCP	nmsas.server.port.remoting.ejb3	デフォルトの EJB3 リモートコネクターポート	server.properties ファイルを変更します。
11092	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。	server.properties ファイルを変更します。
11099	TCP	nmsas.server.port.naming.port	デフォルトのブートストラップ JNP サービスポート (JNDI プロバイダー)	server.properties ファイルを変更します。インストール時に変更することもできます。

表 50 NNM iSPI Performance for Traffic 管理サーバー (トラフィックリーフ) で使用されるポート (続き)

ポート	タイプ	名前	目的	設定の変更
11712	TCP	nmsas.server.port.ts.recovery	トランザクションサービスで使用するデフォルトの復旧ポート。	server.properties ファイルを変更します。
11713	TCP	nmsas.server.port.ts.status	トランザクションサービスで使用するデフォルトのステータスポート。	server.properties ファイルを変更します。
11714	TCP	nmsas.server.port.ts.id	トランザクションサービスで使用するデフォルトポート。	server.properties ファイルを変更します。

表 51 に、NNM iSPI Performance for QA が使用する管理サーバーのポートを示します。ポートが競合する場合、%NnmDataDir%/nmsas/qa/server.properties にある server.properties ファイルを使用してこれらのポート番号のほぼすべてを変更できます。

表 51 NNM iSPI Performance for QA 管理サーバー で使用されるポート

ポート	タイプ	名前	目的	設定の変更
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。このポートは、nms-local.properties ファイルで NNMi に設定するポートと同じです。	N/A
54040	TCP	nmsas.server.port.web.http	Web UI で使用される、デフォルトの HTTP ポート。	server.properties ファイルを変更します。インストール時に変更することもできます。
54043	TCP	nmsas.server.port.web.https	Web UI で使用される、デフォルトのセキュア HTTPS ポート (SSL)。	server.properties ファイルを変更します。インストール時に変更することもできます。
54046	TCP	nmsas.server.port.naming.port	デフォルトのブートストラップ JNP サービスポート (JNDI プロバイダー)	server.properties ファイルを変更します。インストール時に変更することもできます。
54047	TCP	nmsas.server.port.naming.rmi	RMI ネームサービスのデフォルトポート	server.properties ファイルを変更します。

表 51 NNM iSPI Performance for QA 管理サーバー ( 続き ) で使用されるポート

ポート	タイプ	名前	目的	設定の変更
54084	TCP	nmsas.server.port.jmx.jrmp	デフォルトの RMI オブジェクトポート (JRMP 呼び出し元)	server.properties ファイルを変更します。
54085	TCP	nmsas.server.port.jmx.rmi	デフォルトの RMI プール済み呼び出し元ポート	server.properties ファイルを変更します。
54086	TCP	nmsas.server.port.invoker.unified	デフォルトの RMI リモートサーバーコネクタポート	server.properties ファイルを変更します。
54087	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。	server.properties ファイルを変更します。
54088	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。	server.properties ファイルを変更します。
54089	TCP	nmsas.server.port.remoting.ej3	デフォルトの EJB3 リモートコネクタポート	server.properties ファイルを変更します。
54712	TCP	nmsas.server.port.ts.recovery	トランザクションサービスで使用するデフォルトの復旧ポート。	server.properties ファイルを変更します。
54713	TCP	nmsas.server.port.ts.status	トランザクションサービスで使用するデフォルトのステータスポート。	server.properties ファイルを変更します。
54714	TCP	nmsas.server.port.ts.id	トランザクションサービスで使用するデフォルトポート。	server.properties ファイルを変更します。

表 52 に、NNM iSPI Performance for Metrics および Network Performance Server (NPS) で必要となるポートを示します。ポートが競合する場合、これらのポート番号のほぼすべてを変更できます。

表 52 NNM iSPI Performance for Metrics および NPS で必要となるポート

ポート	タイプ	名前	目的	設定の変更
9300	TCP	NPS UI	Web UI と BI Web サービスで使用される、デフォルトの HTTP ポート。	'configureWebAccess.ovpl' を使用して変更します。
9305	TCP	NPS UI - SSL	Web UI と BI Web サービスで使用される、デフォルトのセキュア HTTPS ポート (SSL)。	'configureWebAccess.ovpl' を使用して変更します。

