Cisco Virtual Network Management Center XML API リファレンス ガイド リリース 1.3

2012 年 1 月 27 日

Text Part Number: OL-25847-01-J

【注意】シスコ製品をご使用になる前に、安全上の注意（www.cisco.com/jp/go/safety_warning/）をご確認ください。

本書は、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動 / 変更されている場合がありますことをご了承ください。
あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

また、契約等の記述については、弊社販売パートナー、または、弊社担当者にご確認ください。

Cisco Virtual Network Management Center XML API リファレンス ガイド リリース 1.3

CONTENTS

はじめに

ここでは、『*Cisco Virtual Network Management Center XML API リファレンス ガイド、リリース 1.3*』の対象読者、構成、および表記法について説明します。また、関連マニュアルの入手方法についても説明します。

この章の内容は、次のとおりです。

対象読者

このマニュアルは、プログラミングとアプリケーション プログラミング インターフェイス（API）の知識があるソフトウェア エンジニアを対象としています。エンジニアは、XML、データ システム、ネットワーキング プロトコル、およびストレージ プロトコルに関する知識を持っている必要があります。

マニュアルの構成

このマニュアルは、次の章で構成されています。

章およびタイトル	説明
第 1 章「Cisco VNMC XML API の概要」	Cisco VNMC XML API の概要を提供します。
第 2 章「Cisco VNMC XML API の使用」	使用例のチュートリアルおよび詳細を提供します。
第 3 章「Cisco VNMC XML API メソッドの説明」	API メソッドの説明を構文および作業例とともに提供します。

表記法

このマニュアルでは、次の表記法を使用しています。

（注） 「*注釈*」を意味します。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。

注意 「*要注意*」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

ヒント 「*問題解決に役立つ情報*」です。

コマンドの説明では、次の表記法を使用しています。

表記法	説明
太字	コマンドおよびキーワードは太字で示しています。
イタリック体	ユーザが値を指定するパラメータは、イタリック体で示しています。
[ ]	角カッコの中の要素は、省略可能です。
[x | y | z]	どれか 1 つを選択できる省略可能なキーワードは、角カッコで囲み、縦棒で区切って示しています。
string	引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。

出力例では、次の表記法を使用しています。

`screen font`	スイッチに表示される端末セッションおよび情報は、screen フォントで示しています。
`boldface screen font`	ユーザが入力しなければならない情報は、太字の screen フォントで示しています。
`italic screen font`	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
< >	パスワードのように出力されない文字は、山カッコ（< >）で囲んで示しています。
[ ]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符（!）または番号記号（#）がある場合は、コメント行であることを示します。

マニュアルの入手方法およびテクニカル サポート関連資料

ここでは、Cisco Virtual Network Management Center および関連製品に利用可能なマニュアルについて説明します。

この項は、次の内容で構成されています。

Cisco Virtual Network Management Center に関するマニュアル

以下の Cisco Virtual Network Management Center に関するマニュアルは、Cisco.com の次の URL で入手できます。

http://www.cisco.com/en/US/products/ps11213/tsd_products_support_series_home.html

- 『*Release Notes for Cisco Virtual Network Management Center, Release 1.3*』
- 『*Cisco Virtual Security Gateway, Release 4.2(1)VSG1(3.1) and Cisco Virtual Network Management Center, Release 1.3 Installation and Upgrade Guide*』
- 『*Cisco Virtual Network Management Center CLI Configuration Guide, Release 1.3*』
- 『*Cisco Virtual Network Management Center GUI Configuration Guide, Release 1.3*』
- 『*Cisco Virtual Network Management Center XML API リファレンス ガイド リリース 1.3*』

Cisco Virtual Security Gateway に関するマニュアル

以下の Cisco Virtual Security Gateway for the Nexus 1000V シリーズ スイッチに関するマニュアルは、Cisco.com の次の URL で入手できます。

http://www.cisco.com/en/US/products/ps11208/tsd_products_support_model_home.html

- 『*Cisco Virtual Security Gateway for Nexus 1000V Series Switch Release Notes, Release 4.2(1)VSG1(3.1)*』
- 『*Cisco Virtual Security Gateway, Release 4.2(1)VSG1(3.1) and Cisco Virtual Network Management Center, Release 1.3 Installation and Upgrade Guide*』
- 『*Cisco Virtual Security Gateway for Nexus 1000V Series Switch License Configuration Guide, Release 4.2(1)VSG1(3.1)*』
- 『*Cisco Virtual Security Gateway for Nexus 1000V Series Switch Configuration Guide, Release 4.2(1)VSG1(3.1)*』
- 『*Cisco Virtual Security Gateway for Nexus 1000V Series Switch Command Reference, Release 4.2(1)VSG1(3.1)*』
- 『*Cisco Virtual Security Gateway for Nexus 1000V Series Switch Troubleshooting Guide, Release 4.2(1)VSG1(3.1)*』

Cisco Nexus 1000V シリーズ スイッチのマニュアル

Cisco Nexus 1000V シリーズ スイッチ のマニュアルは、次の URL で入手できます。

http://www.cisco.com/en/US/products/ps9902/tsd_products_support_series_home.html

マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、テクニカル サポート、その他の有用な情報について、次の URL で、毎月更新される『*What's New in Cisco Product Documentation*』を参照してください。シスコの新規および改訂版の技術マニュアルの一覧も示されています。

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

『*What's New in Cisco Product Documentation*』は Really Simple Syndication（RSS）フィードとして購読できます。また、リーダー アプリケーションを使用してコンテンツがデスクトップに直接配信されるように設定することもできます。RSS フィードは無料のサービスです。シスコは現在、RSS バージョン 2.0 をサポートしています。

CHAPTER 1

Cisco VNMC XML API の概要

ここでは、Cisco VNMC XML アプリケーション プログラミング インターフェイス（API）について簡単に説明します。

この章の内容は、次のとおりです。

Cisco VNMC および XML API について

ここでは、次の内容について説明します。

Cisco VNMC の概要

Cisco Virtual Network Management Center（VNMC）は、Cisco Virtual Security Gateway（VSG）および Cisco Nexus 1000V シリーズ スイッチのデバイスおよびセキュリティ ポリシーを一元的に管理できる仮想アプライアンスです。Cisco VNMC はマルチテナント動作用に設計されており、仮想化されたデータセンターとクラウド環境にシームレスで、スケーラブルで、オートメーション中心の管理を提供します。Web ベースの GUI、CLI、および XML API メソッドによって、Cisco VNMC は、中央のロケーションからデータセンター全体に配置されている Cisco VSG を管理することができます。

マルチテナント機能とは、複数のクライアント組織またはテナントが、共有された物理インフラストラクチャに展開される、独自の仮想コンピュータ、ネットワーク、およびストレージ リソースを所有できるアーキテクチャ原則を指します。複数のテナントは同じインフラストラクチャ上で共存できます

が、各テナントは仮想リソースの管理権限を持つことができます。システムは、コンピュータ、ネットワーク、ストレージ、およびセキュリティ ポリシーを含む、各テナントに対して指定された SLA を満たすよう設計されています。

Cisco VNMC は、情報モデル方式アーキテクチャに基づいて構築され、各管理対象デバイスはそのサブコンポーネントによって表されます。このアーキテクチャにより、Cisco VNMC では、マルチテナント インフラストラクチャを素早く簡単にセキュアにすることができます。

VNMC 管理情報モデル

Cisco VNMC サービス コンポーネントを構成するすべての物理および論理コンポーネントは、階層管理情報モデルで表されます。このモデルを管理情報ツリー（MIT）といいます。このツリー内の各ノードは、管理ステータスと動作ステータスを含む、管理対象オブジェクト（またはオブジェクトのグループ）を表します。階層構造は、最上部から始まり、親ノードと子ノードを含みます。このツリー内の各ノードは管理対象オブジェクトであり、各オブジェクトは、オブジェクトとツリー内の位置を示す一意の識別名（DN）を持ちます。

管理対象オブジェクトは、ポリシー、ルール、セキュリティ プロファイル、Cisco VSG インスタンスなどの Cisco VNMC 管理対象エンティティを抽象化したものです。特定の管理対象オブジェクトは、ユーザが作成するのではなく、Cisco VNMC によって自動的に作成されます。API を呼び出すことにより、オブジェクトは MIT から読み取られ、MIT に書き込まれます。個々の Cisco VNMC サービス コンポーネントの情報モデルは、データ管理エンジン（DME）によって一元的に保存および管理されます。ユーザが Cisco VNMC サービス コンポーネントに対して管理上の変更（コンピュータ ファイアウォール プロファイルと Cisco VSG の関連付けなど）を開始すると、DME により最初にその変更が情報モデルに適用され、続いてその変更が実際の Cisco VSG に適用されます。この方法を、モデル方式フレームワークといいます。

Cisco VNMC のコンポーネント

Cisco VNMC は、管理、テナント管理、ポリシー管理、リソース管理などを行うモジュール形式の機能を提供する複数のサービス コンポーネントで構成されます。これらのコンポーネントは、サービス プロバイダーまたはアプリケーションとも呼ばれます。各コンポーネントは、一意の URL を介してアクセスされます。

次の項では、これらのコンポーネントについて説明します。

- 「Management Controller」（P.1-3）
- 「Service Registry」（P.1-3）
- 「Resource Manager」（P.1-3）
- 「Policy Manager」（P.1-4）
- 「VM Manager」（P.1-4）

各コンポーネントは、独自のデータ モデルと、モデル方式サービス要求を処理する DME（データ管理エンジン）を持ちます。各コンポーネントは、独自の管理情報ツリー ストレージ（メモリ内とパーシスタンス ストアの両方）を保持します。異なるサービス コンポーネント間でのデータ共有は、アドホックのサービス間 API 通信、パブリッシュまたはサブスクライブ メソッドによって実現されます。

Management Controller

Management Controller（コア サービスとも呼ばれます）は、Cisco VNMC 仮想マシンにシステム関連サービスを提供します。提供されるサービスは次のとおりです。

- ローカルまたは LDAP モードでユーザ ログイン認証および承認を管理します。
- ロケール、ロール、トラスト ポイントなどのアクセス コントロールを管理します。
- IP アドレス、サブネット マスク、ゲートウェイ、ホスト名などのシステム情報を管理します。
- データベース バックアップ、データ エクスポート、データ インポートなどのシステム保守を実行します。
- 監査ログ、障害、イベント ログ、コア ダンプ ファイルなどのシステム診断情報を保持します。

Management Controller のタイプは mgmt-controller です。すべての Management Controller 関連の要求に対して API URL でこのサービス タイプを使用します。

Service Registry

Cisco VNMC は分散アーキテクチャに基づいて設計されており、サービス レジストリが中央サービス レポジトリとなって、登録されたすべての管理対象エンドポイント（VSM、Cisco VSG）とサービス プロバイダー（Policy Manager や Resource Manager など）に関する情報を保持します。

サービス エンドポイントとサービス プロバイダーは、サービス レジストリに動的に登録され、サービス レジストリから該当するサービス コンポーネントに関する情報を取得します。また、サービス レジストリはテナント管理も行います。提供されるサービスは次のとおりです。

- 組織（テナント、データセンター、vApp、階層）を作成、削除、および更新します。組織の変更は Policy Manager と Resource Manager に自動的に伝播され、ポリシーとリソースが、対象となる組織に割り当てられます。
- 登録されたサービス（プロバイダー、エンドポイント、管理コントローラ、サービス レジストリ）に関する情報を保持します。
- 監査ログ、障害、イベント ログなどの診断情報を保持します。

サービス レジストリのタイプは service-reg です。すべてのサービス レジストリ関連の要求に対して API URL でこのサービス タイプを使用します。

Resource Manager

Resource Manager は、論理的なコンピュータ ファイアウォールと、実際の Cisco VSG との関連付けを管理します。コンピュータ ファイアウォールを Cisco VSG に関連付けると、デバイス設定プロファイル情報（コンピュータ ファイアウォールによって定義されます）は実際の Cisco VSG にプッシュされます。これにより、Cisco VSG が Policy Manager からセキュリティ プロファイルおよびポリシーをダウンロードします。提供されるサービスは次のとおりです。

- Cisco Virtual Security Gateways（VSG）と Virtual Supervisor Modules（VSM）のインベントリを保持します。
- コンピュータ ファイアウォールを定義し、プロビジョニングのために Cisco VSG と関連付けます。
- VMWare vCenter インスタンスと統合し、VM 属性を取得します。
- 検出された仮想マシンのインベントリを保持し、これらの仮想マシンを次の情報とともに Cisco VSG インスタンスに分散します。
  - VM 属性（名前、ハイパーバイザ、親 vApp、クラスタ）
  - vNIC 属性（ポート プロファイル名、IP アドレス）

- Cisco VSG のプールを管理します。
- VSG のヘルス状態および障害を管理します。
- 監査ログ、障害、イベント ログなどの診断情報を保持します。

Resource Manager のタイプは resource-mgr です。すべての Resource Manager 関連の要求に対して API URL でこのサービス タイプを使用します。

Policy Manager

Policy Manager は、デバイス設定プロファイル、セキュリティ ポリシー、および関連するすべてのアーティファクトの中央レポジトリです。コンピュータ ファイアウォールが Resource Manager から Cisco VSG と関連付けられた場合、Cisco VSG はデバイス プロファイル、セキュリティ プロファイル、および参照されたすべてのポリシーを解決するために Policy Manager に問い合わせます。Cisco VSG は、Policy Manager から取得された情報に基づいて設定されます。この場合のサービスは次のとおりです。

- プロファイルを定義します。
- ポリシー セットを定義し、ポリシーを割り当てます。
- 次の条件に基づいてポリシー ルールを定義します。
  - プロトコル、送信元または宛先、IP アドレス、ポートなどのネットワーク属性
  - インスタンス名、ゲスト OS、ゾーン、親アプリケーション、ポート プロファイル、クラスタ、リソース プール、ホスト名、ハイパーバイザなどの仮想マシン属性
  - カスタム属性
- ゾーンを定義します。
- オブジェクト グループを定義します。
- セキュリティ プロファイル ディクショナリとカスタム属性を定義します。
- セキュリティ プロファイルを定義し、ポリシー セットを割り当てます。
- ファイアウォール デバイス プロファイルを定義します。
- NTP、DNS、syslog、および障害に対して Cisco VNMC システム管理デバイス プロファイルおよびポリシーを定義します。
- ポリシー、セキュリティ プロファイル、デバイス プロファイル、および関連するオブジェクトを Cisco VSG インスタンスに分散します。
- 監査ログ、障害、イベント ログなどの診断情報を保持します。

Policy Manager のサービス タイプは policy-mgr です。すべての Policy Manager 関連の要求に対して API URL でこのサービス タイプを使用します。

VM Manager

VM Manager は、VMWare vCenter と対話し、vCenter から取得された VM 情報を保持するために使用します。VM Manager は、ユーザがアクセスできるサービスがないバックエンド サービスです。VM Manager のサービス タイプは vm-manager です。

Cisco VNMC XML API の概要

Cisco VNMC XML API を使用すると、Cisco Virtual Network Management Center との統合または対話をプログラミングによって実行できます。API インターフェイスは、HTTPS プロトコルを使用して XML ドキュメントを受け取ります。開発者は、任意のプログラミング言語を使用して API メソッドを含む XML ドキュメントを生成できます。設定およびステータス情報は、管理情報ツリー（XML API を介して完全に公開されます）という階層ツリー構造に格納されます。

API モデルは、再帰的に駆動され、アプリケーション開発のために主要な機能を提供します。たとえば、変更は、単一のオブジェクト、オブジェクトのサブツリー、またはオブジェクト ツリー全体に対して行うことができます。また、変更は、オブジェクトの単一の属性に行ったり、単一の API コールで Cisco VNMC 構造全体に適用したりできます。

API は寛容モードで動作します。不明な属性は、内部のデータ管理エンジン（DME）で保持されているデフォルト値（該当する場合）で置き換えられます。複数の管理対象オブジェクト（つまり、ポリシー）を設定するときにその管理対象オブジェクトのいずれかを設定できない場合、API はその動作を停止し、設定を元の状態に戻し、API プロセスを停止して障害を通知します。

API は、スケーラビリティとパフォーマンスを向上させるために非同期動作モデルを使用します。完了するのに時間がかかるプロセスはノンブロッキングです（高速な API プロセスは処理を続行できます）。プロセスは、有効な要求時に成功メッセージを受け取り、タスクが完了すると完了メッセージを受け取ります。

完全なイベント サブスクリプションがサポートされます。Cisco VNMC は、発生したイベント（管理対象オブジェクトの変更など）に関する通知をすべてのサブスクライバに送信し、ステータス変更のタイプを示します。

