CHAPTER 2

Cisco VNMC XML API の使用

この章の内容は、次のとおりです。

メソッドおよびフィルタ

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 では、Linux POST コマンドで HTTPS を使用して Cisco VNMC に接続することにより認証要求を送信しています。

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

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.3-23）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.3-20）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.2-8）
- 「Property フィルタ」（P.2-8）
- 「Composite フィルタ」（P.2-11）
- 「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

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

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

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」（P.2-25）
- 「ファイアウォール インスタンスの問い合わせ」（P.2-26）
- 「ファイアウォール インスタンスの関連付け」（P.2-27）

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>
```