



Cisco IMC XML API メソッドの説明

この章は、次の内容で構成されています。

- [aaaGetComputeAuthTokens](#) (1 ページ)
- [aaaKeepAlive](#) (3 ページ)
- [aaaLogin](#) (3 ページ)
- [aaaLogout](#) (5 ページ)
- [aaaRefresh](#) (6 ページ)
- [configConfMo](#) (8 ページ)
- [configConfMos](#) (9 ページ)
- [configResolveChildren](#) (12 ページ)
- [configResolveClass](#) (13 ページ)
- [configResolveDn](#) (14 ページ)
- [configResolveParent](#) (15 ページ)
- [eventSubscribe](#) (16 ページ)
- [eventUnsubscribe](#) (17 ページ)

aaaGetComputeAuthTokens

`aaaGetComputeAuthTokens` メソッドは、KVM を起動するために使用する認証トークンを返します。このメソッドにより、60 秒間有効な 2 つの一時認証トークンが生成されます。1 つめのトークンは KVM ユーザー名、2 つめのトークンはパスワードです。これらの認証トークンをクレデンシャルとして使用することで、Java Network Launching Protocol (JNLP) ファイルをダウンロード可能な URL にアクセスできます。KVM セッションを開始するには、URL から JNLP ファイルをダウンロードして、それを起動します。



- (注)
- Cisco IMC で vKVM オプションが無効にされている場合、トークンを取得することはできません。
 - 認証トークンを取得するには、Cisco IMC に対するユーザ権限または管理者権限が必要です。読み取り専用権限を使用してトークンを取得することはできません。
 - 認証トークンは 60 秒後に期限切れになります。60 秒が経過すると、トークンを使用して URL にアクセスできなくなります。60 秒が経過してからアクセスしようとすると、ログインが失敗し、認証エラーまたはタイムアウト メッセージを受け取ります。

要求構文

```
<xs:element name="aaaGetComputeAuthTokens" type="aaaGetComputeAuthTokens"
substitutionGroup="externalMethod"/>
  <xs:complexType name="aaaGetComputeAuthTokens" mixed="true">
    <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
    <xs:attribute name="response" type="YesOrNo"/>
  </xs:complexType>
```

応答構文

```
<xs:element name="aaaGetComputeAuthTokens" type="aaaGetComputeAuthTokens"
substitutionGroup="externalMethod"/>
  <xs:complexType name="aaaGetComputeAuthTokens" mixed="true">
    <xs:attribute name="cookie" type="xs:string"/>
    <xs:attribute name="response" type="YesOrNo"/>
    <xs:attribute name="outTokens">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="0"/>
          <xs:maxLength value="510"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>
```

例

要求:

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

応答:

```
<aaaGetComputeAuthTokens cookie="<real_cookie>" outTokens="1804289383,846930886"
response="yes"> </aaaGetComputeAuthTokens>
```

aaaKeepAlive

aaaKeepAlive メソッドは、メソッド呼び出し後に同じ Cookie を使用して、デフォルトのセッション時間が経過するまでセッションをアクティブなままにします。

要求構文

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

  <xs:complexType name="aaaKeepAlive" mixed="true">
    <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
    <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

リクエスト

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

レスポンス

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

aaaLogin

aaaLogin メソッドはログインプロセスで、セッションを開始するために必要です。この動作は、クライアントと Cisco IMC の間の HTTP（または HTTPS）セッションを確立します。



(注) パスワードの有効期限機能が有効になっている場合、aaaLogin API は XML API 応答でパスワードの有効期限を示します。

要求構文

```

<xs:element name="aaaLogin" type="aaaLogin" substitutionGroup="externalMethod"/>
  <xs:complexType name="aaaLogin" mixed="true">
    <xs:attribute name="inName" use="required">
      <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" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="0" />
          <xs:maxLength value="510" />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="cookie" type="stringMin0Max47" />
    <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="(read-only|admin|user){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="0" />
          <xs:maxLength value="32" />
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>

```

```

        </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

リクエスト

```
<aaaLogin inName='admin' inPassword='password'/>
```

レスポンス

```
<aaaLogin cookie="" response="yes" outCookie="<real_cookie>" outRefreshPeriod="600"
outPriv="admin" outSessionId="17" outVersion="3.0(0.149)"/> </aaaLogin>
```

aaaLogout

aaaLogout メソッドは、入力としてセッションの Cookie を渡すことによって Web セッションを閉じるプロセスです。これは自動では行われません。ユーザはセッションを終了するために、aaaLogout メソッドを明示的に呼び出す必要があります。

要求構文