管理対象オブジェクト データ モデルの今後の更新では、後方互換性を維持するために既存のオブジェクト モデルに準拠します。製品アップグレード時に既存のプロパティが変更された場合、変更はアップグレード後のデータベース ロード中に処理されます。新しいプロパティにはデフォルト値が割り当てられます。

Cisco VNMC では、モデル方式アーキテクチャを使用しています。このアーキテクチャでは、変更は最初に管理対象オブジェクトの形式の論理構造に適用されます。次に、管理対象オブジェクトは、エンドポイントに変更を適用し、必要な状態（設定）のエンドポイントを実現します。

API の動作はトランザクション型で、単一のデータ モデルで終了します。Cisco VNMC は、すべてのエンドポイント（Cisco VSG、VSM）通信（ステータス更新など）を行います。ユーザはエンドポイントと直接通信できないため、開発者は分離された個々のコンポーネント設定を管理する必要がありません。

Cisco VNMC API モデルには、次のプログラミング エンティティが含まれます。

- クラス：管理情報ツリーのオブジェクトのプロパティとステータス
- メソッド：1 つまたは複数のオブジェクトに対して API が実行するアクション
- タイプ：オブジェクト ステータスに値をマッピングするオブジェクト プロパティ（つまり、policyDeviceProfile）

一般的な要求は Cisco VNMC サービス コンポーネントの DME に送信され、トランザクタ キューに FIFO の順序で配置されます。トランザクタはこのキューから要求を取得し、要求を解釈して認可チェックを実行します。要求が確認されると、トランザクタは管理情報ツリーを更新します。このプロセスは、単一のトランザクションで実行されます。

Cisco VNMC データ モデル スキーマ

すべての Cisco VNMC サービス プロバイダーに対するデータ モデル スキーマ ファイルは Cisco VNMC サーバでパッケージ化され、次の URL を使用して取得できます。

```
https://<vnmc ip address>/schema
```

スキーマ リストには、次のものが含まれます。

- core.in.xsd：Management Controller 設定 API とデータ モデル
- core.out.xsd：Management Controller クエリー API とデータ モデル
- policy-mgr.in.xsd：Policy Manager 設定 API とデータ モデル
- policy-mgr.out.xsd：Policy Manager クエリー API とデータ モデル
- resource-mgr.in.xsd：Resource Manager 設定 API とデータ モデル
- resource-mgr.out.xsd：Resource Manager クエリー API とデータ モデル
- service-reg.in.xsd：サービス レジストリ設定 API とデータ モデル
- service-reg.out.xsd：サービス レジストリ クエリー API とデータ モデル

Cisco VNMC サービスへのアクセス

Cisco VNMC サービスには、HTTPS プロトコルを介した XML API 要求を使用してアクセスできます。HTTPS 要求の URL 形式は次のとおりです。

```
https://<vnmc ip address>/xmlIM/<service type>
```

<service type> は、「Cisco VNMC のコンポーネント」の項で説明されている、該当するサービス プロバイダーのサービス タイプです。たとえば、Policy Manager に要求を送信するには、次の URL でサービス タイプ policy-mgr を使用します。

```
https://<vnmc ip address>/xmlIM/policy-mgr
```

オブジェクトの命名

特定のオブジェクトは、識別名（DN）または相対名（RN）で識別できます。

ここでは、次の内容について説明します。

- 「識別名」（P.1-6）
- 「相対名」（P.1-7）

識別名

DN を使用すると、ターゲット オブジェクトを明確に識別できます。DN は、一連の相対名から構成される次の形式を持ちます。

```
dn = {rn}/{rn}/{rn}/{rn}...
```

次に例を示します。

```
org-root/org-Cisco/zone-trustedServers-0
```

例では、DN は、オブジェクト ツリーの最上部から Zone オブジェクト（zone-trustedServers-0）までの完全修飾パスを提供しています。DN は、API コールが動作する管理対象オブジェクトを指定します。

```
< ... dn = "org-root/org-Cisco/zone-trustedServers-0" />
```

相対名

相対名（RN）は、親オブジェクトのコンテキスト内でオブジェクトを識別します。オブジェクトの DN は、次の形式で親 DN と RN から構成されます。

```
dn = <parent dn>/{rn}
```

たとえば、テナント Cisco に名前 trustedServers-0 が定義されている Zone オブジェクトの場合、その RN は zone-trustedServers-0 です。親テナント DN は org-root/org-Cisco であり、DN は org-root/org-Cisco/zone-trustedServers-0 です。

API メソッド カテゴリ

Cisco VNMC との対話に使用されるメソッドは、4 つのカテゴリに分けられます。各 API はメソッドであり、各メソッドは XML ドキュメントに対応します。これらのメソッドについては、次の項で説明します。

- 「認証方式」（P.1-7）
- 「クエリー メソッド」（P.1-8）
- 「設定メソッド」（P.1-10）
- 「イベント サブスクリプション メソッド」（P.1-10）

（注）　このマニュアルのいくつかのコード例では、用語 <real_cookie> は 1217377205/85f7ff49-e4ec-42fc-9437-da77a1a2c4bf などの実際の Cookie に置き換えられます。Cisco VNMC の Cookie は 47 文字の文字列です。これは、Web ブラウザがセッション情報を保持するためにローカルに保存する種類の Cookie ではありません。

認証方式

認証方式は、セッションを認証して保持します。次のような認証方式があります。

- aaaLogin：この方式は、Cisco VNMC にログインする最初の方式です。
- aaaRefresh：この方式は、現在の認証 Cookie を更新します。
- aaaLogout：この方式は現在のセッションを終了し、現在の認証 Cookie を非アクティブ化します。

認証方式は、アクティブな Cisco VNMC セッションを開始し、保持します。他の API コールが許可される前に、正常な認証を実行する必要があります。API 要求は Cookie により認証されます。

接続セッションが確立および認証されると、応答で Cookie が返されます。これは 7200 秒（120 分）有効です。Cookie は、有効期限が切れないようセッション期間中に更新する必要があります。各更新操作により、デフォルトの期間の間有効な Cookie が作成されます。

接続セッションを確立し、有効な Cookie を取得する場合は、aaaLogin を使用します。セッションを保持し、Cookie をアクティブに保つ場合は、aaaRefresh を使用します。セッションを終了（また、Cookie を無効化）する場合は、aaaLogout を使用します。Cisco VNMC に対して一度に最大 256 個のセッションを開くことができます。最大セッション制限に到達すると、後続するログイン要求が拒否されます。

操作は HTTP の post メソッドを使用して実行されます。Cisco VNMC は、ポート 443 で HTTPS プロトコルだけをサポートします。HTTP のエンベロープには XML の設定が含まれます。

クエリー メソッド

クエリー メソッドは、Cisco VNMC オブジェクトの現在の設定ステータスに関する情報を取得します。次にクエリー例を示します。

- configResolveDn：DN によりオブジェクトを取得します。
- configResolveDns：DN セットによりオブジェクトを取得します。
- configResolveClass：該当するクラスのオブジェクトを取得します。
- configResolveClasses：複数のクラスのオブジェクトを取得します。
- configFindDnsByClassId：指定されたクラスの DN を取得します。
- configResolveChildren：オブジェクトの子オブジェクトを取得します。
- configResolveParent：オブジェクトの親オブジェクトを取得します。
- configScope：管理情報ツリーの DN に対してクラス クエリーを実行します。

ほとんどのクエリー メソッドは、引数 inHierarchical、ブール値 true/yes または false/no を持ちます。true の場合、inHierarchical 引数はすべての子オブジェクトを返します。次に例を示します。

```
<configResolveDn … inHierarchical="false"></>
<configResolveDn … inHierarchical="true"></>
```

また、API クエリー メソッドは、コールを再帰的にするかどうか（つまり、他のオブジェクトまたは親オブジェクトを参照し返すオブジェクトに従うかどうか）を指定する inRecursive 引数を持つ場合があります。

クエリー フィルタ

この API は、クエリー メソッドの有用性を高めるためにフィルタ セットを提供します。これらのフィルタは、クエリーの一部として渡すことができ、必要な結果セットを特定するために使用されます。フィルタは次のように分類されます。

- Simple フィルタ
- Property フィルタ
- Composite フィルタ
- Modifier フィルタ

Simple フィルタ

Simple フィルタには true と false の 2 つがあります。これら 2 つのフィルタは、それぞれ true または false の単純なステータスに反応します。

- True フィルタ：オブジェクトの結果セットが true のブール条件を持ちます。
- False フィルタ：オブジェクトの結果セットが false のブール条件を持ちます。

Property フィルタ

Property フィルタは、結果セットに含める基準としてオブジェクトのプロパティの値を使用します。ほとんどの場合、Property フィルタを作成するには、比較の値とともにターゲット オブジェクトとプロパティの classId と propertyId が必要です。

- Equality フィルタ：結果セットを、特定のプロパティが、指定したプロパティ値と「等しい」オブジェクトに限定します。
- Not equal フィルタ：結果セットを、特定のプロパティが、指定したプロパティ値と「等しくない」オブジェクトに限定します。
- Greater than フィルタ：結果セットを、特定のプロパティが、指定したプロパティ値よりも「大きい」オブジェクトに限定します。
- Greater than or Equal to フィルタ：結果セットを、特定のプロパティが指定したプロパティ値「以上」であるオブジェクトに限定します。
- Less than フィルタ：結果セットを、特定のプロパティが、指定したプロパティ値「未満」であるオブジェクトに限定します。
- Less than or Equal to フィルタ：結果セットを、特定のプロパティが、指定されたプロパティ値「以下」であるオブジェクトに限定します。
- Wildcard フィルタ：結果セットを、特定のプロパティがワイルドカードを含むプロパティと一致するオブジェクトに限定します。サポートされるワイルドカードは、「%」または「*」（任意の文字シーケンス）、「?」または「-」（任意の単一の文字）です。
- Any Bits フィルタ：結果セットを、特定のプロパティが、渡されたビット セットの少なくとも 1 つを含むオブジェクトに限定します。（ビットマスク プロパティにだけ使用します）。
- All Bits フィルタ：結果セットを、特定のプロパティが、渡されたビット セットのすべてを含むオブジェクトに限定します。（ビットマスク プロパティにだけ使用します）。

Composite フィルタ

Composite フィルタは、2 つ以上のコンポーネント フィルタから構成されます。Composite フィルタを使用すると、柔軟に結果セットを作成できます。たとえば、Composite フィルタによって、含まれているフィルタの少なくとも 1 つで受け入れられたオブジェクトだけに結果セットを限定できます。

- AND フィルタ：結果セットは、各コンポーネント フィルタのフィルタリング基準を満たす必要があります。たとえば、totalMemory が 64 メガバイトを超えた動作可能なすべてのコンピュータ ブレードを取得する場合、フィルタは 1 つの Greater than フィルタと 1 つの Equality フィルタから構成されます。
- OR フィルタ：結果セットは、少なくとも 1 つのコンポーネント フィルタのフィルタリング基準を満たす必要があります。たとえば、未割り当ての assignmentState 値または未割り当ての associationState 値を持つすべてのサービス プロファイルを取得する場合、フィルタは 2 つの Equality フィルタから構成されます。
- Between フィルタ：結果セットは、最初の指定値と 2 番めの指定値の範囲内にあるオブジェクトです。たとえば、2 つの日付間で発生したすべての障害などがあります。
- XOR フィルタ：結果セットは、たった 1 つの Composite コンポーネント フィルタのフィルタリング基準を満たすオブジェクトです。

Modifier フィルタ

Modifier フィルタは、含まれているフィルタの結果を変更します。1 つの Modifer フィルタ（NOT フィルタ）だけがサポートされます。このフィルタは、含まれているフィルタの結果を逆にします。含まれている基準に一致しないオブジェクトを取得する場合は、このフィルタを使用します。

設定メソッド

管理対象オブジェクトの設定を変更するメソッドが複数あります。これらの変更は、ツリー全体、サブツリー、または個々のオブジェクトに適用できます。これらのメソッドの例は次のとおりです。

- configConfMo：単一のサブツリー（つまり、DN）に影響を与えます。
- configConfMos：複数のサブツリー（つまり、複数の DN）に影響を与えます。
- configConfMoGroup：複数のサブツリー構造または管理対象オブジェクトに対して同じ設定の変更を加えます。

ほとんどの設定メソッドは、引数 inHierarchical を使用します。子オブジェクトが XML ドキュメントに含まれ、DME が寛容モードで動作するため、設定時にこれらの値は重大な役割を担いません。

イベント サブスクリプション メソッド

ユーザまたはシステムにより開始されたアクションによって、オブジェクトが作成、変更、または削除されると、イベントが生成されます。アプリケーションは、定期的なポーリングまたはイベント サブスクリプションにより、Cisco VNMC のステータス変更情報を取得できます。ポーリングはリソースを大量に消費するため、イベント サブスクリプションが推奨される通知方法です。

イベント サブスクリプションでは、クライアント アプリケーションが Cisco VNMC からのイベント通知を受けるよう登録できます。イベントが発生すると、Cisco VNMC はサブスクライブしているクライアント アプリケーションにイベントとそのタイプを送信します。実際の変更イベントだけが送信され、影響を受けないオブジェクトの属性は送信されません。このプロセスは、システム内のすべてのオブジェクトの変更に適用されます。

Cisco VNMC イベント通知をサブスクライブするには、HTTP セッションを開き、セッションを開いたままにします。次に、以下の例で示されているように eventSubcribe 要求を HTTP セッションを介して送信します。

```
<eventSubscribe cookie="<real_cookie>"></eventSubscribe>
```

eventSubscribe 要求が Cisco VNMC により受け取られると、HTTP セッションを介した、発生するすべての新しいイベントの送信が開始されます。イベント サブスクリプションに対して有効な Cookie を取得するには、Cisco VNMC に最初にログインする必要があります。ログインしていない場合、イベント サブスクリプション要求は拒否され、エラー応答が発生します。

各イベントは一意のイベント ID を持ちます。これらのイベント ID はカウンタとして動作し、すべてのイベント通知に含まれます。イベントが生成されると、イベント ID カウンタが増加し、新しいイベントに新しいイベント ID が割り当てられます。このプロセスにより、イベントを追跡することが可能になり、イベントを見逃すことがなくなります。クライアントがイベントを見逃した場合は、loggingSyncOcns を使用して、見逃したイベントを取得します。

（注） Cisco VNMC 1.0 の場合は、イベント サブスクリプションに対して HTTP プロトコルだけがサポートされます。イベント サブスクリプション要求を送信する場合は HTTP を使用します。

Cisco VNMC GUI と Cisco VNMC サーバ間での XML 交換の取得

Cisco VNMC GUI は Web ベースのアプリケーションです。API が実際に使用されたことを知るために、GUI と Cisco VNMC サーバ間の XML 交換を取得します。Cisco VNMC は、Adobe FLEX GUI フレームワークを使用して開発されているため、Adobe フラッシュ プレーヤーのデバッグ バージョンをインストールしてください。インストールにより、ユーザのホーム ディレクトリ下のログ ファイルに格納されたログ出力が取得されます。Windows 7 では、ログ ファイルは C:¥Users¥<usename>¥AppData¥Roaming¥Macromedia¥Flash Player¥Logs¥flashlog.txt です。たとえば、Cisco VNMC XML API 使用例の項で指定されたほとんどの要求例および応答ペイロードはこのように取得されます。

成功または失敗の応答

Cisco VNMC は、どのような API 要求に対してもほとんど即時に応答します。要求を完了できない場合は、失敗が返されます。クエリーまたはログイン メソッドの場合は、実際の結果が返されます。設定メソッドの場合、成功の応答は要求が有効であることを示しますが、操作が完了したことを示しません。たとえば、DB バックアップ要求が受け入れられ、Cisco VNMC から成功の応答があった場合でも、Cisco VNMC サーバで実際のバックアップ ジョブを完了するのに長い時間がかかることがあります。

ここでは、次の内容について説明します。

成功の応答

成功の応答時、XML ドキュメントが、要求された情報または変更が行われたことを示す確認とともに返されます。

次に、DN org-root/org-tenant d3337/pol-pl を持つポリシーに対する configResolveDn 要求の例を示します。

```
<configResolveDn cookie="<real_cookie>"
    dn="org-root/org-tenant_d3337/pol-p1"
    inHierarchical="false"/>
```

この応答には次の詳細が含まれます。

```
<configResolveDn
    dn="org-root/org-tenant_d3337/pol-p1"
    cookie="<real_cookie>"
    commCookie="7/13/0/a3c"
    srcExtSys="10.193.34.70"
    destExtSys="10.193.34.70"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfig>
        <policyRuleBasedPolicy
            descr=""
```

```
            dn="org-root/org-tenant_d3337/pol-p1"
            intId="10811"
            name="p1"/>
    </outConfig>
</configResolveDn></configResolveDn>
```

