設定パラメータ

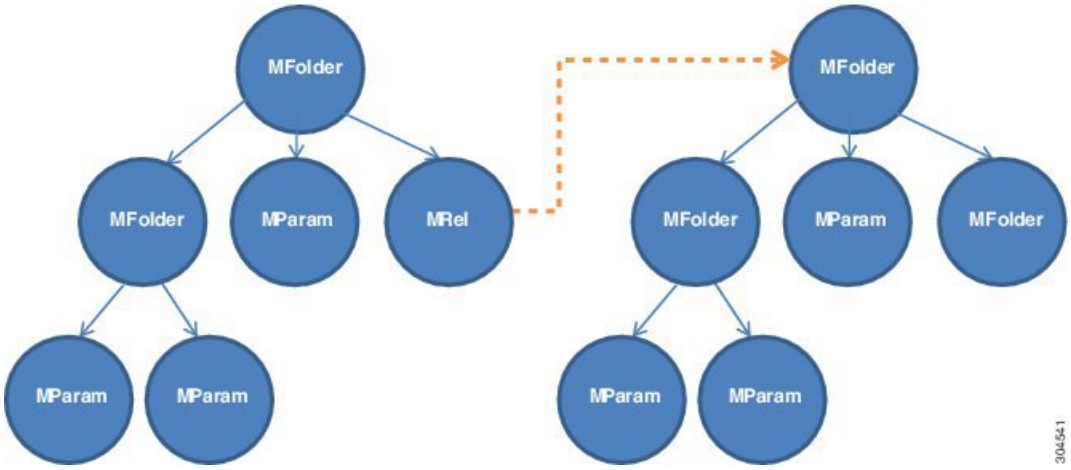
デバイスパッケージ仕様内のコンフィギュレーションパラメータ

デバイス パッケージにはサービス デバイスの仕様を示す XML ファイルが含まれます。この仕様にはデバイス情報およびサービス デバイスによって提供される各種の機能が含まれます。

デバイス仕様の一部として、このファイルにはサービス デバイスによって必要なコンフィギュレーションの宣言が含まれる必要があります。この設定は、グラフのインストール中にサービス デバイスによって提供される各種の機能を設定するために必要です。

次の図は、デバイス パッケージ内のコンフィギュレーションパラメータ階層を示しています。

図 *1:* デバイス パッケージ内のコンフィギュレーション パラメータ階層



MFolder

MFolder は、MParam および他のネストされた MFolder を含むことができるコンフィギュレーション アイテムのグループです。MFolder は次の属性を持ちます。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。濃度のデフォルト値は `1` です。濃度が `N` であれば、Application Policy Infrastructure Controller (APIC) ではコンフィギュレーション パラメータの N インスタンスの設定が可能です。
ScopedBy	パラメータ解決の範囲を指定します。ScopedBy は、APIC がコンフィギュレーション MO からパラメータを解決する場合にパラメータ値を検索する場所を決定します。 デフォルト値は `Epg` です。サポートされる値は `Tenant`、`Ap`、`Bd`、および `Epg` です。
RsConnector	コンフィギュレーション アイテムを MConn に関連付ける関係。
DevCtx	コンフィギュレーションアイテムをデバイス (LDev) 内の特定の物理デバイス (CDev) に関連付けることができます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。

MParam

MParamは、単一のコンフィギュレーションパラメータを宣言するコンフィギュレーションパラメータの基本単位です。MParam は次の属性を持ちます。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。濃度のデフォルト値は 1 です。濃度が N であれば、APIC ではコンフィギュレーション パラメータの N インスタンスの設定が可能です。
RsConnector	コンフィギュレーション アイテムを MConn に関連付ける関係。
必須	コンフィギュレーション アイテムが必須としてマークされます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。
Validation	値の検証方法を指定します。