```

<xs:element name="aaaLogout" type="aaaLogout" substitutionGroup="externalMethod"/>
  <xs:complexType name="aaaLogout" mixed="true">
    <xs:attribute name="inCookie" type="stringMin0Max47" use="required"/>
    <xs:attribute name="cookie" type="stringMin0Max47"/>
    <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:complexType>

```

```

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

```

例

リクエスト

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

レスポンス

```
<aaaLogout cookie="<real_cookie>" response="yes" outStatus="success"> </aaaLogout>
```

aaaRefresh

aaaRefresh メソッドは、ユーザ アクティビティによってセッションをアクティブなままにします（デフォルトのセッション期間中）。デフォルトでは、アクティビティがない時点から 600 秒カウントダウンされます。600 秒を超えると、Cisco IMC はスリープモードに入ります。戻るためにはサインインが必要で、その後カウントダウンが再開されます。これは同じセッション ID を引き続き使用します。



- (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。
 - パスワードの有効期限機能が有効になっている場合、aaaRefresh API は XML API レスポンスでパスワードの有効期限を示します。

リクエスト構文

```

<xs:element name="aaaRefresh" type="aaaRefresh" substitutionGroup="externalMethod"/>
  <xs:complexType name="aaaRefresh" mixed="true">
    <xs:attribute name="inName" use="required">
      <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" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:minLength value="0"/>
          <xs:maxLength value="510"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
  </xs:complexType>

```

```

        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="inCookie" type="stringMin0Max47" use="required"/>
    <xs:attribute name="cookie" type="stringMin0Max47"/>
    <xs:attribute name="response" type="YesOrNo"/>
</xs:complexType>

```

レスポンス構文

```

<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="(read-only|admin|user){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

リクエスト

```

<aaaRefresh
  cookie="<real_cookie>"
  inCookie="<real_cookie>"
  inName='admin'
  inPassword='password'>
</aaaRefresh>

```

```

<aaaRefresh cookie="<real_cookie>" inCookie="<real_cookie>" inName="admin"
inPassword="password">

```

```
</aaaRefresh>
```

レスポンス

```
<aaaRefresh
  cookie="<real_cookie>"
  response="yes"
  outCookie="<real_cookie>"
  outRefreshPeriod="600"
  outPriv="admin">
</aaaRefresh>
```

configConfMo

configConfMo メソッドは1つのサブツリーで指定の管理対象オブジェクトを設定します (DN など)。

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="1"/>
  </xs:all>
  <xs:attribute name="inHierarchical">
    <xs:simpleType>
      <xs:union memberTypes="xs:boolean YesOrNo"/>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
  <xs:attribute name="response" type="YesOrNo"/>
  <xs:attribute name="dn" type="referenceObject" use="required"/>
</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

リクエスト

```
<configConfMo
  cookie="<real_cookie>"
  dn='sys/rack-unit-1/locator-led'>
  <inConfig>
  <equipmentLocatorLed
    adminState='on' dn='sys/rack-unit-1/locator-led'>
  </equipmentLocatorLed>
  </inConfig>
</configConfMo>
```

レスポンス

```
<configConfMo dn="sys/rack-unit-1/locator-led"
  cookie="1461754266/2f609b81-3176-1176-8007-4cc92474a254" response="yes">
<outConfig>
<equipmentLocatorLed dn="sys/rack-unit-1/locator-led" adminState="inactive" color="unknown"
  id="1" name="" operState="off" status="modified"/></outConfig>
</configConfMo>
```

configConfMos

configConfMos メソッドは DN を使用して複数のサブツリーの管理対象オブジェクトを設定します。

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>
```

```
</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_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:complexType>
```



- (注)
- 複数の設定が失敗した場合、以前から存在する設定へのロールバックはサポートされません。
 - 出力の **operStatus** は設定が成功したかどうかを示します。これには次のステータスがあります。
 - 成功 (Success) : 複数の MO の設定が成功したことを示します。
 - 部分的成功 (Partial success) : いくつかの設定が失敗したことを示します。失敗した設定と失敗の理由が表示されます。
 - 失敗 (Failure) : すべての設定が失敗したことを示します。

例

リクエスト