失敗した要求

失敗した要求に対する応答には、errorCode と errorDescr に対する XML 属性が含まれます。次に、システムにすでに存在するポリシーを作成しようとした場合の失敗した要求の例を示します。

```
<configConfMo
    dn="org-root/org-tenant_d3337/pol-p1"
    cookie="<real_cookie>"
    commCookie="7/13/0/2038"
    srcExtSys="10.193.34.70"
    destExtSys="10.193.34.70"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes"
    errorCode="103"
    invocationResult="unidentified-fail"
    errorDescr="can't create; object already exists.">
 </configConfMo>
```

空の結果

存在しないオブジェクトに対するクエリー要求は、DME によって失敗として扱われません。オブジェクトが存在しない場合は、成功メッセージが返されますが、XML ドキュメントには、要求されたオブジェクトが見つからないことを示す空のデータ フィールド <outConfig> </outConfig> が含まれます。

次に、DN による存在しないポリシーの解決の例を示します。

```
<configResolveDn
    dn="org-root/org-tenant_d3337/pol-p1"
    cookie="<real_cookie>"
    commCookie="7/13/0/203e"
    srcExtSys="10.193.34.70"
    destExtSys="10.193.34.70"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfig>
    </outConfig>
</configResolveDn>
```

CHAPTER 2

Cisco VNMC XML API の使用

この章の内容は、次のとおりです。

方式およびフィルタ

ここでは、次の内容について説明します。

認証方式

認証により、API は Cisco VNMC と対話できるようになります。また、認証を使用すると、権限を設定し、実行できる操作を制御できます。

（注） セッション Cookie は 47 文字の文字列であり、Web ブラウザがセッション情報を保持するためにローカルに保存する種類の Cookie ではありません。ほとんどのコード例では、<real_cookie> が 1217377205/85f7ff49-e4ec-42fc-9437-da77a1a2c4bf などの実際の Cookie の代わりに使用されています。

ここでは、次の内容について説明します。

ログイン

例 2-1 は、Cisco VNMC に接続する HTTPS を使用して認証要求を送信する Linux の POST コマンドを使用しています。

例 2-1　　ログイン要求

```
POST https://10.193.34.70/xmlIM/mgmt-controller
Please enter content (application/x-www-form-urlencoded) to be POSTed:
<aaaLogin
    inName="admin"
    inPassword="cisco@123"/>
```

（注）　XML バージョンと DOCTYPES は、aaaLogin に含まれていません。これらを使用しないでください。inName 属性と inPassword 属性はパラメータです。

各 XML API ドキュメントは、実行する操作を表します。要求が XML API ドキュメントとして受け取られると、Cisco VNMC は要求を読み取り、メソッドで指定されているアクションを実行します。Cisco VNMC は、XML ドキュメント形式のメッセージで応答し、要求の成否を示します。

例 2-2 に、一般的な成功の応答を示します。

例 2-2　　ログイン要求応答

```
<aaaLogin
    response="yes"
    outCookie="<real_cookie>"
    outRefreshPeriod="600"
    outPriv="admin,read-only"
    outDomains="mgmt02-dummy"
    outChannel="fullssl"
    outEvtChannel="fullssl">
</aaaLogin>
```

（注）　VNMC 1.0 イベント チャネルは HTTP を使用しています。そのため、暗号化されず、SSL を介して送信されません。

セッションの更新

セッションは、aaaLogin 応答または以前の更新から取得された 47 文字の Cookie を使用して aaaRefresh メソッドで更新されます。

例 2-3 は、セッションを更新するメソッドです。

例 2-3　　セッションの更新

```
<aaaRefresh
    inName="admin"
    inPassword="mypasword"
    inCookie="<real_cookie>"/>
```

セッションからのログアウト

例 2-4 は、セッションからログアウトするメソッドです。

例 2-4　　セッションからのログアウト

```
<aaaLogout
    inCookie="<real_cookie>"/>
```

失敗したログインに対する応答

例 2-5 は、失敗したログインに対する応答です。

例 2-5　　失敗したログインに対する応答

```
<aaaLogin
    response="yes"
    cookie=""
    errorCode="551"
    invocationResult="unidentified-fail"
    errorDescr="Authentication failed">
</aaaLogin>
```

情報収集用クエリー メソッド

クエリー メソッドは、階層、ステータス、およびスコープを含む情報を取得します。

ここでは、次の内容について説明します。

- 「configResolveDn の使用」（P.2-3）
- 「configResolveDns の使用」（P.2-3）
- 「configResolveClass の使用」（P.2-4）
- 「configResolveClasses の使用」（P.2-4）
- 「configFindDnsByClassId の使用」（P.2-4）
- 「configResolveChildren の使用」（P.2-5）
- 「configResolveParent の使用」（P.2-5）
- 「configScope の使用」（P.2-5）

configResolveDn の使用

DN の解決時には、次のことを確認してください。

- DN により指定されたオブジェクトが取得されている。
- 指定された DN が、解決するオブジェクト インスタンスを識別している（この DN は必須です）。
- 認証 Cookie を aaaLogin または aaaRefresh から受け取っている。
- inHierarchical 属性（デフォルト値は false）が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、クラス ID、およびビット マスクが文字列として表示されている。

「configResolveDn」（P.3-22）に示す要求または応答の例を参照してください。

configResolveDns の使用

複数の DN の解決時には、次のことを確認してください。

- DN により指定されたオブジェクトが取得されている。
- 指定された DN が、解決するオブジェクト インスタンスを識別している。
- 認証 Cookie を aaaLogin または aaaRefresh から受け取っている。
- inHierarchical 属性（デフォルト値は false）が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、クラス ID、およびビット マスクが文字列として表示されている。
- 要求の順序によって応答の順序が決まるわけではない。
- 未知の DN が、outUnresolved 属性の一部として返されている。

「configResolveDns」（P.3-24）に示す要求または応答の例を参照してください。

configResolveClass の使用

クラスの解決時には、次のことを確認してください。

- 指定されたクラス タイプのオブジェクトが取得されている。
- classId 属性により、返されるオブジェクト クラス名が指定されている。
- 認証 Cookie を aaaLogin または aaaRefresh から受け取っている。
- inHierarchical 属性（デフォルト値は false）が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、クラス ID、およびビット マスクが文字列として表示されている。

結果セットは大きくなることがあります。結果セットは正確に定義してください。たとえば、サーバのリストだけを取得する場合は、クエリーで classId の属性値として computeBlade を使用します。装置のすべてのインスタンスを取得するには、equipmentItem クラスを問い合わせます。

「configResolveClass」（P.3-19）に示す要求または応答の例を参照してください。

configResolveClasses の使用

複数のクラスの解決時には、次のことを確認してください。

- 指定されたクラス タイプのオブジェクトが取得されている。
- classId 属性により、返されるオブジェクト クラス名が指定されている。
- 認証 Cookie を aaaLogin または aaaRefresh から受け取っている。
- inHierarchical 属性（デフォルト値は false）が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、クラス ID、およびビット マスクが文字列として表示されている。

（注） 無効なクラス名が inId 属性で指定された場合は、XML 解析エラーが生成されます。クエリーは実行できません。

「configResolveClasses」（P.3-21）に示す要求または応答の例を参照してください。

configFindDnsByClassId の使用

指定されたクラスの識別名の検出時に、次のことを確認してください。

- 指定されたクラスの DN が取得されている。
- classId 属性により、取得するオブジェクト タイプが指定されている。
- 認証 Cookie を aaaLogin または aaaRefresh から受け取っている。
- inHierarchical 属性（デフォルト値は false）が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、クラス ID、およびビット マスクが文字列として表示されている。

「configFindDnsByClassId」（P.3-16）に示す要求または応答の例を参照してください。

configResolveChildren の使用

管理情報ツリーの子オブジェクトの解決時に、次のことを確認してください。

- このメソッドが、名前付きクラスのインスタンスである名前付きオブジェクトのすべての子オブジェクトを取得している。

（注）　クラス名を省略すると、名前付きオブジェクトのすべての子オブジェクトが返されます。

- inDn 属性により、子オブジェクトが取得される名前付きオブジェクトが指定されている。
- classId 属性により、返される子オブジェクト クラスの名前が指定されている。
- 認証 Cookie を aaaLogin または aaaRefresh から受け取っている。
- inHierarchical 属性（デフォルト値は false）が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、クラス ID、およびビット マスクが文字列として表示されている。

「configResolveChildren」（P.3-18）に示す要求または応答の例を参照してください。

configResolveParent の使用

オブジェクトの親オブジェクトの解決時に、次のことを確認してください。

- このメソッドが、指定された DN の親オブジェクトを取得している。
- dn 属性が子オブジェクトの DN である。
- 認証 Cookie を aaaLogin または aaaRefresh から受け取っている。
- inHierarchical 属性（デフォルト値は false）が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、クラス ID、およびビット マスクが文字列として表示されている。

「configResolveParent」（P.3-26）に示す要求または応答の例を参照してください。

configScope の使用

クエリーのスコープを制限すると、粒度が細かくリソースを大量に消費しない要求を実現できます。クエリーは、管理情報ツリーでルート以外の場所に固定できます。

クエリー スコープの設定時に、次のことを確認してください。

- メソッドが、指定された DN にクエリーのルート（スコープ）を設定し、指定されたクラス タイプのオブジェクトを返している。

- dn が、クエリーのスコープが設定された名前付きオブジェクトである。
- inClass 属性により、返されるオブジェクト クラスの名前が指定されている。

（注）　クラスが指定されない場合、クエリーは configResolveDn と同じように動作します。

- 認証 Cookie を aaaLogin または aaaRefresh から受け取っている。
- inHierarchical 属性（デフォルト値は false）が true の場合は、結果が階層形式であることが指定されている。
- 列挙値、クラス ID、およびビット マスクが文字列として表示されている。

「configScope」（P.3-28）に示す要求または応答の例を参照してください。

ポリシーに関するクエリー メソッド

ポリシーは、さまざまな組織で利用できます。多くのポリシーがある大規模なシステムでは、一度にすべてのポリシーを問い合わせるとリソースが大量に消費されます。代わりに、ポリシーのタイプと、ポリシーが割り当てられる親組織を指定してください。

例 2-6 に、ルールベースのポリシーに関するクエリーを示します。

例 2-6　ルールベースのポリシーに関するクエリー

```
<configScope
    cookie="<real_cookie>"
    inHierarchical="false"
    dn="org-root/org-tenant_d3338"
    inClass="policyRuleBasedPolicy" />
```

応答は次のとおりです。

```
<configScope
    dn="org-root/org-tenant_d3338"
    cookie="<real_cookie>"
    commCookie="7/13/0/24e7"
    srcExtSys="10.193.34.70"
    destExtSys="10.193.34.70"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <policyRuleBasedPolicy
            descr=""
            dn="org-root/org-tenant_d3338/pol-p9"
            intId="10274"
            name="p9"/>
        <policyRuleBasedPolicy
            descr=""
            dn="org-root/org-tenant_d3338/pol-p1"
            intId="10301"
            name="p1"/>
    </outConfigs>
</configScope>
```

DN を使用する次のクエリーはさらに効率的です。

```
<configResolveDn
    inHierarchical="false"
```

```
    cookie="<real_cookie>"
    dn="sys org-root/org-tenant_d3338/pol-p1">
</configResolveDn>
```

ポリシー オブジェクトとそのルール データを取得するには、inHierarchical を true に変更します。

```
<configResolveDn
    inHierarchical="true"
    cookie="<real_cookie>"
    dn=" org-root/org-tenant_d3338/pol-p1">
</configResolveDn>
```

障害に関するクエリー メソッド

例 2-7 に、障害に関するクエリーを示します。

例 2-7　　障害に関するクエリー

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="faultInst"/>
```

次の例には、フィルタ <inFilter> eq class= "faultInst" が含まれていて、すべての主要または重大な障害のリストを取得します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="faultInst">
    <inFilter>
        <eq class="faultInst"
            property="highestSeverity"
            value="major" />
    </inFilter>
</configResolveClass>
```

フィルタ

目的の特定のクエリーを作成するには、フィルタを使用します。

ここでは、次の内容について説明します。

- 「Simple フィルタ」（P.2-7）
- 「Property フィルタ」（P.2-8）
- 「Composite フィルタ」（P.2-10）
- 「Modifier フィルタ」（P.2-12）

Simple フィルタ

Simple フィルタは true 条件と false 条件を使用します。

ここでは、次の内容について説明します。

- 「false 条件」（P.2-8）
- 「true 条件」（P.2-8）

false 条件

次に、false 条件を使用した Simple フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    classId="topSystem"
    inHierarchical="false">
    <inFilter>
    </inFilter>
</configResolveClass>
```

true 条件

次に、true 条件を使用した Simple フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    classId="topSystem"
    inHierarchical="true">
    <inFilter>
    </inFilter>
</configResolveClass>
```

Property フィルタ

ここでは、次の内容について説明します。

Equality フィルタ

次に、Equality フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="fwComputeFirewall">
    <inFilter>
        <eq class="fwComputeFirewall"
            property="assocState"
            value="associated" />
    </inFilter>
</configResolveClass>
```

Not equal フィルタ

次に、Not equal フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="fwComputeFirewall">
    <inFilter>
        <ne class="fwComputeFirewall"
            property="assocState"
            value="associated" />
    </inFilter>
</configResolveClass>
```

Greater than フィルタ

次に、Greater than フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="memoryArray">
    <inFilter>
        <gt class="memoryArray"
            property="currCapacity"
            value="1024" />
    </inFilter>
</configResolveClass>
```

Greater than or Equal to フィルタ

次に、Greater than or Equal to フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="sysdebugCore">
    <inFilter>
        <ge class="sysdebugCore"
            property="size"
            value="2097152" />
    </inFilter>
</configResolveClass>
```

Less than フィルタ

次に、Less than フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="sysdebugCore">
    <inFilter>
        <lt class="sysdebugCore"
            property="size"
            value="2097152" />
    </inFilter>
</configResolveClass>
```

Less Than or Equal to フィルタ

次に、Less than or Equal to フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
```

```
    classId="sysdebugCore">
    <inFilter>
        <le class="sysdebugCore"
            property="size"
            value="2097152" />
    </inFilter>
</configResolveClass>
```

Wildcard フィルタ

次に、名前がプレフィックス「dmz」で始まるすべての仮想ファイアウォールを検索する Wildcard フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="fwInstance">
    <inFilter>
        <wcard class="fwInstance"
            property="name"
            value="dmz.*" />
    </inFilter>
</configResolveClass>
```

Any Bits フィルタ

次に、Any Bits フィルタの例を示します。

```
<configResolveClass
    cookie="null"
    inHierarchical="false"
    classId="extpolClient">
    <inFilter>
        <anybit class="extpolClient"
            property="capability"
            value="vm-fw,vm-vasw" />
    </inFilter>
</configResolveClass>
```

All Bits フィルタ

次に、All Bits フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="extpolClient">
    <inFilter>
        <allbits class="extpolClient"
            property="capability"
            value="vm-fw,vm-vasw" />
    </inFilter>
</configResolveClass>
```

Composite フィルタ

ここでは、次の内容について説明します。

- 「AND フィルタ」（P.2-11）
- 「OR フィルタ」（P.2-11）

- 「Between フィルタ」（P.2-11）
- 「AND OR NOT Composite フィルタ」（P.2-12）

AND フィルタ

次に、設定に関連付けられたファイアウォールを検索する AND フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="fwComputeFirewall">
    <inFilter>
        <and>
            <eq class="fwComputeFirewall"
                property="assocState"
                value="associated" />
            <eq class="fwComputeFirewall"
                property="configState"
                value="applied" />
        </and>
    </inFilter>
</configResolveClass>
```

OR フィルタ

次に、操作ステータスが可視性なし、または未登録であるすべての管理対象エンドポイントを返す OR フィルタの例を示します。

```
<configResolveClass
    inHierarchical="false"
    cookie="<real_cookie>"
    classId="extpolClient">
    <inFilter>
        <or>
            <eq class="extpolClient"
                property="operState"
                value="unregistered"/>
            <eq class="extpolClient"
                property="operState"
                value="lost-visibility"/>
        </or>
    </inFilter>
</configResolveClass>
```

Between フィルタ

次に、スロット 1、2、3、4、または 5 が使用されているメモリ アレイを検索する Between フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="memoryarray">
    <inFilter>
        <bw class="memoryArray"
            property="populated"
            firstValue="1"
            secondValue="5"/>
    </inFilter>
</configResolveClass>
```