MRel

MRel は 1 つの MFolder が別の MFolder を参照することを可能にします。MFolder 内の MRel を使用して、管理者は含む側の MFolder を、MRel 内に含まれる RsTarget 関係によって MRel からポイントされる MFolder に関連付けることができます。MRel は次の属性を持ちます。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。濃度のデフォルト値は 1 です。
RsTarget	コンフィギュレーション フォルダを別の MFolder に関連付ける関係。この関係に対する TDn の値はターゲット フォルダの DN です。
RsConnector	コンフィギュレーション アイテムを MConn に関連付ける関係。
必須	コンフィギュレーション アイテムが必須としてマークされます。

デバイス パッケージ仕様の設定スコープ

デバイス仕様ファイルで、コンフィギュレーションアイテムは異なるセクションで配置されます。

MDevCfg

MDevCfg のセクションでは、デバイスを使用するすべてのサービス グラフで共有されるデバイス レベルの設定について説明します。Application Policy Infrastructure Controller（APIC）は、この項で説明されるコンフィギュレーションアイテムを使用して作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトは、デバイスを使用しているすべてのグラフ インスタンスが削除された後にのみサービス デバイスから削除されます。

MFuncCfg

MFuncCfg は、サービス機能に対してローカルで、サービス機能に固有なコンフィギュレーションについて説明します。APIC は、このセクションで説明されるコンフィギュレーション アイテムによって作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトが作成され、サービス機能がインスタンス化または削除されたときに削除されます。

MGrpCfg

MGrpCfg は、デバイスを使用するサービス グラフのすべての機能によって共有される設定を説明します。APIC は、このセクションで説明されるコンフィギュレーション アイテムを使用して作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトは、サービス グラフからすべての機能が削除された後にサービス デバイスから削除されます。

デバイス パッケージ内のコンフィギュレーション パラメータの XML の例

次の XML の例は、デバイス パッケージ内のコンフィギュレーション パラメータを示しています。

```
<vnsMFolder key="VServer" scopedBy="epg">
    <vnsRsConnector tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-external"/>
    <vnsMParam key="vservername" description="Name of VServer" mandatory="true"/>
    <vnsMParam key="vip" description="Virtual IP"/>
    <vnsMParam key="subnet" description="Subnet IP"/>
    <vnsMParam key="port" description="Port for Virtual server"/>
    <vnsMParam key="persistencetype" description="persistencetype"/>
    <vnsMParam key="servicename" description="Service bound to this vServer"/>
    <vnsMParam key="servicetype" description="Service bound to this vServer"/>
    <vnsMParam key="clttimeout" description="Client timeout"/>
    <vnsMFolder key="VServerGlobalConfig"
      description="This references the global configuration">
        <vnsMRel key="ServiceConfig">
            <vnsRsTarget tDn="uni/infra/mDev-Acme-ADC-1.0/mDevCfg/mFolder-Service"/>
        </vnsMRel>
        <vnsMRel key="ServerConfig">
```

```
            <vnsRsTarget tDn="uni/infra/mDev-Acme-ADC-1.0/mDevCfg/mFolder-Server"/>
        </vnsMRel>
        <vnsMRel key="VipConfig">
            <vnsRsTarget
              tDn="uni/infra/mDev-Acme-ADC-1.0/mDevCfg/mFolder-Network/mFolder-vip"/>
           <vnsRsConnector tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-external"/>

        </vnsMRel>
    </vnsMFolder>
</vnsMFolder>
```