**注意:** NNM と NPS が共存していない場合、OS のネットワークファイル共有で使用されるネットワークポートも必要になります (Linux では NFS サービス、Windows では Windows ファイル共有)。

同じサーバーで実行されているプロセスで使用する (ネットワーク上のサーバー間の通信で使用されない) ポート

表 52 NNM iSPI Performance for Metrics および NPS で必要となるポート (続き)

ポート	タイプ	名前	目的	設定の変更
9301	TCP	Sybase ASE	Sybase ASE (BI コンテンツマネージャーのデータベース)	変更できません。
9302	TCP	Sybase IQ Agent	Sybase IQ Agent サービス	変更できません。
9303	TCP	Sybase IQ - PerfSPI DB	すべてのNPS拡張パックのデータを保存するために使用する Sybase IQ データベース。	変更できません。
9304	TCP	Sybase IQ - PerfSPI <b>DEMO DB</b>	拡張パックの <b>DEMO</b> データを保存するために使用する Sybase IQ データベース。	変更できません。
9306	TCP	データベースの SQL 再書き込みプロキシ - PerfSPI DB	BI サーバーによって使用される、Perfspi データベースの SQL 再書き込みプロキシ。	変更できません。
9307	TCP	データベースの SQL 再書き込みプロキシ - PerfSPI <b>DEMO DB</b>	BI サーバーによって使用される、Perfspi <b>DEMO</b> データベースの SQL 再書き込みプロキシ。	変更できません。
9308	TCP	Sybase ASE バックアップサーバー	BI コンテンツマネージャーのデータベースの Sybase ASE バックアップサーバー。	変更できません。

表 53 に、NNM iSPI NET 診断サーバーが使用するポートを示します。NNM iSPI NET 診断サーバーによって HP Operations Orchestration (HP OO) がインストールされます。詳細については、『HP Operations Orchestration 管理者ガイド』(HP Operations Orchestration Administrator's Guide) を参照してください。

表 53 NNM iSPI NET 診断サーバーで使用されるポート

ポート	タイプ	名前	目的	設定の変更
3306	TCP	MySQL データベースポート	MySQL データベースへのアクセスを提供します。	変更できません。
8080	TCP	jetty http ポート	Web UI と Web サービスで使用される、デフォルトの HTTP ポート。	インストール後の変更はサポートされていません。
8443	TCP	jetty SSL/https ポート	Web UI と Web サービスで使用される、デフォルトの HTTPS ポート。	インストール後の変更はサポートされていません。
9004	TCP	HP OO RAS ポート	HP OO リモートアクションサービスへのアクセスを提供します。	変更できません。

# 設定変更の提案

一般的な問題とその対処法をいくつか紹介します。

## 問題および解決策

**問題:** NNMi が、SNMP データおよび MIB 文字列を正しく解釈して表示しないことがある。

**解決方法:** これは、NNMi がどの文字セットを使用してこのデータを解釈するのかわからないことがあることが原因です。その結果、NNMi は、sysDescription、sysContact、その他のデータなど、一部の SNMP トラップからの文字化けした文字列およびその他の `octetstring` データを表示します。正しい文字セットを使用してこのデータを解釈することで解決できます。

不適切な文字セットを使用しているために、SNMP トラップおよびその他の `octetstring` データが文字化けしたテキストで表示されてしまう場合は、以下の手順を実行してください。

- a 以下のファイルを編集します。
  - Windows: `%NNM_PROPS%\nms-jboss.properties`
  - UNIX: `$NNM_PROPS/nms-jboss.properties`
- b `#!/com.hp.nnm.sourceEncoding=` で始まる行からコメント(#! 文字)を削除します。
- c `nms-jboss.properties` ファイルの例を使用し、ご使用の環境で現在サポートされているソースエンコーディングをカンマで区切ったリストに `com.hp.nnm.sourceEncoding JVM` プロパティを設定します。この例は、`Shift_JIS`、`EUC_JP`、`UTF-8`、`ISO-8859-1` の文字セットの組み合わせを示します。
- d 変更を保存します。
- e コマンドプロンプトから `ovstop` を実行します。
- f コマンドプロンプトから `ovstart` を実行します。
- g 変更内容をテストするには、疑わしいトラップを NNMi に再送信し、文字化け表示の問題が発生しないことを確認します。

バイナリデータ、または何らかの理由で解釈できないデータが文字化けテキストに含まれる場合は、以下の手順を実行し、16 進数形式で文字列を表示するように NNMi を設定します。