AND OR NOT Composite フィルタ

次に、シャーシ 5 以外のすべてのシャーシのスロット 1 または 8 に存在する、タイプ computeBlade のすべてのオブジェクトを返す AND OR NOT Composite フィルタの例を示します。

```
<configResolveClass
    inHierarchical="false"
    cookie="<real_cookie>"
    classId="computeBlade">
    <inFilter>
        <and>
            <or>
                <eq class="computeBlade"
                    property="slotId"
                    value="1"/>
                <eq class="computeBlade"
                    property="slotId"
                    value="8"/>
            </or>
            <not>
                <eq class="computeBlade"
                    property="chassisId"
                    value="5"/>
            </not>
        </and>
    </inFilter>
</configResolveClass>
```

Modifier フィルタ

NOT フィルタ

次に、NOT Modifier フィルタの例を示します。

```
<configResolveClass
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="extpolClient">
    <inFilter>
        <not>
            <anybit class="extpolClient"
                property="capability"
                value="vm-fw" />
        </not>
    </inFilter>
</configResolveClass>
```

API の例

ここでは、API の設定および管理の例について説明します。

ここでは、次の内容について説明します。

Management Controller を使用した認証

Cisco VNMC へのアクセスはセッションベースであり、ログイン要求で最初に認証する必要があります。デフォルトのセッションの長さは 120 分です。aaaKeepAlive メソッドと aaaRefresh メソッドを使用して、セッションを延長できます。アクティビティが完了したときに、aaaLogout を使用してユーザがセッションから手動でログアウトすることが推奨されます。Cisco VNMC にログイン要求を送信する場合は、サービス タイプ mgmt-controller を使用します。

ここでは、次の内容について説明します。

- 「認証要求」（P.2-13）
- 「認証応答」（P.2-13）

認証要求

次に、認証要求の例を示します。

```
POST URL:  https://10.193.33.221/xmlIM/mgmt-controller
XML API payload:
<aaaLogin
    inName="admin"
    inPassword="Nbv12345"/>
```

認証応答

次に、認証応答の例を示します。

```
<aaaLogin
    cookie=""
    commCookie=""
    srcExtSys="0.0.0.0"
    destExtSys="0.0.0.0"
    srcSvc="" destSvc=""
    response="yes"
    outCookie="<real_cookie>"
    outRefreshPeriod="600"
    outPriv="admin,read-only"
    outDomains=""
    outChannel="fullssl"
    outEvtChannel="fullssl"
    outSessionId="web_13528"
    outVersion="1.0">
</aaaLogin>
```

サービス レジストリを使用したテナント管理

サービス レジストリ（サービス タイプが service-reg）は、ポリシーとリソースを整理するために使用されるテナントとサブ組織を作成および管理します。これらのテナントとサブ組織は、ボトムアップ ポリシーおよびリソース解決のためにツリー階層で配置されます。テナントとサブ組織の作成で使用されるクラス名は次の 4 つのレベルをサポートします。

- テナント、クラス orgTenant：最上位の組織を定義します。
- データセンター、クラス orgDatatcenter：テナント下のデータセンターを定義します。
- アプリケーション、クラス orgApp：データセンター下のアプリケーションを定義します。
- 階層、クラス orgTier：アプリケーション下の階層を定義します。

新しい組織を作成したり、既存の組織を更新したりする場合は、以下を指定します。

- オブジェクト DN
- 属性値
- created または modified に等しいステータス

組織を削除するには、要求で status = deleted を使用します。

ここでは、次の内容について説明します。

- 「組織の作成または更新の要求」（P.2-14）
- 「組織の作成または更新の応答」（P.2-14）

組織の作成または更新の要求

次に、demoTenant という名前のテナントを作成する例を示します。

```
POST URL:  https://10.193.33.221/xmlIM/service-reg
<configConfMos
    cookie="<real_cookie>">
  <inConfigs>
    <pair key="org-root/org-demoTenant">
        <orgTenant dn="org-root/org-demoTenant"
                               name="demoTenant"
                               status="created"/>
    </pair>
  </inConfigs>
</configConfMos>
```

組織の作成または更新の応答

次に、demoTenant という名前のテナントの作成後に受け取る応答の例を示します。

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="2/15/0/839"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="service-reg_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/org-demoTenant">
            <orgTenant
                descr=""
                dn="org-root/org-demoTenant"
                fltAggr="0"
                level="1"
                name="demoTenant"
                status="created"/>
        </pair>
    </outConfigs>
</configConfMos>
```

ポリシー管理

（注）　configConfMos API が単一の設定オブジェクトを指定するために単一の XML 要素 <pair> を使用する各例では、同じ結果を得るために configConfMo API を代わりに使用できます。

この項は、次の内容で構成されています。

デバイス ポリシー

この項は、次の内容で構成されています。

syslog ポリシー

syslog ポリシー オブジェクトは、システム内で実行されるすべてのアクションを追跡します。また、デバイス ポリシーのロギング レベルを設定し、mysyslog を作成します。例 2-8 に、syslog ポリシー要求を示します。

例 2-8　　syslog ポリシー要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/syslog-mysyslog">
            <commSyslog dn="org-root/syslog-mysyslog">
            <commSyslogConsole
                adminState="enabled"
                severity="emergencies"/>
            <commSyslogClient
                name="primary"
                adminState="enabled"
                hostname="5.6.7.8"
                severity="notifications"
                forwardingFacility="local7"/>
            <commSyslogClient
                name="secondary"
                adminState="enabled"
```

```
                hostname="123.23.53.123"
                severity="warnings"
                forwardingFacility="local5"/>
            </commSyslog>
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/1b9"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
        <outConfigs>
            <pair key="org-root/syslog-mysyslog">
            <commSyslog
                adminState="enabled"
                descr=""
                dn="org-root/syslog-mysyslog"
                intId="25301"
                name="mysyslog"
                port="514"
                proto="udp"
                severity="warnings"
                status="created"/>
            </pair>
    </outConfigs>
</configConfMos>
```

SNMP ポリシー

次の例では、**mysnmp** という名前の SNMP ポリシーが作成されます。作成されたポリシーは、デバイス プロファイル リストでこの名前で利用できます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/snmp-mysnmp">
        <commSnmp dn="org-root/snmp-mysnmp"
            adminState="enabled"
            descr="The default SNMP policy"
            sysContact="Andrew Jackson"
            sysLocation="San Jose" >
        <commSnmpCommunity
            community="public"
            role="read-only"/>
        <commSnmpTrap
            hostname="nms-1.cisco.com"
            community="private"/>
        <commSnmpTrap
            hostname="nms-2.cisco.com"
            community="private"/>
        </commSnmp>
        </pair>
    </inConfigs>
```

```
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/1bc"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/snmp-mysnmp">
        <commSnmp
            adminState="enabled"
            descr="The default SNMP policy"
            dn="org-root/snmp-mysnmp"
            intId="25281"
            name="mysnmp"
            port="161"
            proto="all"
            sysContact="Andrew Jackson"
            sysLocation="San Jose"/>
        </pair>
    </outConfigs>
</configConfMos>
```

LogProfile ポリシー

次の例では、**debugLog** という名前の LogProfile ポリシーに対してデバッグ レベルとロギング レベルが設定されます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/logprof-default">
        <policyLogProfile
            dn="org-root/logprof-debugLog"
            name="debugLog"
            level="debug1"
            size="10000000"
            backupCount="4"/>
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/33"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/logprof-debugLog">
            <policyLogProfile
```

```
            adminState="enabled"
            backupCount="4"
            descr="the log level for every process"
            dn="org-root/logprof-debugLog"
            intId="10068"
            level="debug1"
            name="debugLog"
            size="10000000"
            status="modified"/>
        </pair>
    </outConfigs>
</configConfMos>
```

デバイス プロファイル

このアクションにより、**myDeviceProfile** という名前のデバイス プロファイルが作成されます。これにより、プロファイルが有効になり、プロファイルの内容が指定されます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/org-Cola/fwdevprofile-myDeviceProfile">
        <fwpolicyFirewallDeviceProfile
            dn="org-root/org-Cola/fwdevprofile-myDeviceProfile"
            adminState="enabled"
            snmpPolicy="mysnmp"
            syslogPolicy="mysyslog"
            timezone="America/Los_Angeles" >
        <commDns
            name="somedns.com"
            domain="somedns.com"/>
<!-- The order means the device should first use the DNS provider with the smallest
"order" value. If that DNS server is not responsive, use the DNS provider with the next
smaller number. -->
        <commDnsProvider
            hostip="171.70.168.183"
            order="1"/>
        <commDnsProvider
            hostip="171.68.226.120"
            order="2"/>
        <commDnsProvider
            hostip="64.102.6.247"
            order="3"/>
        <commNtpProvider
            name="somentp.com"
            order="1"/>
        <commNtpProvider
            name="north-america.pool.ntp.org"
            order="2"/>
        </fwpolicyFirewallDeviceProfile>
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/1ba"
```

```
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/org-Cola/fwdevprofile-myDeviceProfile">
        <fwpolicyFirewallDeviceProfile
            adminState="enabled"
            coreFilePolicy=""
            descr=""
            dn="org-root/org-Cola/fwdevprofile-myDeviceProfile"
            dnsPolicy=""
            enablePolicyDecisionLog="no"
            faultPolicy=""
            httpPolicy=""
            httpsPolicy=""
            intId="25326"
            logProfilePolicy=""
            name="myDeviceProfile"
            snmpPolicy="mysnmp"
            status="created"
            syslogPolicy="mysyslog"
            telnetPolicy=""
            timezone="America/Los_Angeles"/>
        </pair>
    </outConfigs>
</configConfMos>
```

ゾーン

次の例では、**trustedClients-0** と **trustedServers-0** という名前の 2 つのゾーンが作成されます。また、値を持つ属性セットも割り当てられます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/org-Cisco/zone-trustedClients-0">
<!-- Create a zone of all VMs in the 192.168.1.0/24 subnet -->
            <policyZone dn="org-root/org-Cisco/zone-trustedClients-0">
                <policyZoneCondition id="1" order="1">
                    <policyZoneExpression opr="prefix">
                        <policyIPAddress id="1" value="192.168.1.0" />
                            <policyIPSubnet id="1" value="255.255.255.0" />
                    </policyZoneExpression>
                </policyZoneCondition>
            </policyZone>
        </pair>
        <pair key="org-root/org-Cisco/zone-trustedServers-0">
<!-- Create a zone of all VMs attached to a VNSP where the "appType" is set to
"BuildServer" -->
            <policyZone dn="org-root/org-Cisco/zone-trustedServers-0">
                <policyZoneCondition id="1" order="1">
                    <policyZoneExpression opr="eq">
                        <policyParentAppName id="1" value="BuildServer" />
                    </policyZoneExpression>
                </policyZoneCondition>
            </policyZone>
        </pair>
```

```
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/1a6"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/org-tenant0/zone-zone0">
            <policyZone
                descr=""
                dn="org-root/org-tenant0/zone-zone0"
                intId="24295"
                name="zone0"
                status="created"/>
        </pair>
        <pair key="org-root/org-Cisco/zone-trustedServers-0">
            <policyZone
                descr=""
                dn="org-root/org-Cisco/zone-trustedServers-0"
                intId="24404"
                name="trustedServers-0"
                status="created"/>
        </pair>
        <pair key="org-root/org-Cisco/zone-trustedClients-0">
            <policyZone
                descr=""
                dn="org-root/org-Cisco/zone-trustedClients-0"
                intId="24408"
                name="trustedClients-0"
                status="created"/>
        </pair>
    </outConfigs>
</configConfMos>
```

オブジェクト グループ

次の例では、**ftpPorts0** という名前のオブジェクト グループが作成され、属性セットが割り当てられます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/org-tenant0/objgrp-ftpPorts0">
            <policyObjectGroup dn="org-root/org-tenant0/objgrp-ftpPorts0">
                <policyObjectGroupExpression id="1" order="1" opr="eq">
                    <policyNetworkPort id="1" value="20" />
                </policyObjectGroupExpression>
                <policyObjectGroupExpression id="2" order="2" opr="eq">
                    <policyNetworkPort id="1" value="21" />
                </policyObjectGroupExpression>
            </policyObjectGroup>
        </pair>
```

```
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/1a4"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/org-tenant0/objgrp-ftpPorts0">
            <policyObjectGroup
                attributeName=""
                descr=""
                dn="org-root/org-tenant0/objgrp-ftpPorts0"
                intId="24265"
                name="ftpPorts0"
                status="created"/>
        </pair>
    </outConfigs>
</configConfMos>
```

属性ディクショナリ

次の例では、さまざまな ID と名前を定義するディクショナリが設定されます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
<!-- Create Sample VnspCustomDictionary under org-Cisco -->
        <pair key="org-root/attr-dict-custom-userAttrs">
            <policyVnspCustomDictionary dn="org-root/attr-dict-custom-userAttrs">
                <policyVnspCustomAttr dataType="string" id="1" name="userAttr1" />
                <policyVnspCustomAttr dataType="string" id="2" name="userAttr2" />
                <policyVnspCustomAttr dataType="string" id="3" name="dept" />
                <policyVnspCustomAttr dataType="string" id="4" name="production" />
            </policyVnspCustomDictionary>
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/1a3"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/attr-dict-custom-userAttrs">
            <policyVnspCustomDictionary
                descr=""
                dn="org-root/attr-dict-custom-userAttrs"
```

```
                intId="24245"
                name="userAttrs"
                status="created"/>
        </pair>
    </outConfigs>
</configConfMos>
```

ポリシー

次の例では、**trustedHosts** という名前のポリシーが作成され、使用できるルールが設定されます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/org-Cisco/pol-trustedHosts">
            <policyRuleBasedPolicy dn="org-root/org-Cisco/pol-trustedHosts">
                <policyRule name="allowSsh" order="1">
<!-- This rule allows all VMs in zone "trustedClients" to initiate an SSH connection to
VMs in zone "trustedServers" -->
                <policyRuleCondition id="100" order="1">
                    <policyNetworkExpression opr="eq">
                    <policyNwAttrQualifier attrEp="source"/>
                    <policyZoneNameRef id="1" value="trustedClients-0" />
                    </policyNetworkExpression>
                </policyRuleCondition>
                <policyRuleCondition id="101" order="20">
                    <policyNetworkExpression opr="eq">
                    <policyNwAttrQualifier attrEp="destination"/>
                    <policyZoneNameRef id="1" value="trustedServers-0" />
                    </policyNetworkExpression>
                </policyRuleCondition>
                <policyRuleCondition id="103" order="30">
                    <policyNetworkExpression opr="eq">
                    <policyNwAttrQualifier attrEp="destination"/>
                    <policyNetworkPort id="1" placement="0" value="22" />
                    </policyNetworkExpression>
                </policyRuleCondition>
                <fwpolicyAction actionType="permit"/>
                </policyRule>
                <policyRule name="allowTacacs" order="2">
                <fwpolicyAction actionType="permit"/>
                </policyRule>
            </policyRuleBasedPolicy>
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/1b5"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/org-Cisco/pol-trustedHosts">
```

```
            <policyRuleBasedPolicy
                descr=""
                dn="org-root/org-Cisco/pol-trustedHosts"
                intId="25131"
                name="trustedHosts"
                status="created"/>
        </pair>
    </outConfigs>
</configConfMos>
```

PolicySet

次の例では、**myPolicySet0** が作成され、ポリシーが適用される順序が設定されます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/org-tenant0/pset-myPolicySet0">
            <policyPolicySet
                dn="org-root/org-tenant0/pset-myPolicySet0"
<!-- Policy Set contains references to three policies. Policies are resolved by name in
the org hierarchy. Ordering of policies is important, so we use the "order" property to
order the policies. Note that two policy sets can refer to the same  policies, and use a
different order. -->
            <policyPolicyNameRef order="1" policyName="trustedHosts"/>
            </policyPolicySet>
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookied>"
    commCookie="7/15/0/1a8"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/org-tenant0/pset-myPolicySet0">
            <policyPolicySet
                adminState="enabled"
                descr=""
                dn="org-root/org-tenant0/pset-myPolicySet0"
                intId="24431"
                name="myPolicySet0"
                status="created"/>
        </pair>
    </outConfigs>
</configConfMos>
```