```
<configConfMos
  cookie="<real_cookie>"
  <inConfigs>
    <pair key="sys/user-ext/user-13">
      <aaaUser priv="admin" id="13" name="TESTUSER13" pwd="testTEST15"
        accountStatus="active" dn="sys/user-ext/user-13"/>
    </pair>
    <pair key="sys/user-ext/user-14">
      <aaaUser priv="admin" id="14" name="TESTUSER14" pwd="testTEST15"
        accountStatus="active" dn="sys/user-ext/user-14"/>
    </pair>
    <pair key="sys/svc-ext/snmp-svc/snmpv3-user-5">
      <commSnmpUser id="5" name="testuser_5" securityLevel="authpriv"
```

```

    auth="MD5" authPwd="testtest" privacy="DES" privacyPwd="testTest5"/>
  </pair>
  <pair key="sys/svc-ext/snmp-svc/snmpv3-user-1">
    <commSnmpUser id="1" name="testuser_1" securityLevel="authpriv"
      auth="MD5" authPwd="testtest" privacy="DES" privacyPwd="testTest5"/>
  </pair>
  <pair key="sys/rack-unit-1/boot-precision">
    <lsbootDevPrecision dn="sys/rack-unit-1/boot-precision"
      rebootOnUpdate="no" status="modified">
      <lsbootHdd name="testhdd_2" order="2"/>
      <lsbootHdd name="testhdd_1" order="1"/>
    </lsbootDevPrecision>
  </pair>
</inConfigs>
</configConfMos>

```

レスポンス

```

<configConfMos
  cookie="<real_cookie>"
  <outConfigs>
    <pair key="sys/user-ext/user-13">
      <aaaUser id="13" accountStatus="active" name="TESTUSER13"
        priv="admin" pwd="" adminAction="no-op"
        dn="sys/user-ext/user-13" status="modified"/>
    </pair>
    <pair key="sys/user-ext/user-14">
      <aaaUser id="14" accountStatus="inactive" name="" priv="" pwd=""
        adminAction="no-op" dn="sys/user-ext/user-14" status="modified"/>
    </pair>
    <pair key="sys/svc-ext/snmp-svc/snmpv3-user-5">
      <commSnmpUser id="5" name="testuser_5" securityLevel="authpriv"
        auth="MD5" authPwd="" privacy="DES" privacyPwd=""
        adminAction="no-op" dn="sys/svc-ext/snmp-svc/snmpv3-user-5"
        status="modified"/>
    </pair>
    <pair key="sys/svc-ext/snmp-svc/snmpv3-user-1">
      <commSnmpUser id="1" name="" securityLevel="" auth="" authPwd=""
        privacy="" privacyPwd="" adminAction="no-op"
        dn="sys/svc-ext/snmp-svc/snmpv3-user-1" status="modified"/>
    </pair>
    <pair key="sys/rack-unit-1/boot-precision">
      <lsbootDevPrecision dn="sys/rack-unit-1/boot-precision"
        name="boot-precision" purpose="operational" rebootOnUpdate="no"
        reapply="no" configuredBootMode="Legacy"
        lastConfiguredBootOrderSource="CIMC" status="modified">
        <lsbootVMedia name="CDDVD" type="VMEDIA" subtype="kvm-mapped-dvd"
          access="read-only-local" order="3" state="Disabled" rn="vm-CDDVD"
          status="modified"/>
        <lsbootHdd name="testhdd_1" type="LOCALHDD" order="1"
          state="Disabled" rn="hdd-testhdd_1" status="modified"/>
        <lsbootHdd name="testhdd_2" type="LOCALHDD" order="2"
          state="Disabled" rn="hdd-testhdd_2" status="modified"/>
        <lsbootHdd name="HDD1" type="LOCALHDD" slot="HBA" order="4"
          state="Disabled" rn="hdd-HDD1" status="modified"/>
      </lsbootDevPrecision>
    </pair>
    <operStatus>all success</operStatus>
  </outConfigs>
</configConfMos>

```

configResolveChildren

configResolveChildren メソッドは、管理情報ツリーの特定の DN 下の管理対象オブジェクトの子を取得します。

要求構文