- a 以下のファイルを編集します。
  - Windows: `%NNMDATADIR%\$shared\nnm\conf\nnmvbnosrcenc.conf`
  - UNIX: `$NNMDATADIR/shared/nnm/conf/nnmvbnosrcenc.conf`
- b NNMi が文字化けした形式で表示するトラップ OID、`varbind` OID 値の組み合わせを追加します。バイナリデータなど、NNMi にデコードさせない `varbind` 値からの組み合わせも追加します。`nnmvbnosrcenc.conf` ファイルの例をテンプレートとして使用し、組み合わせを設定します。これは、NNMi に 16 進値を使用してインシデントフォームのカスタムインシデント属性値を表示するように指示します。
- c 変更を保存します。



- d コマンドプロンプトから **ovstop** を実行します。
- e コマンドプロンプトから **ovstart** を実行します。
- f 変更内容をテストし、この変更によって、以前文字化けしていた文字列が **16** 進数で表示されるようになったことを確認します。

### 問題: NNMi が、ホスト (NNMi 管理サーバー) と一致しないライセンスキーに関するメッセージを表示する。

**解決方法:** これは、NNMi 管理サーバーの IP アドレスと一致しない IP アドレスで作成された NNMi ライセンスキーがインストールされた後に発生します。以下の手順で無効なライセンスキーを削除することで解決できます。

- 1 コマンドプロンプトで、以下のコマンドを開き、Autopass ユーザーインターフェースを開きます。

```
nnmlicense.ovpl NNM -gui
```

- 2 [Autopass] ウィンドウの左側で [ **ライセンスキーの削除** ] をクリックします。
- 3 無効なライセンスキーを選択します。
- 4 [ **削除** ] をクリックします。

NNM を影響される製品で置き換えて、その他の影響される NNMi 製品統合に手順 1 から手順 4 を繰り返します。たとえば NNM iSPI ネットワークエンジニアリングツールセットソフトウェアに関連するライセンスを操作するには、以下のコマンドを使用して Autopass ユーザーインターフェースを開きます。

```
nnmlicense.ovpl iSPI-NET -gui
```

ライセンスの詳細については、「[NNMi のライセンス](#)」(123 ページ) を参照してください。

### 問題: ESXi サーバー、および ESXi サーバーで動作する仮想マシンと仮想サーバーが NNMi マップに表示される。NNMi では、すべてのシステムが雲のシンボルで接続されて表示されます。これは、仮想マシンと仮想サーバーを含む ESXi サーバーを NNMi マップに表示しない場合に限り問題となります。

**解決方法:** 仮想マシンと仮想サーバーを含む ESXi サーバーが NNMi に表示されないようにするには、以下の手順を実行します。

- 1 NNMi コンソールを開きます。
- 2 削除するノードを表示しているトポロジマップに移動し、**ESXi** サーバー、仮想マシン、および仮想サーバーを表すノードを削除します。
- 3 [ **設定** ] ワークスペースの [ **検出の設定** ] をクリックします。
- 4 [ **自動検出ルール** ] タブをクリックします。
- 5 新しい自動検出ルールを作成します。
- 6 比較的小さい数値を [ **順序** ] フィールドに入力し、このルールの優先順位を高くします。[ **含められたノードを検出する** ] チェックボックスがオフになっていることを確認します。
- 7 このルールの新しい IP アドレス範囲を追加します。
- 8 **ESXi** サーバー、仮想マシン、および仮想サーバーを表すノードの場合は、このノードのそれぞれの IP アドレスまたは IP アドレス範囲を追加し、[ **範囲のタイプ** ] を [ **ルールにより無視された** ] ではなく [ **ルールにより含める** ] に変更します。



- 9 [保存して閉じる] を 3 回クリックして作業を保存します。



この手順によって既存ノードが削除されるのではなく、除外された IP アドレス範囲内のノードが今後検出されなくなります。

**問題: ESXi サーバーとノードではなく、Linux サーバーが NNMi マップに表示される。**

**解決方法:** Net-SNMP エージェントが有効になっている Linux サーバーで VMWARE が導入されました。NNMi によって ESXi サーバーを検出して表示する場合は、ESXi サーバーとノードのベアーメタルインストールを完了する必要があります。詳細については、<http://www.vmware.com> を参照してください。