セキュリティ プロファイル

次の例では、**vnsp-sp1** という名前のセキュリティ プロファイルが作成され、VNSP がそのセキュリティ プロファイルに割り当てられます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/policy-mgr
XML API payload:
<configConfMos
cookie="<real_cookie>">
<inConfigs>
<pair key="org-root/org-tenant0/vnsp-sp1">
    <policyVirtualNetworkServiceProfile
        dn="org-root/org-tenant0/vnsp-sp1"
        policySetNameRef="myPolicySet0">
        <policyVnspAVPair id="1">
            <policyAttributeDesignator attrName="dept"/>
            <policyAttributeValue value="DEV" />
        </policyVnspAVPair>
        </policyVirtualNetworkServiceProfile>
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/1ac"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/org-tenant0/vnsp-sp1">
            <policyVirtualNetworkServiceProfile
                descr=""
                dn="org-root/org-tenant0/vnsp-sp1"
                intId="24512"
                name="sp1"
                policySetNameRef="myPolicySet0"
                status="created"
                vnspId="2"/>
        </pair>
    </outConfigs>
</configConfMos>
```

リソース管理

リソース管理コンポーネントは、次の機能を実行します。

- コンピュータ ファイアウォールをデバイス ポリシーに割り当てます。
- 利用可能なファイアウォール インスタンスがないか問い合わせます。
- ファイアウォール インスタンスを管理対象オブジェクトに関連付けます。

この項は、次の内容で構成されています。

コンピュータ ファイアウォール

次の例では、デバイス ポリシーに **computeFirewall** が割り当てられます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/resource-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/org-Cisco/cfw-VSNaaa">
            <fwComputeFirewall dn="org-root/org-Cisco/cfw-VSNaaa"
                devicePolicy="myDeviceProfile"
                hostname="cfw-VSNaaa"
                dataIpAddress="10.10.10.100">
            </fwComputeFirewall>
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="5/15/0/19a"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="resource-mgr_dme"
    response="yes">
     <outConfigs>
        <pair key="org-root/org-Cisco/cfw-VSNaaa">
            <fwComputeFirewall
                assocState="unassociated"
                configState="not-applied"
                dataIpAddress="10.10.10.100"
                dataIpSubnet="255.255.255.0"
                descr=""
                devicePolicy="myDeviceProfile"
                dn="org-root/org-Cisco/cfw-VSNaaa"
                fltAggr="0"
                hostname="cfw-VSNaaa"
                intId="12368"
                name="VSNaaa"
                status="created"/>
        </pair>
    </outConfigs>
</configConfMos>
```

ファイアウォール インスタンスの問い合わせ

次の例では、すべてのファイアウォール インスタンスとその属性の問い合わせが行われます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/resource-mgr
XML API payload:
<configResolveClass
    cookie="<real_cookie>"
    classId="fwInstance"
    inHierarchical="false">
    <inFilter>
```

```
        <eq class="fwInstance"
            property="mgmtIp"
            value="10.193.33.221" />
    </inFilter>
</configResolveClass>
```

応答

```
configResolveClass
    cookie=<real_cookie>"
    commCookie="5/15/0/1d9"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="resource-mgr_dme"
    response="yes"
    classId="fwInstance">
    <outConfigs>
        <fwInstance
            assignedToDn=""
            assoc="none"
            descr=""
            dn="fw/inst-1005"
            fltAggr="0"
            fsmDescr=""
            fsmPrev="DisassociateSuccess"
            fsmProgr="100"
            fsmRmtInvErrCode="none"
            fsmRmtInvErrDescr=""
            fsmRmtInvRslt=""
            fsmStageDescr=""
            fsmStamp="2010-12-03T23:18:13.304"
            fsmStatus="nop"
            fsmTry="0"
            intId="10155"
            mgmtIp="10.193.33.221"
            model=""
            name="firewall"
            pooled="0"
            registeredClientDn="extpol/reg/clients/client-1005"
            revision="0"
            serial=""
            svcId="1005"
            vendor=""/>
    </outConfigs>
</configResolveClass>
```

ファイアウォール インスタンスの関連付け

次の例では、ファイアウォール インスタンスが管理対象オブジェクトに関連付けられ、firewall-0 が inst-1005 に割り当てられます。

要求

```
POST URL:  https://10.193.33.221/xmlIM/resource-mgr
XML API payload:
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/org-Cola/cfw-firewall-0">
            <fwComputeFirewall dn="org-root/org-Cola/cfw-firewall-0">
            <fwResourceBinding assignedToDn="fw/inst-1005"/>
            </fwComputeFirewall>
```

```
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="5/15/0/1df"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="resource-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/org-Cola/cfw-firewall-0">
            <fwComputeFirewall
                assocState="associated"
                configState="not-applied"
                dataIpAddress="168.1.1.221"
                dataIpSubnet="255.255.255.0"
                descr=""
                devicePolicy="myDeviceProfile"
                dn="org-root/org-Cola/cfw-firewall-0"
                fltAggr="1"
                hostname="cfw-firewall-0"
                intId="12514"
                name="firewall-0"
                status="modified"/>
        </pair>
    </outConfigs>
</configConfMos>
```

CHAPTER 3

Cisco VNMC XML API メソッドの説明

この章では、次の内容を説明します。

API メソッドの詳細は以下に示されています。

Cisco VNMC XML API メソッド

これらのメソッドは、GUI コンソールからも呼び出されます。ここでは、API メソッド、構文（要求および応答）、使用例について説明します。Cisco VNMC の API メソッドは次のように定義されます。

- aaaGetRemoteUserRoles
- aaaGetUserLocales
- aaaKeepAlive
- aaaLogin
- aaaLogout
- aaaRefresh
- configConfFiltered
- configConfMo
- configConfMoGroup
- configConfMos
- configFindDnsByClassId
- configConfMoGroup
- configMoChangeEvent
- configResolveClass
- configResolveClasses
- configResolveDn
- configResolveDns
- configResolveParent
- configScope
- eventSendHeartbeat
- eventSubscribe
- eventSubscribeApps
- faultAckFault
- faultAckFaults
- faultResolveFault
- loggingSyncOcns
- methodVessel
- orgResolveElements
- orgResolveInScope
- orgResolveLogicalParents
- policyEstimateImpact
- poolResolveInScope

aaaGetRemoteUserRoles

この例は、リモート ロケーションのユーザ特権を返します。

要求構文

```
<xs:element name="aaaGetRemoteUserRoles" type="aaaGetRemoteUserRoles"
substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaGetRemoteUserRoles" mixed="true">
            <xs:attribute name="inRemoteUserName">
                <xs:simpleType>
```

```
                    <xs:restriction base="xs:string">
                        <xs:pattern value="[a-zA-Z][a-zA-Z0-9_.@-]{0,31}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="aaaGetRemoteUserRoles" type="aaaGetRemoteUserRoles"
substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaGetRemoteUserRoles" mixed="true">
            <xs:attribute name="outRemoteUserPriv">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern
value="((policy|aaa|read-only|admin|tenant|operations|res-config|fault),){0,7}(policy|aaa|
read-only|admin|tenant|operations|res-config|fault){0,1}"/>
                </xs:restriction>
            </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<aaaGetRemoteUserRoles
    cookie="<real_cookie>"
    inRemoteUserName="adminuser"
    outRemoteUserPriv/>
```

応答

```
<aaaGetRemoteUserRoles
    cookie="<real_cookie>"
    commCookie="11/15/0/2964"
    srcExtSys="10.193.33.109"
    destExtSys="10.193.33.109"
    srcSvc="sam_extXMLApi"
    destSvc="mgmt-controller_dme"
    response="yes"
    outRemoteUserPriv="admin">
</aaaGetRemoteUserRoles>
```

aaaGetUserLocales

この例は、認可されたユーザ ロケーションのリストを返します。

要求構文

```
<xs:element name="aaaGetUserLocales" type="aaaGetUserLocales"
substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaGetUserLocales" mixed="true">
            <xs:attribute name="inUserName">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="[a-zA-Z][a-zA-Z0-9_.@-]{0,31}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inIsUserRemote">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="aaaGetUserLocales" type="aaaGetUserLocales"
substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaGetUserLocales" mixed="true">
            <xs:attribute name="outUserLocales">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:minLength value="0"/>
                        <xs:maxLength value="512"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<aaaGetUserLocales
    cookie="<real_cookie>"
    inUserName="john"
    inIsUserRemote="no"
    outUserLocales/>
```

応答

```
<aaaGetUserLocales
```

```
    cookie="<real_cookie>"
    commCookie="11/15/0/2962"
    srcExtSys="10.193.33.109"
    destExtSys="10.193.33.109"
    srcSvc="sam_extXMLApi"
    destSvc="mgmt-controller_dme"
    response="yes"
    outUserLocales="TestSanity">
</aaaGetUserLocales>
```

aaaKeepAlive

この例では、デフォルトのセッション時間が経過し、メソッド呼び出しの後で同じ Cookie が使用されるまでセッションがアクティブなままになります。

要求構文

```
<xs:element name="aaaKeepAlive" type="aaaKeepAlive" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaKeepAlive" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="aaaKeepAlive" type="aaaKeepAlive" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaKeepAlive" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<aaaKeepAlive
    cookie="<real_cookie>" />
```

応答

```
<aaaKeepAlive
    cookie="<real_cookie>"
    commCookie="11/15/0/2969"
    srcExtSys="10.193.33.109"
    destExtSys="10.193.33.109"
    srcSvc="sam_extXMLApi"
    destSvc="mgmt-controller_dme"
    response="yes">
</aaaKeepAlive>
```

aaaLogin

この例は、セッションを開始するために必要で、クライアントと VNMC 間で認証された HTTPS セッションを確立するログイン プロセスを示しています。

要求構文

```
<xs:element name="aaaLogin" type="aaaLogin" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaLogin" mixed="true">
            <xs:attribute name="inName">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="[¥-¥.:_a-zA-Z0-9]{0,16}"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inPassword">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:minLength value="0"/>
                        <xs:maxLength value="512"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="aaaLogin" type="aaaLogin" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaLogin" mixed="true">
            <xs:attribute name="outCookie" type="xs:string"/>
            <xs:attribute name="outRefreshPeriod" type="xs:unsignedInt"/>
            <xs:attribute name="outPriv">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern
value="((policy|aaa|read-only|admin|tenant|operations|res-config|fault),){0,7}(policy|aaa|
read-only|admin|tenant|operations|res-config|fault){0,1}"/>
                </xs:restriction>
            </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="outDomains" type="xs:string"/>
            <xs:attribute name="outChannel">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="fullssl"/>
                        <xs:enumeration value="noencssl"/>
                        <xs:enumeration value="plain"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="outEvtChannel">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="fullssl"/>
                        <xs:enumeration value="noencssl"/>
                        <xs:enumeration value="plain"/>
                    </xs:restriction>
```

```
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="outSessionId">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                     <xs:minLength value="1"/>
                     <xs:maxLength value="64"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="outVersion" type="xs:string"/>
        <xs:attribute name="cookie" type="xs:string"/>
        <xs:attribute name="response" type="YesOrNo"/>
        <xs:attribute name="errorCode" type="xs:unsignedInt"/>
        <xs:attribute name="errorDescr" type="xs:string"/>
        <xs:attribute name="invocationResult" type="xs:string"/>
    </xs:complexType>
```

例

要求

```
<aaaLogin
    inName="admin"
    inPassword="Nbv12345"/>
```

応答

```
<aaaLogin cookie=""
    commCookie=""
    srcExtSys="0.0.0.0"
    destExtSys="0.0.0.0"
    srcSvc="" destSvc=""
    response="yes"
    outCookie="<real_cookie>"
    outRefreshPeriod="600"
    outPriv="admin"
    outDomains=""
    outChannel="fullssl"
    outEvtChannel="fullssl"
    outSessionId="web_49019"
    outVersion="1.0(0.39938)">
</aaaLogin>
```

aaaLogout

次に、現在のセッションを終了するログアウト プロセスの例を示します。デフォルトのセッション時間が経過すると、このプロセスは自動的に呼び出されます。

要求構文

```
<xs:element name="aaaLogout" type="aaaLogout" substitutionGroup="externalMethod"/>
    <xs:complexType name="aaaLogout" mixed="true">
        <xs:attribute name="inCookie" type="xs:string"/>
        <xs:attribute name="cookie" type="xs:string"/>
        <xs:attribute name="response" type="YesOrNo"/>
    </xs:complexType>
```

応答構文

```
<xs:element name="aaaLogout" type="aaaLogout" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaLogout" mixed="true">
            <xs:attribute name="outStatus">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:enumeration value="success"/>
                         <xs:enumeration value="failure"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<aaaLogout
    inCookie="<real_cookie>"
    outStatus/>
```

応答

```
<aaaLogout cookie=""
    commCookie=""
    srcExtSys="0.0.0.0"
    destExtSys="0.0.0.0"
    srcSvc=""
    destSvc=""
    response="yes"
    outStatus="success">
</aaaLogout>
```

aaaRefresh

セッションはユーザ アクティビティによってアクティブに保つことができます（デフォルトのセッション時間枠内）。デフォルトでは、アクティビティがない時点から 7200 秒カウントダウンされます。7200 秒が経過すると、Cisco VNMC はスリープ モードを開始し、再度サインインを要求します。これにより、カウント ダウンが再起動されます。セッションは同じセッション ID を使用し続けます。

（注）　このメソッドを使用すると、以前の Cookie の有効期限が切れ、新しい Cookie が発行されます。

要求構文

```
<xs:element name="aaaRefresh" type="aaaRefresh" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaRefresh" mixed="true">
            <xs:attribute name="inName">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                         <xs:pattern value="[¥-¥.:_a-zA-Z0-9]{0,16}"/>
```

```
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inPassword">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:minLength value="0"/>
                        <xs:maxLength value="512"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inCookie" type="xs:string"/>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="aaaRefresh" type="aaaRefresh" substitutionGroup="externalMethod"/>
        <xs:complexType name="aaaRefresh" mixed="true">
            <xs:attribute name="outCookie" type="xs:string"/>
            <xs:attribute name="outRefreshPeriod" type="xs:unsignedInt"/>
            <xs:attribute name="outPriv">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern
value="((policy|aaa|read-only|admin|tenant|operations|res-config|fault),){0,7}(policy|aaa|
read-only|admin|tenant|operations|res-config|fault){0,1}"/>
                </xs:restriction>
            </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="outDomains" type="xs:string"/>
            <xs:attribute name="outChannel">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="fullssl"/>
                        <xs:enumeration value="noencssl"/>
                        <xs:enumeration value="plain"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="outEvtChannel">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:enumeration value="fullssl"/>
                        <xs:enumeration value="noencssl"/>
                        <xs:enumeration value="plain"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<aaaRefresh
    cookie="<real_cookie>"
    inName="admin"
    inPassword="Nbv12345"
    inCookie="<real_cookie>"/>
```

応答

```
<aaaRefresh
    cookie="<real_cookie>"
    commCookie="" srcExtSys="0.0.0.0"
    destExtSys="0.0.0.0"
    srcSvc=""
    destSvc=""
    response="yes"
    outCookie="<real_cookie>"
    outRefreshPeriod="600"
    outPriv="admin"
    outDomains=""
    outChannel="fullssl"
    outEvtChannel="fullssl">
</aaaRefresh>
```

configConfFiltered

次に、設定されたポリシーに基づいて、データおよびアクティビティが制限されている例を示します。

要求構文

```
<xs:element name="configConfFiltered" type="configConfFiltered"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfFiltered" mixed="true">
            <xs:all>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
                <xs:element name="inConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                         <xs:simpleType>
                             <xs:restriction base="xs:string">
                                 <xs:enumeration value="no"/>
                                 <xs:enumeration value="yes"/>
                             </xs:restriction>
                         </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configConfFiltered" type="configConfFiltered"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfFiltered" mixed="true">
```

```
            <xs:all>
                <xs:element name="outConfigs" type="configSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

例

要求

```
<configConfFiltered
    cookie="<real_cookie>"
    inHierarchical="false"
    classId="orgTenant">
    <inFilter>
        <eq class="orgTenant"
            property="name"
            value="Cisco" />
    </inFilter>
    <inConfig>
        <orgDatacenter
            dn="org-HR"
            descr="HR (Human Resources- new Descr)"/>
    </inConfig>
</configConfFiltered>
```

応答

```
<configConfFiltered
    cookie="<real_cookie>"
    commCookie="5/15/0/617"
    srcExtSys="10.193.33.206"
    destExtSys="10.193.33.206"
    srcSvc="sam_extXMLApi"
    destSvc="resource-mgr_dme"
    response="yes"
    classId="orgTenant">
    <outConfigs>
        <orgDatacenter
            descr="HR (Human Resources- new Descr)"
            dn="org-root/org-Cisco/org-HR"
            fltAggr="0"
            level="2"
            name="HR"
            status="modified"/>
    </outConfigs>
</configConfFiltered>
```

configConfMo

次に、単一のサブツリー（DN など）で、指定された管理対象オブジェクトを設定する例を示します。

要求構文

```
<xs:element name="configConfMo" type="configConfMo" substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfMo" mixed="true">
            <xs:all>
                <xs:element name="inConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                         <xs:simpleType>
                             <xs:restriction base="xs:string">
                                 <xs:enumeration value="no"/>
                                 <xs:enumeration value="yes"/>
                             </xs:restriction>
                         </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configConfMo" type="configConfMo" substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfMo" mixed="true">
            <xs:all>
                <xs:element name="outConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

例

要求