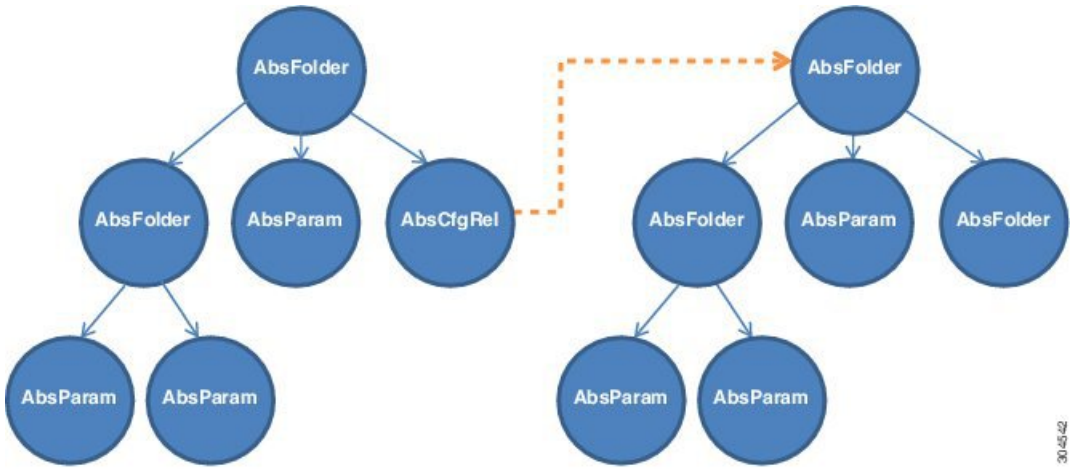
抽象機能プロファイル内のコンフィギュレーションパラメータ

抽象プロファイルを使用すると、管理者はコンフィギュレーションパラメータのデフォルト値を設定できます。抽象機能プロファイルには値を持つコンフィギュレーションパラメータが含まれます。これらの値がグラフ インスタンス作成時にデフォルト値として使用されます。

抽象機能プロファイルはサービス グラフの機能ノードに接続されます。抽象機能プロファイルで指定されたデフォルト値は、グラフのインスタンス化の際にサービスデバイスに機能をレンダリングする場合に使用されます。

次の図は、抽象機能プロファイル内のコンフィギュレーション パラメータ階層を示しています。

図 *2*:抽象機能プロファイル内部のコンフィギュレーション パラメータ階層



AbsFolder

AbsFolder は、AbsParam および他のネストされた AbsFolder を含むことができるコンフィギュレーション アイテムのグループです。デバイス パッケージ内に各 AbsFolder の MFolder が必要

です。Application Policy Infrastructure Controller（APIC） は、各 AbsFolder を検証して、パッケージ内に AbsFolder に対応する MFolder が存在することを確認します。AbsFolder には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は `1` です。
ScopedBy	パラメータ解決の範囲を指定します。ScopedBy は、APIC がコンフィギュレーション MO からパラメータを解決する場合にパラメータ値を検索する場所を決定します。 デフォルト値は `Epg` です。サポートされる値は `Tenant`、`Ap`、`Bd`、および `Epg` です。
DevCtx	コンフィギュレーション アイテムをデバイス クラスタ（LDev）内の特定の物理デバイス（CDev）に関連付けることができます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。

AbsParam

AbsParam はコンフィギュレーション パラメータの基本単位です。 AbsParam は単一のコンフィギュレーション パラメータを定義します。AbsFolder と同様、各 AbsParam に対してデバイス仕様内に対応する MFolder が存在する必要があります。APIC は仕様を検証して、パッケージ内に AbsParam に対応する MFolder が存在することを確認します。AbsParam の値は、MParam 内で指定される検証メソッドを使用して検証されます。AbsParam には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は `1` です。
必須	コンフィギュレーション アイテムが必須としてマークされます。

属性	説明
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。
Validation	コンフィギュレーション パラメータの検証に使用する検証メカニズムを指定します。

AbsRel

AbsRel は 1 つの AbsFolder が別の AbsFolder を参照することを可能にします。AbsRel には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は 1 です。
必須	コンフィギュレーション アイテムが必須としてマークされます。

抽象機能プロファイルの設定スコープ

抽象機能プロファイルでは、コンフィギュレーション パラメータはデバイス パッケージ内の場合と似た方法で構成されます。3 種類のスコープがあります。

AbsDevCfg

このセクションは、デバイスパッケージ内のデバイス レベル設定と宣言される、コンフィギュレーション アイテムのデフォルト値を提供します。コンフィギュレーション アイテムは MDevCfg で指定されます。