**問題: ESXi デバイスではなく、[SNMP なし] が NNMi マップに表示される。**

**解決方法:** NNMi が ESXi サーバーとノードを検出してマッピングするためには、ESXi SNMP エージェントをインストールして有効にする必要があります。ESXi SNMP エージェントをアンインストールしたか無効にした可能性があります。これを解決するには、ESXi SNMP エージェントをインストールするか有効にします。詳細については、<http://www.vmware.com> を参照してください。

**問題: NNMi で Oracle データベースを使用している。大きいノードグループを設定すると、ノードグループマップの生成中にエラーが発生する。**

**解決方法:** これは、NNMi を以下のように設定した場合に生じる可能性があります。

- NNMi で Oracle データベースを使用している。
- 子ノードグループを含む最上位レベルのノードグループを作成している。
- いずれかの子ノードグループに 1000 以上のメンバーが含まれている。
- これらのノードグループの [ノードグループマップの設定] -> [接続] -> [ノードグループ接続] セクションで、以下のいずれか、あるいは両方を選択している。

— ノードからノードグループへ

— ノードグループからノードグループへ

これを修正するには、子ノードグループに含まれるメンバーを 1000 未満にするか、これらのノードグループの [ノードグループマップの設定] -> [接続] -> [ノードグループ接続] セクションで、[ノードからノードグループへ] または [ノードグループからノードグループへ] のいずれか、あるいは両方を選択しないようにします。

**問題: PAgP (ポート集約プロトコル) を使用している一部の Cisco デバイスの場合、ポート集約の一部となっているリンクが停止すると、NNMi でそのデバイスのポートがポート集約の一部ではなくなったとみなされる可能性がある。これにより、ポート集約のパフォーマンス低下状態が NNMi からレポートされなくなる場合がある。**

**解決方法:** NNMi 9.0x パッチ 4 から、PAgP を使用する Cisco デバイスを NNMi で管理しやすくする機能が備わっています。この NNMi の機能を設定して、停止中のインタフェースがポート集約の一部としてまだ設定されているかどうかを判断できます。この機能を有効にするには、以下の手順を実行します。

- 1 以下のファイルを編集します。
  - Windows: %NNM\_PROPS%\nms-disco.properties
  - UNIX: \$NNM\_PROPS/nms-disco.properties

- 2 enablePagpOperDownHeuristic エントリーを特定します。このエントリーは以下の行のように記述されています。

```
#!com.hp.ov.nms.disco.enablePagpOperDownHeuristic=false
```

enablePagpOperDownHeuristic を有効にするには、以下のように行を変更します。

```
com.hp.ov.nms.disco.enablePagpOperDownHeuristic=true
```



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します。
- 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。

### 問題: Internet Explorer 8 および Internet Explorer ESC (セキュリティ強化の構成) を使用している場合、ポップアップダイアログの問題が発生する。

**解決方法:** Windows 2003 および Windows 2008 サーバーでは、Internet Explorer 8 の Internet Explorer ESC (セキュリティ強化の構成) という機能を提供しています。この機能を有効にすると (現在、この機能はデフォルトで有効になっています)、すべてのポップアップダイアログおよびウィンドウが信頼済みサイトのリストに対してテストされます。ポップアップに関連付けられている URL が信頼済みサイトのリストにない場合、ダイアログまたはウィンドウのすべてのコントロールが無効になります。たとえば、これが発生すると、[OK]、[適用]、および [キャンセル] ボタンをクリックしても何も反応しなくなります。

通常、ESC が有効になっていると、ダイアログを開いたときに、ポップアップに関連付けられている URL を信頼済みサイトとして有効にするかどうかを尋ねるプロンプトがブラウザに表示されます。続行するには、URL を許可します。URL を許可しないと、ダイアログのコントロールが機能しなくなり、ダイアログまたはウィンドウを開いたときにプロンプトが表示されるようになります。すべての重要な URL が信頼済みサイトのリストに登録されるため、このプロンプトは徐々に表示されなくなります。このリストに配置する必要のある特別な URL の 1 つとして about:blank が挙げられます。

NNMi コンソールが機能しなくなる状況に陥る可能性があります。ある時点で、[信頼済みサイト] メッセージの [今後、このメッセージを表示しない] チェックボックスをオンにすると、後続のプロンプトが発行されなくなります。NNMi をインストールする前にこれを行うと、NNMi コンソールがほとんど機能しなくなります。ダイアログがポップアップされなかったり、ダイアログのコントロールが機能しなくなったりします。たとえば、[ヘルプ]->[バージョン情報] ダイアログを開いた場合、[OK] ボタンをクリックしてもダイアログが閉じません。また、すべてのテーブルビューのフィルターダイアログが機能しません。後者の場合、about:blank URL が信頼済みのリストにないことが原因です。