```
<configConfMo
    dn=""
    cookie="<real_cookie>"
    inHierarchical="false">
    <inConfig>
        <aaaLdapEp
            attribute="CiscoAvPair"
            basedn="dc=pasadena,dc=cisco,dc=com"
            descr=""
            dn="sys/ldap-ext"
            filter="sAMAccountName=$userid"
            retries="1"
            status="modified"
            timeout="30"/>
    </inConfig>
</configConfMo>
```

応答

```
<configConfMo
    dn=""
    cookie="<real_cookie>"
    commCookie="11/15/0/28"
    srcExtSys="10.193.33.101"
    destExtSys="10.193.33.101"
    srcSvc="sam_extXMLApi"
    destSvc="mgmt-controller_dme"
    response="yes">
    <outConfig>
        <aaaLdapEp
            attribute="CiscoAvPair"
            basedn="dc=pasadena,dc=cisco,dc=com"
            childAction="deleteNonPresent"
            descr=""
            dn="sys/ldap-ext"
            filter="sAMAccountName=$userid"
            fsmDescr=""
            fsmPrev="updateEpSuccess"
            fsmProgr="100"
            fsmStageDescr=""
            fsmStamp="2010-11-22T23:41:01.826"
            fsmStatus="nop"
            fsmTry="0"
            intId="10027"
            name=""
            retries="1"
            status="modified"
            timeout="30"/>
    </outConfig>
</configConfMo>
```

configConfMoGroup

次に、設定されたポリシーに基づいて、管理対象オブジェクトのグループを設定する例を示します。

要求構文

```
<xs:element name="configConfMoGroup" type="configConfMoGroup"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfMoGroup" mixed="true">
            <xs:all>
                <xs:element name="inDns" type="dnSet" minOccurs="0"/>
                <xs:element name="inConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                         <xs:simpleType>
                             <xs:restriction base="xs:string">
                                 <xs:enumeration value="no"/>
                                 <xs:enumeration value="yes"/>
                             </xs:restriction>
                         </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
```

```
        </xs:complexType>
```

応答構文

```
<xs:element name="configConfMoGroup" type="configConfMoGroup"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfMoGroup" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

（注）　orgDataCenter（org-root/org-Cisco および org-root/org-Cola 下）の descr プロパティが変更されます。descr プロパティは暗黙的でないため、変更できます。暗黙的な場合、変更は適用されず、新しい orgDataCenter が作成されます。

要求

```
<configConfMoGroup
    cookie="<real_cookie>"
    inHierarchical="false">
    <inDns>
        <dn value="org-root/org-Cisco" />
        <dn value="org-root/org-Cola" />
    </inDns>
    <inConfig>
        <orgDatacenter
            dn="org-HR"
            descr="HR (Human Resources)"/>
    </inConfig>
</configConfMoGroup>
```

応答

```
<configConfMoGroup
    cookie="<real_cookie>"
    commCookie="5/15/0/600"
    srcExtSys="10.193.33.206"
    destExtSys="10.193.33.206"
    srcSvc="sam_extXMLApi"
    destSvc="resource-mgr_dme"
    response="yes">
    <outConfigs>
        <orgDatacenter
            descr="HR (Human Resources)"
            dn="org-root/org-Cola/org-HR"
            fltAggr="0"
            level="2"
            name="HR"
            status="modified"/>
        <orgDatacenter
            descr="HR (Human Resources)"
```

```
            dn="org-root/org-Cisco/org-HR"
            fltAggr="0"
            level="2"
            name="HR"
            status="modified"/>
    </outConfigs>
</configConfMoGroup>
```

configConfMos

次に、複数のサブツリーで DN を使用して管理対象オブジェクトを設定する例を示します。

要求構文

```
<xs:element name="configConfMos" type="configConfMos" substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfMos" mixed="true">
            <xs:all>
                <xs:element name="inConfigs" type="configMap" minOccurs="0">
                    <xs:unique name="unique_map_key_2">
                         <xs:selector xpath="pair"/>
                         <xs:field xpath="@key"/>
                    </xs:unique>
                </xs:element>
            </xs:all>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                         <xs:simpleType>
                             <xs:restriction base="xs:string">
                                 <xs:enumeration value="no"/>
                                 <xs:enumeration value="yes"/>
                             </xs:restriction>
                         </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configConfMos" type="configConfMos" substitutionGroup="externalMethod"/>
        <xs:complexType name="configConfMos" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configMap" minOccurs="0">
                    <xs:unique name="unique_map_key_4">
                         <xs:selector xpath="pair"/>
                         <xs:field xpath="@key"/>
                    </xs:unique>
                </xs:element>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<configConfMos
    cookie="<real_cookie>">
    <inConfigs>
        <pair key="org-root/logprof-default">
            <policyLogProfile dn="org-root/logprof-default"
                name="default"
                level="debug1"
                size="10000000"
                backupCount="4"/>
        </pair>

    <!-- Update Controller Device Profile -->
        <pair key="org-root/controller-profile-default">
            <policyControllerDeviceProfile
                dn="org-root/controller-profile-default"
                adminState="enabled">

                <commDnsProvider hostip="171.70.168.183" order="1"/>
                <commDnsProvider hostip="171.68.226.120" order="2"/>
                <commDnsProvider hostip="64.102.6.247"   order="3"/>
            </policyControllerDeviceProfile>
        </pair>
    </inConfigs>
</configConfMos>
```

応答

```
<configConfMos
    cookie="<real_cookie>"
    commCookie="7/15/0/1a74"
    srcExtSys="10.193.34.70"
    destExtSys="10.193.34.70"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="org-root/logprof-default">
            <policyLogProfile
                adminState="enabled"
                backupCount="4"
                descr="the log level for every process"
                dn="org-root/logprof-default"
                intId="10065"
                level="debug1"
                name="default"
                size="10000000"/>
        </pair>
        <pair key="org-root/controller-profile-default">
            <policyControllerDeviceProfile
                adminState="enabled"
                coreFilePolicy=""
                descr="default profile for management server virtual machine"
                dn="org-root/controller-profile-default"
                dnsPolicy=""
                faultPolicy="default"
                httpPolicy="default"
                httpsPolicy="default"
                intId="10057"
                logProfilePolicy="default"
                name="default"
```

```
                snmpPolicy=""
                syslogPolicy=""
                telnetPolicy=""
                timezone=""/>
        </pair>
    </outConfigs>
</configConfMos>
```

configFindDnsByClassId

次に、識別名を検索し、クラス ID でソートされた識別名を返す例を示します。

要求構文

```
<xs:element name="configFindDnsByClassId" type="configFindDnsByClassId"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configFindDnsByClassId" mixed="true">
            <xs:all>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configFindDnsByClassId" type="configFindDnsByClassId"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configFindDnsByClassId" mixed="true">
            <xs:all>
                <xs:element name="outDns" type="dnSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

例

要求

```
<configFindDnsByClassId
    classId="computeBlade"
    cookie="<real_cookie>" />
```

応答

```
<configFindDnsByClassId
    cookie="<real_cookie>"
    response="yes"
    classId="computeBlade">
    <outDns>
        <dn value="sys/chassis-1/blade-7"/>
        <dn value="sys/chassis-1/blade-5"/>
```

```
        <dn value="sys/chassis-1/blade-3"/>
        <dn value="sys/chassis-1/blade-1"/>
    </outDns>
</configFindDnsByClassId>
```

configMoChangeEvent

次に、指定された管理対象オブジェクトに対する変更イベントを返す例を示します。

要求構文

```
<xs:element name="configMoChangeEvent" type="configMoChangeEvent"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configMoChangeEvent" mixed="true">
            <xs:all>
                <xs:element name="inConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inEid" type="xs:unsignedLong"/>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configMoChangeEvent" type="configMoChangeEvent"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configMoChangeEvent" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<configMoChangeEvent
    cookie=""
    commCookie=""
    srcExtSys="10.193.33.120"
    destExtSys="10.193.33.120"
    srcSvc="service-reg_dme"
    destSvc="sam_extXMLApi"
    inEid="71449">
    <inConfig>
        <extpolRegistry
            dn="extpol/reg"
            lastPollTs="2010-11-12T20:33:51.071"
            status="modified"/>
    </inConfig>
</configMoChangeEvent>
```

応答

```
Event (12:35:20:675):
<eventSendHeartbeat
```

```
    cookie="0/0/0/2a74"
    commCookie=""
    srcExtSys="0.0.0.0"
    destExtSys="0.0.0.0"
    srcSvc="" destSvc=""
    response="yes"
    outSystemTime="2010-11-12T20:34:19.630">
</eventSendHeartbeat>

Event (12:35:20:690):
<eventSendHeartbeat
    cookie="0/0/0/2a74"
    commCookie=""
    srcExtSys="0.0.0.0"
    destExtSys="0.0.0.0"
    srcSvc="" destSvc=""
    response="yes"
    outSystemTime="2010-11-12T20:34:19.630">
</eventSendHeartbeat>
```

configResolveChildren

次に、管理対象情報ツリーで特定の DN 下にある管理対象オブジェクトの子を取得する例を示します。返される子の数を減らすためにフィルタを使用できます。

要求構文

```
<xs:element name="configResolveChildren" type="configResolveChildren"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveChildren" mixed="true">
            <xs:all>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inDn" type="referenceObject"/>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                         <xs:simpleType>
                             <xs:restriction base="xs:string">
                                 <xs:enumeration value="no"/>
                                 <xs:enumeration value="yes"/>
                             </xs:restriction>
                         </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configResolveChildren" type="configResolveChildren"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveChildren" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configSet" minOccurs="0"/>
            </xs:all>
```

```
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

例

要求

```
<configResolveChildren
    cookie="<real_cookie>"
    classId="aaaUser"
    inDn="sys/user-ext"
    inHierarchical="false">
    <inFilter>
    </inFilter>
</configResolveChildren>
```

応答

```
<configResolveChildren
    cookie="<real_cookie>"
    commCookie="11/15/0/2a59"
    srcExtSys="10.193.33.120"
    destExtSys="10.193.33.120"
    srcSvc="sam_extXMLApi"
    destSvc="mgmt-controller_dme"
    response="yes"
    classId="aaaUser">
    <outConfigs>
        <aaaUser descr="" dn="sys/user-ext/user-doe"
            email="" expiration="never" expires="no" firstName="John" intId="12999"
            lastName="Doe" name="doe" phone="" priv="admin,read-only" pwdSet="yes"/>
        <aaaUser descr="" dn="sys/user-ext/user-jacks" email="" expiration="never"
            expires="no" firstName="Play" intId="12734" lastName="Jacks" name="jacks"
            phone="" priv="fault,operations,policy,read-only,res-config,tenant"
            pwdSet="yes"/>
        <aaaUser descr="" dn="sys/user-ext/user-admin" email="" expiration="never"
            expires="no" firstName="" intId="10052" lastName="" name="admin" phone=""
            priv="admin,read-only" pwdSet="yes"/>
        <aaaUser descr="" dn="sys/user-ext/user-over" email="" expiration="never"
            expires="no" firstName="Roll" intId="12711" lastName="Over" name="over"
            phone="" priv="fault,operations,policy,read-only,res-config,tenant"
            pwdSet="yes"/>
        <aaaUser descr="" dn="sys/user-ext/user-fun" email="" expiration="never"
            expires="no" firstName="Have" intId="12708" lastName="Fun" name="fun" phone=""
            priv="read-only" pwdSet="yes"/>
        <aaaUser descr="testuser" dn="sys/user-ext/user-aaa" email="" expiration="never"
            expires="no" firstName="a" intId="10620" lastName="aa" name="aaa" phone=""
            priv="aaa,read-only" pwdSet="no"/>
    </outConfigs>
</configResolveChildren>
```

configResolveClass

次に、該当するクラスの要求された管理対象オブジェクトを返す例を示します。*inHierarchical* が true の場合は、オブジェクトに子が含まれます。

要求構文

```
<xs:element name="configResolveClass" type="configResolveClass"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveClass" mixed="true">
            <xs:all>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                         <xs:simpleType>
                             <xs:restriction base="xs:string">
                                 <xs:enumeration value="no"/>
                                 <xs:enumeration value="yes"/>
                             </xs:restriction>
                         </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configResolveClass" type="configResolveClass"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveClass" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="classId" type="namingClassId"/>
        </xs:complexType>
```

例

要求

```
<configResolveClass
    cookie="<real_cookie>"
    classId="pkiEp"
    inHierarchical="false">
    <inFilter>
    </inFilter>
</configResolveClass>
```

応答

```
<configResolveClass
    cookie="<real_cookie>"
    commCookie="11/15/0/2a5b"
    srcExtSys="10.193.33.120"
    destExtSys="10.193.33.120"
    srcSvc="sam_extXMLApi"
```

```
    destSvc="mgmt-controller_dme"
    response="yes"
    classId="pkiEp">
    <outConfigs>
        <pkiEp descr=""
            dn="sys/pki-ext"
            intId="10037"
            name=""/>
    </outConfigs>
</configResolveClass>
```

configResolveClasses

次に、複数のクラスの要求された管理対象オブジェクトを返す例を示します。*inHierarchical* が true の場合は、オブジェクトに子が含まれます。

要求構文

```
<xs:element name="configResolveClasses" type="configResolveClasses"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveClasses" mixed="true">
            <xs:all>
                <xs:element name="inIds" type="classIdSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                         <xs:simpleType>
                             <xs:restriction base="xs:string">
                                 <xs:enumeration value="no"/>
                                 <xs:enumeration value="yes"/>
                             </xs:restriction>
                         </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configResolveClasses" type="configResolveClasses"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveClasses" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<configResolveClasses
    cookie="<real_cookie>"
    inHierarchical="false">
    <inIds>
        <Id value="computeBlade"/>
        <Id value="equipmentChassis"/>
    </inIds>
</configResolveClasses>
```

応答

(この応答は、省略されています)。

```
<configResolveClasses
    cookie="<real_cookie>"
    response="yes">
    <outConfigs>
        <computeBlade
            adminPower="policy"
            adminState="in-service"
            dn="sys/chassis-1/blade-1"
            .
            .
        <computeBlade
            adminPower="policy"
            adminState="in-service"
            dn="sys/chassis-1/blade-3"
            .
            .
        <computeBlade
            adminPower="policy"
            adminState="in-service"
            dn="sys/chassis-1/blade-5"
            .
            .
        <computeBlade
            adminPower="policy"
            adminState="in-service"
            dn="sys/chassis-1/blade-7".
            .
            .
        <equipmentChassis
            adminState="acknowledged"
            configState="ok"
            .
            .
    </outConfigs>
</configResolveClasses>
```

configResolveDn

次に、指定された DN について単一の管理対象オブジェクトを取得する例を示します。

要求構文

```
<xs:element name="configResolveDn" type="configResolveDn"
substitutionGroup="externalMethod"/>
```

```
        <xs:complexType name="configResolveDn" mixed="true">
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configResolveDn" type="configResolveDn"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveDn" mixed="true">
            <xs:all>
                <xs:element name="outConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

例

要求

```
<configResolveDn
    cookie="<real_cookie>"
    dn="vmmEp/vm-mgr-vcenter1" />
```

応答

```
<configResolveDn dn="vmmEp/vm-mgr-vcenter1"
    cookie="<real_cookie>"
    commCookie="9/15/0/1c0d"
    srcExtSys="10.193.34.70"
    destExtSys="10.193.34.70"
    srcSvc="sam_extXMLApi"
    destSvc="vm-mgr_dme"
    response="yes">
    <outConfig>
        <vmManager
            adminState="enable"
            descr=""
            dn="vmmEp/vm-mgr-vcenter1"
            fltAggr="0"
            fsmDescr="AG registration with
vCenter(FSM:sam:dme:VmManagerRegisterWithVCenter)"
            fsmPrev="RegisterWithVCenterRegistering"
            fsmProgr="13"
```

```
            fsmRmtInvErrCode="none"
            fsmRmtInvErrDescr=""
            fsmRmtInvRslt=""
            fsmStageDescr="AG registration with
vCenter(FSM-STAGE:sam:dme:VmManagerRegisterWithVCenter:Registering)"
            fsmStamp="2010-11-11T21:37:15.696"
            fsmStatus="RegisterWithVCenterRegistering"
            fsmTry="1"
            hostName="savbu-vpod-dev-31.cisco.com"
            intId="21959"
            name="vcenter1"
            operState="unknown"
            stateQual=""
            type="vmware"
            version=""/>
    </outConfig>
</configResolveDn>
```

configResolveDns

次に、DN のリストについて管理対象オブジェクトを取得する例を示します。

要求構文