```
<xs:element name="configResolveChildren" type="configResolveChildren"
substitutionGroup="externalMethod"/>
  <xs:complexType name="configResolveChildren" mixed="true">
    <xs:attribute name="inDn" type="referenceObject" use="required"/>
    <xs:attribute name="inHierarchical">
      <xs:simpleType>
        <xs:union memberTypes="xs:boolean YesOrNo"/>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
    <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

リクエスト

```
<configResolveChildren cookie="<real_cookie>" inHierarchical="false" inDn="sys"/>
```

レスポンス

```
<configResolveChildren cookie="1461754266/2f609b81-3176-1176-8007-4cc92474a254"
response="yes"> <outConfigs> <computeRackUnit dn="sys/rack-unit-1" adminPower="policy"
availableMemory="16384" model="UCSC-server-name" memorySpeed="1333" name="UCS server_name"
numOfAdaptors="1" numOfCores="8" numOfCoresEnabled="8" numOfCpus="2" numOfEthHostIfs="2"
numOfFcHostIfs="2" numOfThreads="8" operPower="on"
originalUuid="3FDC58B1-26CF-4CFA-BFA9-B028047280B1" presence="equipped" serverId="1"
serial="FCH1917V0P1" totalMemory="16384" usrLbl=""
```

```

uuid="3FDC58B1-26CF-4CFA-BFA9-B028047280B1" vendor="Cisco Systems Inc"
cimcResetReason="graceful-reboot " assetTag="in demo" ></computeRackUnit><aaaUserEp
dn="sys/user-ext" ></aaaUserEp><aaaLdap dn="sys/ldap-ext" adminState="disabled" basedn=""
domain="" filter="sAMAccountName" attribute="CiscoAvPair" timeout="60"
encryption="enabled" locateDirectoryUsingDNS="no" dnsDomainSource="extracted-domain"
dnsSearchDomain="" dnsSearchForest="" ldapServer1="" ldapServerPort1="389" ldapServer2=""
ldapServerPort2="389" ldapServer3="" ldapServerPort3="389" ldapServer4=""
ldapServerPort4="3268" ldapServer5="" ldapServerPort5="3268" ldapServer6=""
ldapServerPort6="3268" bindMethod="login-credentials" bindDn="" password=""
groupAuth="disabled" groupAttribute="memberOf" groupNestedSearch="128"
></aaaLdap><commSvcEp dn="sys/svc-ext" ></commSvcEp><certificateManagement
dn="sys/cert-mgmt" description="Certificate Management"
></certificateManagement><mgmtImporter dn="sys/import-config" adminState="disabled"
fsmStageDescr="" fsmRmtInvErrCode="" fsmRmtInvErrDescr="" fsmDescr="import-config"
proto="none" hostname="" remoteFile="" user="" pwd="" passphrase=""
></mgmtImporter><mgmtBackup dn="sys/export-config" adminState="disabled" fsmStageDescr=""
fsmRmtInvErrCode="" fsmRmtInvErrDescr="" fsmDescr="export-config" proto="none" hostname=""
remoteFile="" user="" pwd="" passphrase="" ></mgmtBackup><mgmtInventory dn="sys/inventory"
adminState="triggered" proto="none" hostname="" remoteFile="" user="" pwd=""
fsmStatus="COMPLETED" progress="100%" ></mgmtInventory><huuController dn="sys/huu"
description="Host Upgrade Utility (HUU)" ></huuController><iodController dn="sys/iod"
description="Non-Interactive Offline Diagnostics (IOD)" ></iodController></outConfigs>
</configResolveChildren>

```

configResolveClass

configResolveClass メソッドは、特定のクラスの要求された管理対象オブジェクトを返します。inHierarchical=true の場合、結果には子が含まれます。

要求構文

```

<xs:element name="configResolveClass" type="configResolveClass"
substitutionGroup="externalMethod"/>
  <xs:complexType name="configResolveClass" mixed="true">
    <xs:attribute name="inHierarchical">
      <xs:simpleType>
        <xs:union memberTypes="xs:boolean YesOrNo"/>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
    <xs:attribute name="response" type="YesOrNo"/>
    <xs:attribute name="classId" type="namingClassId" use="required"/>
  </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:complexType>

```

```

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

```

例

リクエスト

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

レスポンス

```

<configResolveClass cookie="<real_cookie>" response="yes" classId="topSystem">
  <outConfigs> <topSystem dn="sys" address="10.10.10.10" currentTime="Wed Apr 27 10:51:08
2016 "
    localTime="Wed Apr 27 13:51:08 2016 EAT +0300" timeZone="Africa/Addis Ababa"
mode="stand-alone"
    name="Cxxx-FCH1917V0P1" >
  </topSystem>
</outConfigs>
</configResolveClass>