この問題を解決する方法はいくつかあります。

- サーバーマネージャーを使用して [Internet Explorer セキュリティ強化の構成] 機能を無効にする。
- IE -> [ツール]-> [オプション]-> [セキュリティ] タブを使用して必要な URL (特に about:blank) を信頼済みセキュリティサイトに追加する。
- IE のポップアップウィンドウで信頼済みセキュリティサイトへの追加を許可する。

## 問題 : Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files ライブラリを誤って NNMi 管理サーバーから削除してしまった。

NNMi コンソールの [SNMPv3 設定] フォームでは、SNMPv3 デバイスとのやり取りに使用するプライバシープロトコルを指定できます。Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files ライブラリが NNMi 管理サーバーにインストールされている場合に限り、AES-192、AES-256、TripleDES のプロトコルを使用できます。

Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files ライブラリを誤って削除してしまい、SNMPv3 通信に使用する AES-192、AES-256、および TripleDES のプライバシープロトコルを NNMi で使用できるようにする必要がある場合、以下の手順を実行します。

- 1 Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files ライブラリを、Java 開発者用の Oracle Technology Network Web サイト (<http://www.oracle.com/technetwork/java/index.html>) からダウンロードします。直接リンクは以下のとおりです。  
**[https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS\\_Developer-Site/en\\_US/-/USD/ViewProductDetail-Start?ProductRef=jce\\_policy-6-oth-JPR@CDS-CDS\\_Developer](https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_Developer-Site/en_US/-/USD/ViewProductDetail-Start?ProductRef=jce_policy-6-oth-JPR@CDS-CDS_Developer)**
- 2 ダウンロードしたパッケージを展開してから、両方の JAR ファイル (local\_policy.jar および US\_export\_policy.jar) を以下の場所にコピーします。
  - Windows: %NnmInstallDir%\nonOV\jdk\nnm\jre\lib\security
  - UNIX: \$NnmInstallDir/nonOV/jdk/nnm/jre/lib/security
- 3 以下のコマンドを実行して NNMi を再開します。
  - a **ovstop**
  - b **ovstart**



# 用語集

## ア

### 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。では、NNMi は、検出されたノードで ARP キャッシュエントリ (ならびに他のテクニック) 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。

### Causal Engine

因果関係ベースの方法を使って、ルール (RCA) をネットワーク現象に適用する NNMi テクノロジー。Causal Engine RCA をトリガーするのは、SNMP トラップ、特定のインターネット制御メッセージプロトコルの結果として検出された変更など、特定のオカレンスです。Causal Engine は RCA を使って、管理対象オブジェクトのシステムアカウントを調べ、これらオブジェクトに関するケを明確化し、ルールを生成します。

### iSPI

NNM iSPI を参照してください。

### RCA

ルールを参照してください。

### 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。

### 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ーノード

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)。

## イ

### 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オカレンスの通知は、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ーできるようにするいくつかの [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 6.x/7.x 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。

### インタフェース

ノードをネットワークに接続するのに使われる物理ポート。

## インタフェースグループ

NNMi の主要なフィルターテクニックの 1 つ。ただし、グループごとに、グループまたはフィルター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、モニタリングの設定、テーブルビューのフィルター、マップビューのカスタマイズのいずれか、またはすべてに使用できます。「[ノードグループ](#)」も参照。

## 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、用の SNMP クエリーとともに NNMi が使います。

## エ

### HA

[タ](#)を参照してください。

### HA リソースグループ

HP ServiceGuard、Veritas Cluster Server、Microsoft Cluster Service などの最新の[タ](#)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わされます。リソースは HA リソースグループで構成されます。これはクラスター環境で実行中のアプリケーションを表します。

### エピソード

NNMi [ルール](#)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トリガーされ、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付けられます。

## HP Network Node Manager i Software

ネットワーク管理の支援や統合のために設計された HP のソフトウェア商品です (短縮形は NNMi)。ネットワークノードの継続検出、イベントの監視、ネットワーク障害管理といった機能を備えています。主に [NNMi コンソール](#)からアクセスします。

### 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。

## NNM 6.x/7.x 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。

## NNM iSPI

[タ](#)ファミリ内のスマートプラグイン。NNM iSPI は、MPLS のような特殊テクノロジー用に、またはネットワークエンジニアリングのような特定の分野用に、NNMi に機能を追加します。