```
<xs:element name="configResolveDns" type="configResolveDns"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveDns" mixed="true">
            <xs:all>
                <xs:element name="inDns" type="dnSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                         <xs:simpleType>
                             <xs:restriction base="xs:string">
                                 <xs:enumeration value="no"/>
                                 <xs:enumeration value="yes"/>
                             </xs:restriction>
                         </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configResolveDns" type="configResolveDns"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveDns" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configSet" minOccurs="0"/>
                <xs:element name="outUnresolved" type="dnSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
```

```
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<configResolveDns
    cookie="<real_cookie>"
    inHierarchical="false">
    <inDns>
        <dn value="sys/chassis-1" />
        <dn value="sys/chassis-1/blade-1/board/cpu-1" />
        <dn value="sys/chassis-1/blade-1/board/t-stats" />
    </inDns>
</configResolveDns>
```

応答

```
<configResolveDns
    cookie="<real_cookie>"
    response="yes">
    <outConfigs>
        <processorUnit
            arch="Xeon"
            cores="4"
            dn="sys/chassis-1/blade-1/board/cpu-1"
            id="1"
            model="Intel(R) Xeon(R) CPU E5520 @ 2.27GHz"
            operState="operable"
            operability="operable"
            perf="not-supported"
            power="not-supported"
            presence="equipped"
            revision="0"
            serial=""
            socketDesignation="CPU1"
            speed="2.266000"
            stepping="5"
            thermal="ok"
            threads="8"
            vendor="Intel(R) Corporation"
            voltage="ok"/>
        <equipmentChassis
            adminState="acknowledged"
            configState="ok"
            connPath="A,B"
            connStatus="A,B"
            dn="sys/chassis-1"
            fabricEpDn="fabric/server/chassis-1"
            fltAggr="2"
            fsmDescr=""
            fsmPrev="PsuPolicyConfigSuccess"
            fsmProgr="100"
            fsmRmtInvErrCode="none"
            fsmRmtInvErrDescr=""
            fsmRmtInvRslt=""
            fsmStageDescr=""
            fsmStamp="2009-09-13T21:34:37"
            fsmStatus="nop"
            fsmTry="0"
            id="1"
            lcTs="1969-12-31T16:00:00"
```

```
            managingInst="A"
            model="N20-C6508"
            operQualifier="fabric-conn-problem"
            operState="fabric-conn-problem"
            operability="operable"
            power="ok"
            presence="unknown"
            revision="0"
            serial="FOX1252GG84"
            thermal="ok"
            vendor="Cisco Systems Inc"
            versionHolder="yes"/>
    </outConfigs>
    <outUnresolved>
        <dn value="sys/chassis-1/blade-1/board/t-stats"/>
    </outUnresolved>
</configResolveDns>
```

configResolveParent

次に、指定された DN について管理対象オブジェクトの親を取得する例を示します。

要求構文

```
<xs:element name="configResolveParent" type="configResolveParent"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveParent" mixed="true">
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configResolveParent" type="configResolveParent"
substitutionGroup="externalMethod"/>
        <xs:complexType name="configResolveParent" mixed="true">
            <xs:all>
                <xs:element name="outConfig" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

例

要求

```
<configResolveParent
    cookie="<real_cookie>"
    inHierarchical="false"
    dn="sys/chassis-1/blade-1/adaptor-1">
</configResolveParent>
```

応答

```
<configResolveParent dn="sys/chassis-1/blade-1/adaptor-1"
    cookie="<real_cookie>"
    response="yes">
    <outConfig>
        <computeBlade
            adminPower="policy"
            adminState="in-service"
            assignedToDn=""
            association="none"
            availability="available"
            availableMemory="10240"
            chassisId="1"
            checkPoint="discovered"
            connPath="A,B"
            connStatus="A,B"
            descr=""
            diagnostics="complete"
            discovery="complete"
            dn="sys/chassis-1/blade-1"
            fltAggr="0"
            fsmDescr=""
            fsmFlags=""
            fsmPrev="DiscoverSuccess"
            fsmProgr="100"
            fsmRmtInvErrCode="none"
            fsmRmtInvErrDescr=""
            fsmRmtInvRslt=""
            fsmStageDescr=""
            fsmStamp="2009-09-23T23:44:30"
            fsmStatus="nop"
            fsmTry="0"
            intId="768052"
            lc="discovered"
            lcTs="1969-12-31T16:00:00"
            managingInst="B"
            model="N20-B6620-1"
            name=""
            numOfAdaptors="1"
            numOfCores="8"
            numOfCpus="2"
            numOfEthHostIfs="2"
            numOfFcHostIfs="0"
            numOfThreads="16"
            operPower="off"
            operQualifier=""
            operState="unassociated"
            operability="operable"
            originalUuid="e3516842-d0a4-11dd-baad-000bab01bfd6"
            presence="equipped"
            revision="0"
            serial="QCI12520024"
            slotId="1"
```

```
            totalMemory="10240"
            uuid="e3516842-d0a4-11dd-baad-000bab01bfd6"
            vendor="Cisco Systems Inc"/>
    </outConfig>
</configResolveParent>
```

configScope

次に、管理対象オブジェクトとその設定に関する詳細を返す例を示します。

要求構文

```
<xs:element name="configScope" type="configScope" substitutionGroup="externalMethod"/>
        <xs:complexType name="configScope" mixed="true">
            <xs:all>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inClass" type="namingClassId"/>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inRecursive">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>

            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="configScope" type="configScope" substitutionGroup="externalMethod"/>
        <xs:complexType name="configScope" mixed="true"> <xs:all>
                <xs:element name="outConfigs" type="configSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
```

```
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

例

要求

```
<configScope
    dn="org-root"
    cookie="<real_cookie>"
    inClass="orgOrgRes"
    inHierarchical="false"
    inRecursive="false">
    <inFilter>
    </inFilter>
</configScope>
```

応答

```
<configScope dn="org-root"
    cookie="<real_cookie>"
    commCookie="2/15/0/2a53"
    srcExtSys="10.193.33.120"
    destExtSys="10.193.33.120"
    srcSvc="sam_extXMLApi"
    destSvc="service-reg_dme"
    response="yes">
    <outConfigs>
        <orgOrgCaps
            dn="org-root/org-caps"
            org="512"
            tenant="64"/>
        <orgOrgCounts
            dn="org-root/org-counter"
            org="36"
            tenant="7"/>
    </outConfigs>
</configScope>
```

eventSendHeartbeat

次に、現在のセッションがまだアクティブであることを示すイベントを送信する例を示します。

要求構文

```
<xs:element name="eventSendHeartbeat" type="eventSendHeartbeat"
substitutionGroup="externalMethod"/>
        <xs:complexType name="eventSendHeartbeat" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="eventSendHeartbeat" type="eventSendHeartbeat"
substitutionGroup="externalMethod"/>
        <xs:complexType name="eventSendHeartbeat" mixed="true">
            <xs:attribute name="outSystemTime" type="dateTime"/>
```

```
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

クライアント アプリケーションが eventSubscribeApps または eventSubscribe を使用してイベントをサブスクライブする場合、Cisco VNMC は eventSendHeartbeat を定期的（デフォルトでは 120 秒）に送信します。

応答

```
<eventSendHeartbeat cookie="0/0/0/2a76"
    commCookie=""
    srcExtSys="0.0.0.0"
    destExtSys="0.0.0.0"
    srcSvc=""
    destSvc=""
    response="yes"
    outSystemTime="2010-11-12T20:38:19.630">
</eventSendHeartbeat>
```

eventSubscribe

次に、アクティビティに対するサブスクライブ要求を送信する例を示します。

要求構文

```
<xs:element name="eventSubscribe" type="eventSubscribe"
substitutionGroup="externalMethod"/>
        <xs:complexType name="eventSubscribe" mixed="true">
            <xs:all>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="eventSubscribe" type="eventSubscribe"
substitutionGroup="externalMethod"/>
        <xs:complexType name="eventSubscribe" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<eventSubscribe
    cookie="<real_cookie>">
    <inFilter>
    </inFilter>
</eventSubscribe>
```

応答

```
NO RESPONSE OR ACKNOWELDGEMENT.
```

eventSubscribeApps

次に、指定されたアプリケーションのアクティビティに対するサブスクライブ要求の例を示します。クライアント アプリケーションは、さまざまなアプリケーションからイベントを受け取るために Cisco VNMC システムをサブスクライブできます。eventApplication では、ip はアプリケーション（DME）が実行されている VM の IP アドレスです。クライアント アプリケーションは Cisco VSG からのイベントも受け取るようサブスクライブできます。この場合、*ip* は Cisco VSG の IP アドレスである必要があり、タイプは管理対象エンドポイントです。

要求構文

```
<xs:element name="eventSubscribeApps" type="eventSubscribeApps"
substitutionGroup="externalMethod"/>
        <xs:complexType name="eventSubscribeApps" mixed="true">
            <xs:all>
                <xs:element name="inAppList" type="configSet" minOccurs="0"/>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="eventSubscribeApps" type="eventSubscribeApps"
substitutionGroup="externalMethod"/>
        <xs:complexType name="eventSubscribeApps" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<eventSubscribeApps
    cookie="<real_cookie>"
    commCookie=""
    srcExtSys="0.0.0.0"
    destExtSys="0.0.0.0"
```

```
    srcSvc=""
    destSvc="">
    <inAppList>
        <eventApplication
            ip="10.193.33.101"
            type="service-reg"/>
            <eventApplication
            ip="10.193.33.101"
            type="policy-mgr"/>
        <eventApplication
            ip="10.193.33.101"
            type="mgmt-controller"/>
    </inAppList>
    <inFilter>
    </inFilter>
</eventSubscribeApps>
```

応答

```
IF SUCCESSFUL, NO RESPONSE OR ACKNOWELDGEMENT.
```

faultAckFault

次に、障害が記録されたときに確認応答を送信する例を示します。

要求構文

```
<xs:element name="faultAckFault" type="faultAckFault" substitutionGroup="externalMethod"/>
        <xs:complexType name="faultAckFault" mixed="true">
            <xs:attribute name="inId" type="xs:unsignedLong"/>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="faultAckFault" type="faultAckFault" substitutionGroup="externalMethod"/>
        <xs:complexType name="faultAckFault" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<faultAckFault
    inHierarchical="false"
    cookie="<real_cookie>"
    inId="10120" />
```

応答

```
<faultAckFault
    cookie="<real_cookie>"
```

```
    commCookie="5/15/0/6c"
    srcExtSys="10.193.33.214"
    destExtSys="10.193.33.214"
    srcSvc="sam_extXMLApi"
    destSvc="resource-mgr_dme"
    response="yes">
</faultAckFault>
```

faultAckFaults

次に、複数の障害が記録されたときに確認応答を送信する例を示します。

要求構文

```
<xs:element name="faultAckFaults" type="faultAckFaults"
substitutionGroup="externalMethod"/>
        <xs:complexType name="faultAckFaults" mixed="true">
            <xs:all>
                <xs:element name="inIds" type="idSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="faultAckFaults" type="faultAckFaults"
substitutionGroup="externalMethod"/>
        <xs:complexType name="faultAckFaults" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<faultAckFaults
    cookie="<real_cookie>">
    <inIds>
        <id value="10656"/>
        <id value="10660"/>
    </inIds>
</faultAckFaults>
```

応答

```
<faultAckFaults
    cookie="<real_cookie>"
    commCookie="11/15/0/505"
    srcExtSys="10.193.34.70"
    destExtSys="10.193.34.70"
    srcSvc="sam_extXMLApi"
    destSvc="mgmt-controller_dme"
    response="yes">
```

```
</faultAckFaults>
```

faultResolveFault

次に、障害が解決されたときに応答を送信する例を示します。

要求構文

```
<xs:element name="faultResolveFault" type="faultResolveFault"
substitutionGroup="externalMethod"/>
        <xs:complexType name="faultResolveFault" mixed="true">
            <xs:attribute name="inId" type="xs:unsignedLong"/>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="faultResolveFault" type="faultResolveFault"
substitutionGroup="externalMethod"/>
        <xs:complexType name="faultResolveFault" mixed="true">
            <xs:all>
                <xs:element name="outFault" type="configConfig" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<faultResolveFault
    inHierarchical="false"
    cookie="<real_cookie>"
    inId="10120" />
```

応答

```
<faultResolveFault
    cookie="<real_cookie>"
    commCookie="5/15/0/6a"
    srcExtSys="10.193.33.214"
    destExtSys="10.193.33.214"
    srcSvc="sam_extXMLApi"
    destSvc="resource-mgr_dme"
    response="yes">
    <outFault>
        <faultInst
            ack="yes"
            cause="empty-pool"
            changeSet=""
            code="F0135"
            created="2010-11-19T11:02:41.568"
            descr="Virtual Security Gateway pool default is empty"
```

```
            dn="org-root/fwpool-default/fault-F0135"
            highestSeverity="minor"
            id="10120"
            lastTransition="2010-11-19T11:02:41.568"
            lc=""
            occur="1"
            origSeverity="minor"
            prevSeverity="minor"
            rule="fw-pool-empty"
            severity="minor"
            tags=""
            type="equipment"/>
    </outFault>
</faultResolveFault>
```

loggingSyncOcns

次に、DME からイベント ID を取得する例を示します。

要求構文

```
<xs:element name="loggingSyncOcns" type="loggingSyncOcns"
substitutionGroup="externalMethod"/>
        <xs:complexType name="loggingSyncOcns" mixed="true">
            <xs:attribute name="inFromOrZero" type="xs:unsignedLong"/>
            <xs:attribute name="inToOrZero" type="xs:unsignedLong"/>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="loggingSyncOcns" type="loggingSyncOcns"
substitutionGroup="externalMethod"/>
        <xs:complexType name="loggingSyncOcns" mixed="true">
            <xs:all>
                <xs:element name="outStimuli" type="MethodSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<loggingSyncOcns
    cookie="<real_cookie>"
    inFromOrZero="0"
    inToOrZero="4567000"/>
```

応答

イベント ID のリスト。

methodVessel

次に、単一のペイロードに複数の configMoChangeEvent を含むバッチ イベント通知の例を示します。

要求構文

```
<xs:element name="methodVessel" type="methodVessel" substitutionGroup="externalMethod"/>
        <xs:complexType name="methodVessel" mixed="true">
            <xs:all>
                <xs:element name="inStimuli" type="MethodSet" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="methodVessel" type="methodVessel" substitutionGroup="externalMethod"/>
        <xs:complexType name="methodVessel" mixed="true">
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
        </xs:complexType>
```

例

要求

```
<methodVessel
    cookie=""
    commCookie=""
    srcExtSys="10.193.77.66"
    destExtSys="10.193.77.66"
    srcSvc="resource-mgr_dme"
    destSvc="sam_extXMLApi">
    <inStimuli>
        <configMoChangeEvent
            cookie=""
            commCookie=""
            srcExtSys="10.193.77.66"
            destExtSys="10.193.77.66"
            srcSvc="resource-mgr_dme"
            destSvc="sam_extXMLApi"
            inEid="434920">
        <inConfig>
            <resstateNormalizedHealthState
                dn="fw/inst-1010/normalized-health-state"
                operState="config"
                status="modified"/>
        </inConfig>
    </configMoChangeEvent>

    <configMoChangeEvent
        cookie=""
        commCookie=""
        srcExtSys="10.193.77.66"
        destExtSys="10.193.77.66"
```

```
        srcSvc="resource-mgr_dme"
        destSvc="sam_extXMLApi"
        inEid="434921">
        <inConfig>
            <fwComputeFirewall
                configState="applying"
                dn="org-root/org-tenant_d3337/org-dc1/cfw-vsn-d3340"
                status="modified"/>
        </inConfig>
    </configMoChangeEvent>

    <configMoChangeEvent
        cookie=""
        commCookie=""
        srcExtSys="10.193.77.66"
        destExtSys="10.193.77.66"
        srcSvc="resource-mgr_dme"
        destSvc="sam_extXMLApi"
        inEid="434922">
        <inConfig>
            <faultInst
                dn="org-root/org-tenant_d3337/org-dc1/cfw-vsn-d3340/fault-F0117"
                lc="flapping,soaking-clear"
                status="modified"/>
        </inConfig>
    </configMoChangeEvent>
    </inStimuli>
</methodVessel>
Event (10:32:2:807):

<methodVessel
cookie=""
commCookie=""
srcExtSys="10.193.77.66"
destExtSys="10.193.77.66"
srcSvc="resource-mgr_dme"
destSvc="sam_extXMLApi">
  <inStimuli>

    <configMoChangeEvent
        cookie=""
        commCookie=""
        srcExtSys="10.193.77.66"
        destExtSys="10.193.77.66"
        srcSvc="resource-mgr_dme"
        destSvc="sam_extXMLApi"
        inEid="434923">
    <inConfig>
        <eventLog
            dn="event-log"
            size="1412"
            status="modified"/>
    </inConfig>
    </configMoChangeEvent>

    <configMoChangeEvent
        cookie=""
        commCookie=""
        srcExtSys="10.193.77.66"
        destExtSys="10.193.77.66"
        srcSvc="resource-mgr_dme"
        destSvc="sam_extXMLApi"
        inEid="434924">
        <inConfig>
```

```
            <eventRecord
                affected="fw/inst-1009"
                cause="transition"
                changeSet=""
                code="E4194509"
                created="2010-11-19T18:32:02.622"
                descr="[FSM:STAGE:REMOTE-ERROR]: WRONG STATE:Result: not-applicable Code:
unspecified Message: (sam:dme:FwInstanceAssociate:configPA)"
                dn="event-log/83081"
                id="83081"
                ind="state-transition"
                severity="info"
                status="created"
                trig="special"
                txId="219376"
                user="internal"/>
        </inConfig>
    </configMoChangeEvent>
    </inStimuli>
</methodVessel>
```