```

configResolveDn

configResolveDn メソッドは、指定された DN の 1 つの管理対象オブジェクトを取得します。

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 YesOrNo"/>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
    <xs:attribute name="response" type="YesOrNo"/>
    <xs:attribute name="dn" type="referenceObject" use="required"/>
  </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:complexType>

```

```

<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

リクエスト

```

<configResolveDn cookie="1461754266/2f609b81-3176-1176-8007-4cc92474a254"
inHierarchical="false" dn="sys/rack-unit-1"/>

```

レスポンス

```

<configResolveDn cookie="<real_cookie>" response="yes" dn="sys/rack-unit-1">
<outConfig> <computeRackUnit dn="sys/rack-unit-1" adminPower="policy"
availableMemory="16384"
  model="UCSC-server-name" memorySpeed="1333" name="UCS server_name" numOfAdaptors="1"
numOfCores="8"
  numOfCoresEnabled="8" numOfCpus="2" numOfEthHostIfs="2" numOfFcHostIfs="2"
numOfThreads="8"
  operPower="on" originalUuid="3FDC58B1-26CF-4CFA-BFA9-B028047280B1" presence="equipped"

  serverId="1" serial="FCH1917V0P1" totalMemory="16384" usrLbl=""
  uuid="3FDC58B1-26CF-4CFA-BFA9-B028047280B1" vendor="Cisco Systems Inc"
  cimcResetReason="graceful-reboot" assetTag="in demo" >
</computeRackUnit>
</outConfig>
</configResolveDn>

```

configResolveParent

指定された DN について、configResolveParent メソッドは管理対象オブジェクトの親を取得します。

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 YesOrNo"/>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
    <xs:attribute name="response" type="YesOrNo"/>
    <xs:attribute name="dn" type="referenceObject" use="required"/>
  </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

リクエスト

```
<configResolveParent cookie="<real_cookie>" inHierarchical="false" dn="sys/rack-unit-1"/>
```

レスポンス

```
<configResolveParent cookie="<real_cookie>" response="yes" dn="sys/rack-unit-1">
  <outConfig>
    <topSystem dn="sys" address="10.197.125.42" currentTime="Wed Apr 27 10:53:26 2016 "
      localTime="Wed Apr 27 13:53:26 2016 EAT +0300" timeZone="Africa/Addis Ababa"
      mode="stand-alone"
      name="server-FCH1917V0P1" >
    </topSystem>
  </outConfig>
</configResolveParent>
```

eventSubscribe

eventSubscribe メソッドによって、Cisco IMC が生成した非同期のシステム イベント ログ (SEL) イベントをクライアントがサブスクライブできるようになります。

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 IMC からのイベント通知を受けるように登録できます。イベントが発生したときに、Cisco IMC は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通知します。実際の変更情報だけが送信されません。オブジェクトの影響を受けない属性は含まれません。

次の例に示すように、イベントに登録するために eventSubscribe を使用します。

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


要求構文

```
<xs:element name="eventSubscribe" type="eventSubscribe"
substitutionGroup="externalMethod"/>
  <xs:complexType name="eventSubscribe" mixed="true">
    <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
    <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

リクエスト

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

レスポンス

NO RESPONSE OR ACKNOWLEDGMENT.

eventUnsubscribe

eventUnsubscribe メソッドにより、クライアントは Cisco IMC が生成する非同期システム イベント ログ (SEL) イベントへのサブスクリプションを解除し、eventUnsubscribe による イベント サブスクリプションを無効にすることができます。

イベントへのサブスクリプションを解除するには、次の例に示すように eventUnsubscribe を使用します。

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

要求構文

```
<xs:element name="eventUnsubscribe" type="eventUnsubscribe"
substitutionGroup="externalMethod"/>
  <xs:complexType name="eventUnsubscribe" mixed="true">
    <xs:attribute name="cookie" type="stringMin0Max47" use="required"/>
    <xs:attribute name="response" type="YesOrNo"/>
  </xs:complexType>
```

応答構文

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

リクエスト

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

レスポンス

NO RESPONSE OR ACKNOWLEDGMENT.

翻訳について

このドキュメントは、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。