## NNMi

[HP Network Node Manager i Software](#) を参照してください。

## NNMi コンソール

NNMi ユーザーインタフェース。オペレーターや管理者は、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。

## SNMP

[スパイラル検出](#)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 エージェントから請求された応答) は、処理をできるだけ簡単にするための SNMP の設計原則です。しかし、このプロトコルは、SNMP エージェントから SNMP マネージャープロセス (この場合、NNMi) への要請されないメッセージの通信も提供します。要請されないエージェントメッセージは、「トラップ」として知られており、内部状態の変化または障害条件に回答して 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 [インターネット制御メッセージプロトコル](#)を生成します。

## SNMP トラップストーム

要請されない大量の SNMP エージェントメッセージ。SNMP マネージャープロセス (この場合、NNMi) を圧倒する可能性があります。nmtrapconfig.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。

## MIB

[ハ](#)を参照してください。

## sysObjectID

[システムオブジェクト ID](#) を参照してください。



## オ

### OID

**ovstop** コマンド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** リファレンスページまたはUNIXのマン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** リファレンスページまたはUNIXのマンページを参照してください。

#### オブジェクト識別子

SNMPで、ヘ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階層のそのレベルにおける特定のデータオブジェクトを表します。OIDはMIBオブジェクト名と同等の数字です。たとえば、MIBオブジェクト名

iso.org.dod.internet.mgmt.mib-2

bgp.bgpTraps.bgpEstablishedはそのOID

1.3.6.1.2.1.15.0.1と同等です。

#### 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** リファレンスページまたはUNIXのマンページを参照してください。

## カ

### 管理情報ベース

SNMPで、管理対照ネットワークに関するデータの階層的に組織化された集合。管理情報ベース内のデータオブジェクトは管理対照デバイスの特色を参照します。NNMiは、ネットワーク管理情報を収集する場合、MIBデータオブジェクト(「MIBオブジェクト」、「オブジェクト」、「MIB」と呼ばれることもあります)を使って、管理対象ノードとの間でSNMPクエリーを出し、またはSNMPトラップを受け取ります。

### 管理サーバー

NNMi管理サーバーは、NNMiソフトウェアがインストールされるコンピューターシステムです。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)

### 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、管理対照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、SNMPv2c、およびSNMPv3と3つをサポートしています。

### 仮想ホスト名

仮想IPアドレスと関連付けられたホスト名。

### 仮想IPアドレス

特別な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。

## ク

### クラスター

NNMiの関係では、高可用性テクノロジーまたはjbossクラスター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、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実にします。クラスター内のコンピューターは一般に高速LAN経由でお互いに接続されます。クラスターは、通常、可用性かパフォーマンス、またはその両方を向上させるために導入します。

### クラスター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クラスター内のシステム。

### 組み込みデータベース

NNMiに組み込まれたデータベース。NNMiは、ほとんどのテーブルについて、組み込みデータベースの代わりに外部のOracleデータベースを使うよう設定することもできます。「PostgreSQL」も参照。

## グローバルネットワーク管理

地理的に分散している 1 つ以上のリージョナルマネージャーからのデータを統合する 1 つ以上のグローバルマネージャーを持つ、NNMi の分散型の配備です。

## ケ

### 結論

NNMi で、管理対象オブジェクト用に **Causal Engine** がシステムアカウントとルール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。

### 検出のヒント

SNMP ARP キャッシュクエリー、CDP、EDP、またはその他の検出プロトコルクエリー、または ping スニープを使用して 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。

最初の検出の後に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。では、新しいデバイスは現在の**検出シード**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**」、「**」**、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

プロセスを制限するのに使われる、ある範囲のユーザー定義 IP アドレスかシステムオブジェクト ID (**ovstop コマンド**)、またはその両方。検出ルールは、**NNMi コンソール**の [ **自動検出ルール** ] の [ **検出の設定** ] 部分に設定します。「**」**も参照。

### 検出シード

**シード**を参照してください。

## コ

### 高可用性

このガイドでは、設定の一部に障害があっても中断されないサービスを提供するハードウェアおよびソフトウェアの設定のことで、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、市販されているいくつかの 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。

### 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。NNMi で、根本原因とは、関連付けられた問題の現象が処理されていない場合、すぐに実施できる問題です。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**ルール**の生成、あるいはその両方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 からみてルーターの反対側にある管理対象ノードが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、状態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。