応答

```
<methodVessel
    cookie=""
    commCookie=""
    srcExtSys="10.193.77.66"
    destExtSys="10.193.77.66"
    srcSvc="resource-mgr_dme"
    destSvc="sam_extXMLApi">
    <inStimuli>
        <configMoChangeEvent
            cookie=""
            commCookie=""
            srcExtSys="10.193.77.66"
            destExtSys="10.193.77.66"
            srcSvc="resource-mgr_dme"
            destSvc="sam_extXMLApi"
            inEid="434925">
            <inConfig>
                <resstateNormalizedHealthState
                dn="fw/inst-1010/normalized-health-state"
                operState="config-failure"
                status="modified"/>
            </inConfig>
        </configMoChangeEvent>

    <configMoChangeEvent
        cookie=""
        commCookie=""
        srcExtSys="10.193.77.66"
        destExtSys="10.193.77.66"
        srcSvc="resource-mgr_dme"
        destSvc="sam_extXMLApi"
        inEid="434926">
        <inConfig>
            <fwComputeFirewall
            configState="failed-to-apply"
            dn="org-root/org-tenant_d3337/org-dc1/cfw-vsn-d3340"
            status="modified"/>
        </inConfig>
    </configMoChangeEvent>

    <configMoChangeEvent
```

```
        cookie=""
        commCookie=""
        srcExtSys="10.193.77.66"
        destExtSys="10.193.77.66"
        srcSvc="resource-mgr_dme"
        destSvc="sam_extXMLApi"
        inEid="434927">
        <inConfig>
            <faultInst
            dn="org-root/org-tenant_d3337/org-dc1/cfw-vsn-d3340/fault-F0117"
            lastTransition="2010-11-19T18:32:02.680"
            lc="flapping"
            status="modified"/>
        </inConfig>
    </configMoChangeEvent>
    </inStimuli>
</methodVessel>
```

orgResolveElements

この例では、指定された DN 内で、クエリー フィルタを満たす管理対象オブジェクトが取得され、組織（オプションでは子組織）で開始された管理対象オブジェクトが検索されます。

この DN を持つ組織がない場合は、空のマップが返されます。この DN を持つ組織がある場合は、指定されたクラスとフィルタで管理対象オブジェクトが検索されます。

inHierarchical が true の場合は、一致するすべてのオブジェクトとその子孫が返されます。false の場合は、一致するオブジェクトだけが返されます。*inSingleLevel* が true の場合は、検索が組織レベルで停止します。false の場合は、子組織を含めます。

要求構文

```
<xs:element name="orgResolveElements" type="orgResolveElements"
substitutionGroup="externalMethod"/>
        <xs:complexType name="orgResolveElements" mixed="true">
            <xs:all>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inClass" type="namingClassId"/>
            <xs:attribute name="inSingleLevel">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
```

```
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="orgResolveElements" type="orgResolveElements"
substitutionGroup="externalMethod"/>
        <xs:complexType name="orgResolveElements" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configMap" minOccurs="0">
                    <xs:unique name="unique_map_key_5">
                         <xs:selector xpath="pair"/>
                         <xs:field xpath="@key"/>
                    </xs:unique>
                </xs:element>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

例

要求

```
<orgResolveElements
    dn="org-root/org-Cola"
    cookie="<real_cookie>"
    commCookie="7/15/0/19"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    inClass="policyPolicySet"
    inSingleLevel="no"
    inHierarchical="no">
    <inFilter>
    </inFilter>
</orgResolveElements>
```

応答

```
<orgResolveElements
    dn="org-root/org-Cola"
    cookie="<real_cookie>"
    commCookie="7/15/0/19"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes"
    errorCode="0"
    errorDescr="">
    <outConfigs>
```

```
        <pair key="pset-default">
            <policyPolicySet
            adminState="enabled"
            descr="The default Policy Set"
            dn="org-root/pset-default"
            intId="10082"
            name="default"/>
        </pair>
        <pair key="pset-myPolicySet3">
            <policyPolicySet
            adminState="enabled"
            descr=""
            dn="org-root/org-Cola/pset-myPolicySet3"
            intId="12289"
            name="myPolicySet3"/>
        </pair>
        <pair key="pset-policySetSanity">
            <policyPolicySet
            adminState="enabled"
            descr=""
            dn="org-root/org-Cola/pset-policySetSanity"
            intId="24627"
            name="policySetSanity"/>
        </pair>
        <pair key="pset-pci_compliance_f">
            <policyPolicySet
            adminState="enabled"
            descr=""
            dn="org-root/pset-pci_compliance_f"
            intId="24539"
            name="pci_compliance_f"/>
        </pair>
        <pair key="pset-pci_compliance_h">
            <policyPolicySet
            adminState="enabled"
            descr=""
            dn="org-root/pset-pci_compliance_h"
            intId="24541"
            name="pci_compliance_h"/>
        </pair>
    </outConfigs>
 </orgResolveElements>
```

orgResolveInScope

次に、システムが、該当する DN を持つ組織と親組織（オプション）をルートまで再帰的に検索する例を示します。組織がない場合は、空のマップが返されます。組織がある場合は、指定されたクラスとフィルタですべてのプールが検索されます。

（注）　*inSingleLevel* が false の場合は、親組織が、ルート ディレクトリまで検索されます。

要求構文

```
<xs:element name="orgResolveInScope" type="orgResolveInScope"
substitutionGroup="externalMethod"/>
        <xs:complexType name="orgResolveInScope" mixed="true">
            <xs:all>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
```

```
            </xs:all>
            <xs:attribute name="inClass" type="namingClassId"/>
            <xs:attribute name="inSingleLevel">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="orgResolveInScope" type="orgResolveInScope"
substitutionGroup="externalMethod"/>
        <xs:complexType name="orgResolveInScope" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configMap" minOccurs="0">
                    <xs:unique name="unique_map_key_6">
                        <xs:selector xpath="pair"/>
                        <xs:field xpath="@key"/>
                    </xs:unique>
                </xs:element>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

例

要求

```
<orgResolveInScope
    cookie="<real_cookie>"
    dn="org-root/org-Cola"
    inClass="policyVirtualNetworkServiceProfile"
    inHierarchical="true"
```

```
    InSingleLevel="false" >
    <inFilter>
        <eq class="policyVirtualNetworkServiceProfile"
        property="name"
        value="spsanity" />
    </inFilter>
</orgResolveInScope>
```

応答

```
<orgResolveInScope
    dn="org-root/org-Cola"
    cookie="<real_cookie>"
    commCookie="7/15/0/1c35"
    srcExtSys="10.193.34.70"
    destExtSys="10.193.34.70"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes">
    <outConfigs>
        <pair key="vnsp-spsanity">
            <policyVirtualNetworkServiceProfile
                childAction="deleteNonPresent"
                descr=""
                dn="org-root/org-Cola/vnsp-spsanity"
                intId="82018"
                name="spsanity"
                policySetNameRef="policySetSanity"
                vnspId="41">
            <policyVnspAVPair
                childAction="deleteNonPresent"
                descr=""
                id="1"
                intId="82019"
                name=""
                rn="vnsp-avp-1">
                <policyAttributeValue
                    childAction="deleteNonPresent"
                    id="1"
                    rn="attr-val1"
                    value="DEV"/>
                <policyAttributeDesignator
                    attrName="dept"
                    childAction="deleteNonPresent"
                    rn="attr-ref"/>
                </policyVnspAVPair>
            </policyVirtualNetworkServiceProfile>
        </pair>
    </outConfigs>
</orgResolveInScope>
```

orgResolveLogicalParents

この例は、システムが、指定された DN を持つ論理的な親をルート ディレクトリまでどのように検索するかを示しています。

要求構文

```
<xs:element name="orgResolveLogicalParents" type="orgResolveLogicalParents"
substitutionGroup="externalMethod"/>
```

```
        <xs:complexType name="orgResolveLogicalParents" mixed="true">
            <xs:attribute name="inSingleLevel">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="orgResolveLogicalParents" type="orgResolveLogicalParents"
substitutionGroup="externalMethod"/>
        <xs:complexType name="orgResolveLogicalParents" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configMap" minOccurs="0">
                    <xs:unique name="unique_map_key_7">
                        <xs:selector xpath="pair"/>
                        <xs:field xpath="@key"/>
                    </xs:unique>
                </xs:element>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

例

要求

```
<orgResolveLogicalParents
    dn="org-root/org-Cisco/org-HR/zone-clients"
    cookie="<real_cookie>"
    commCookie="7/15/0/12"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
```

```
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    inSingleLevel="no"
    inHierarchical="no">
</orgResolveLogicalParents>
```

応答

```
<orgResolveLogicalParents
    dn="org-root/org-Cisco/org-HR/zone-clients"
    cookie="<real_cookie>"
    commCookie="7/15/0/12"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="policy-mgr_dme"
    response="yes"
    errorCode="0"
    errorDescr="">
    <outConfigs>
        <pair key="org-root/org-Cisco/zone-clients">
            <policyZone
                descr=""
                dn="org-root/org-Cisco/zone-clients"
                intId="24841"
                name="clients"/>
        </pair>
    </outConfigs>
</orgResolveLogicalParents>
```

policyEstimateImpact

この例では、管理対象オブジェクト（MO）が取得され、この変更により影響を受ける MO のリストが返されます。これは、ポリシー変更（作成、変更、または削除）の影響度を推定するために使用します。

MO に影響を与える可能性があるポリシー変更は次のとおりです。

- 名前参照が変更されています（つまり、解決された名前が変更されます）。
- 削除時に、このオブジェクトを参照する別のオブジェクト参照を、ツリーの上部にある同じ名前の異なるオブジェクト（または、デフォルトのオブジェクト）に解決できるようになりました。
- 作成時に、オブジェクト参照は新しく作成されたオブジェクトを参照できます。応答では、outDeleteAllowed パラメータは、指定されたオブジェクトを削除できるかどうかを示します。削除できる場合は、このオブジェクトを参照する名前参照はツリーの上部の別のオブジェクトを参照します。
- 指定されたオブジェクトがオブジェクト参照の参照されたオブジェクトであり、これをオーバーライドする他のオブジェクトがツリーにない場合は、指定されたオブジェクトを削除できません。

要求構文

```
<!--
           - Method: policy:EstimateImpact
         -->
        <xs:element name="policyEstimateImpact" type="policyEstimateImpact"
substitutionGroup="externalMethod"/>

        <xs:complexType name="policyEstimateImpact" mixed="true">
```

```
            <xs:all>
                <xs:element name="inConfig" type="configConfig" minOccurs="0"/>
            </xs:all>

            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>

        </xs:complexType>
```

応答構文

```
<!--
           - Method: policyEstimateImpact
         -->
        <xs:element name="policyEstimateImpact" type="policyEstimateImpact"
substitutionGroup="externalMethod"/>

        <xs:complexType name="policyEstimateImpact" mixed="true">
            <xs:all>
                <xs:element name="outImpactedMoSet" type="configMap" minOccurs="0">
                    <xs:unique name="unique_map_key_8">
                         <xs:selector xpath="pair"/>
                         <xs:field xpath="@key"/>
                    </xs:unique>
                </xs:element>
            </xs:all>

            <xs:attribute name="outDeleteAllowed">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                         <xs:simpleType>
                             <xs:restriction base="xs:string">
                                 <xs:enumeration value="no"/>
                                 <xs:enumeration value="yes"/>
                             </xs:restriction>
                         </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>

            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>

        </xs:complexType>
```

例

要求

```
<policyEstimateImpact
   cookie="1309385844/5d9899e6-5b5b-4ab0-b12c-cb6e50b5a7db"
   commCookie="7/12/0/1d18"
   srcExtSys="10.193.33.221"
   destExtSys="10.193.33.221"
   srcSvc="sam_extXMLApi"
   destSvc="policy-mgr_dme">
     <inConfig>
```

```
    <policyRuleBasedPolicy
    dn="org-root/org-tenant0/pset-myPolicySet1"
    status="deleted,modified"/>
    </inConfig>
  </policyEstimateImpact>
```

応答

```
<policyEstimateImpact
 cookie="1309385844/5d9899e6-5b5b-4ab0-b12c-cb6e50b5a7db"
 commCookie="7/12/0/1d18"
 srcExtSys="10.193.33.221"
 destExtSys="10.193.33.221"
 srcSvc="sam_extXMLApi"
 destSvc="policy-mgr_dme"
 response="yes"
 errorCode="0"
 errorDescr=""
 outDeleteAllowed="yes">
   <outImpactedMoSet>
 <pair key="org-root/org-tenant0/vnsp-VNSP1">
   <policyVirtualNetworkServiceProfile
   descr=""
   dn="org-root/org-tenant0/vnsp-VNSP1"
   intId="40930"
   name="VNSP1"
   policySetNameRef="myPolicySet1"
   vnspId="17"/>
 </pair>
   </outImpactedMoSet>
 </policyEstimateImpact>
```

poolResolveInScope

次に、システムが、該当する DN があるプールと親プール（オプション）をルートまで再帰的に検索する例を示します。プールがない場合は、空のマップが返されます。プールがある場合は、指定されたクラスとフィルタですべてのプールが検索されます。

（注）　*inSingleLevel* が false の場合は、親プールがルート ディレクトリまで検索されます。

要求構文

```
<xs:element name="poolResolveInScope" type="poolResolveInScope"
substitutionGroup="externalMethod"/>
        <xs:complexType name="poolResolveInScope" mixed="true">
            <xs:all>
                <xs:element name="inFilter" type="filterFilter" minOccurs="0"/>
            </xs:all>
            <xs:attribute name="inClass" type="namingClassId"/>
            <xs:attribute name="inSingleLevel">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
```

```
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="inHierarchical">
                <xs:simpleType>
                    <xs:union memberTypes="xs:boolean">
                        <xs:simpleType>
                            <xs:restriction base="xs:string">
                                <xs:enumeration value="no"/>
                                <xs:enumeration value="yes"/>
                            </xs:restriction>
                        </xs:simpleType>
                    </xs:union>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

応答構文

```
<xs:element name="poolResolveInScope" type="poolResolveInScope"
substitutionGroup="externalMethod"/>
        <xs:complexType name="poolResolveInScope" mixed="true">
            <xs:all>
                <xs:element name="outConfigs" type="configMap" minOccurs="0">
                    <xs:unique name="unique_map_key_9">
                        <xs:selector xpath="pair"/>
                        <xs:field xpath="@key"/>
                    </xs:unique>
                </xs:element>
            </xs:all>
            <xs:attribute name="cookie" type="xs:string"/>
            <xs:attribute name="response" type="YesOrNo"/>
            <xs:attribute name="errorCode" type="xs:unsignedInt"/>
            <xs:attribute name="errorDescr" type="xs:string"/>
            <xs:attribute name="invocationResult" type="xs:string"/>
            <xs:attribute name="dn" type="referenceObject"/>
        </xs:complexType>
```

例

要求

```
<poolResolveInScope
    dn="org-root/org-Cisco"
    cookie="<real_cookie>"
    class=fwPool />
```

応答

```
<poolResolveInScope
    dn="org-root/org-Cisco"
    cookie="<real_cookie>"
    commCookie="5/15/0/5bf"
    srcExtSys="10.193.33.221"
    destExtSys="10.193.33.221"
    srcSvc="sam_extXMLApi"
    destSvc="resource-mgr_dme"
    response="yes">
    <outConfigs>
```

```
        <pair key="fwpool-default">
            <fwPool
                assigned="0"
                descr="Default Pool of Virtual Security Gateway resources"
                dn="org-root/fwpool-default"
                fltAggr="65536"
                id="1"
                intId="10065"
                name="default"
                size="0"/>
        </pair>
</poolResolveInScope>
```

INDEX

ま