### 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 **インターネット制御メッセージプロトコル**。NNMi は、関連問題の現象が処理されていない場合、**ルール (RCA)** を使って現象を解決するすぐ実施できる課題として根本原因インシデントを確定します。**ルール**を参照してください。

### コンソール

**NNMi コンソール**を参照してください。

### コントローラー

NNMi **アプリケーションフェイルオーバー**での、マスタークラスターの状態を持つクラスターメンバーを表す **JGroups** 用語。**JGroups** により、クラスターのどの



2012年5月

メンバーが最下位の IP アドレスに基づくコントローラーであるかが判別されます。

## 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、SNMP トラップ v1 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脆くなります。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。

## サ

### 自動検出

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か ICMP ping、また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。障害ポーリングはのサブセットです。

### 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、コンテンツメント、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。「検出シード」、「」、および「リストに基づいた検出」も参照してください。

### 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ケから Causal Engine が計算します。状態と比べてください。

### システムアカウント

NNMi では、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のみに使用されます。ユーザーアカウントと比べてください。

## 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 **ovstop** コマンド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ハ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場合には、HP ProCurve スイッチファミリー、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。他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 デバイスのプロファイルルの定義にあります。デバイスのプロファイルルは、ネットワーク要素の種類が分かると、削減できるネットワーク要素の特徴を指定します。

### 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。たとえば、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。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。

### シード済み検出

リストに基づいた検出を参照してください。

### 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。システムアカウントと比べてください。

### 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。「コ」も参照。

## タ

### 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。

## ナ

### 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ーシステムやデバイス（プリンター、ルーター、ブリッジなど）のことです。SNMP クエリーに回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 の主要なフィルターテクニックの 1 つ。ただし、グループごとに、グループまたはフィルター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、モニタリングの設定、テーブルビューのフィルター、マップビューのカスタマイズのいずれか、またはすべてに使用できます。「[インターネット制御メッセージプロトコル](#)」も参照。

## ハ

### 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。

### Ping スィープ

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。で有効にすると、NNMi は、設定された IP アドレス範囲で ping スィープを使用してその他のノードを検索できます。サービス拒絶攻撃に ping スィ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。

### ボリュームグループ

コンピューター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 がサポートするいくつかのタ製品は、共有ファイルシステムにおいてボリュームグループを使用します。

### ポート

ネットワークハードウェアの関係において、ネットワークデバイスを經由して情報の受け渡しを行うコネクターです。

## 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。NNMi では、ほとんどのテーブルについて PostgreSQL の代わりに Oracle を使用するよう設定することもできます。

## マ

### 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 NNMi が検出した他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のみであり、[ルーター] ノードグループのノードに含まれます。

## ヤ

### ユーザーアカウント

NNMi では、ユーザーまたはユーザーグループのために 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り、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、管理者、Web サービスクライアント、オペレーターレベル 2、オペレーターレベル 1、ゲストなど、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階層型ユーザーロールがあります。「[ユーザーアカウント](#)」も参照。

## ラ

### 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のみ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のみを保守します。と比べてください。「[検出シード](#)」と「[スパイラル検出](#)」も参照。

### リージョナルマネージャー

デバイスの検出、ポーリング、およびトラップ受信を行い、情報をグローバルマネージャーに転送する、グ

2012年5月

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。

## ルール

[検出シード](#)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[検出シード](#)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 [ [自動検出ルール](#) ] の [ [検出の設定](#) ] 部分に設定します。 [リストに基づいた検出](#)と比べてください。

### 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。

### 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内の任意のサイズの容量を指すコンピューターストレージ仮想化の用語。NNMi がサポートするいくつかのタ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。

### 領域

NNMi において、タイムアウト値やアクセス資格認定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。

## ろ

### ロール

[ユーザーロール](#)を参照してください。



# フィードバックをお待ちしております。

ご使用のシステムに電子メールクライアントが設定されている場合は、デフォルトで、ここをクリックすると電子メールウィンドウが開きます。

使用可能な電子メールクライアントがない場合は、Web メールクライアントの新規メッセージに以下の情報をコピーして、**ovdoc-nsm@hp.com** にこのメッセージを送信してください。

**製品名およびバージョン:** NNMi 9.20

**ドキュメントタイトル:** NNMi デプロイメントリファレンス

**フィードバック:**