各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションで説明される設定は、デバイスを使用するサービス グラフで共有されます。Application Policy Infrastructure Controller（APIC）は、このセクションで説明されるコンフィギュレーション アイテムを使用して作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトは、デバイスを使用しているすべてのグラフインスタンスが削除された後にのみサービス デバイスから削除されます。

AbsGrpCfg

このセクションは、デバイスパッケージ内のデバイス レベル設定と宣言される、コンフィギュレーションアイテムのデフォルト値を提供します。コンフィギュレーションアイテムはMGrpCfgで指定されます。

各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションで説明される設定は、デバイスを使用するサービス グラフのすべての機能で共有されます。APIC は、このセクションで説明されるコンフィギュレーション アイテムを使用して作成されたコンフィギュレーションオブジェクトの参照カウントを実行します。オブジェクトは、デバイスを使用しているすべてのグラフインスタンスが削除された後にのみサービスデバイスから削除されます。

AbsFuncCfg

このセクションは、デバイスパッケージ内の機能レベル設定と宣言される、コンフィギュレーションアイテムのデフォルト値を提供します。コンフィギュレーションアイテムはMFuncCfgで指定されます。

各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションは、サービス機能にローカルな設定を説明するために使用されます。このセクションで説明されている設定は、サービス機能に固有のものです。APIC は、このセクションで説明されるコンフィギュレーション アイテムによって作成されたコンフィギュレーションオブジェクトの参照カウントを実行します。オブジェクトが作成され、サービス機能がインスタンス化または削除されたときに削除されます。

コンフィギュレーションパラメータを持つ抽象機能プロファイルに対する XML POST の例

次の XML POST の例は、コンフィギュレーション パラメータを持つ抽象機能プロファイルを示しています。

```
<vnsAbsFuncProfContr name = "NP">
    <vnsAbsFuncProfGrp name = "Grp1">
        <vnsAbsFuncProf name = "P1">
            <vnsRsProfToMFunc tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB"/>
            <vnsAbsDevCfg name="D1">
                <vnsAbsFolder key="Service" name="Service-Default" cardinality="n">
                    <vnsAbsParam name="servicetype" key="servicetype" value="TCP"/>
                    <vnsAbsParam name="serviceport" key="serviceport" value="80"/>
                    <vnsAbsParam name="maxclient" key="maxclient" value="1000"/>
                    <vnsAbsParam name="maxreq" key="maxreq" value="100"/>
                    <vnsAbsParam name="cip" key="cip" value="enable"/>
                    <vnsAbsParam name="usip" key="usip" value="enable"/>
                    <vnsAbsParam name="sp" key="sp" value=""/>
                    <vnsAbsParam name="svrtimeout" key="svrtimeout" value="60"/>
                    <vnsAbsParam name="clttimeout" key="clttimeout" value="60"/>
                    <vnsAbsParam name="cka" key="cka" value="NO"/>
                    <vnsAbsParam name="tcpb" key="tcpb" value="NO"/>
```

```
                    <vnsAbsParam name="cmp" key="cmp" value="NO"/>
                </vnsAbsFolder>
            </vnsAbsDevCfg>
            <vnsAbsFuncCfg name="SLB">
                <vnsAbsFolder key="VServer" name="VServer-Default">
                    <vnsAbsParam name="port" key="port" value="80"/>
                    <vnsAbsParam name="persistencetype" key="persistencetype"
                      value="cookie"/>
                    <vnsAbsParam name="clttimeout" key="clttimeout" value="100"/>
                    <vnsAbsParam name="servicetype" key="servicetype" value="TCP"/>
                    <vnsAbsParam name="servicename" key="servicename"/>
                </vnsAbsFolder>
            </vnsAbsFuncCfg>
        </vnsAbsFuncProf>
    </vnsAbsFuncProfGrp>
</vnsAbsFuncProfContr>
```

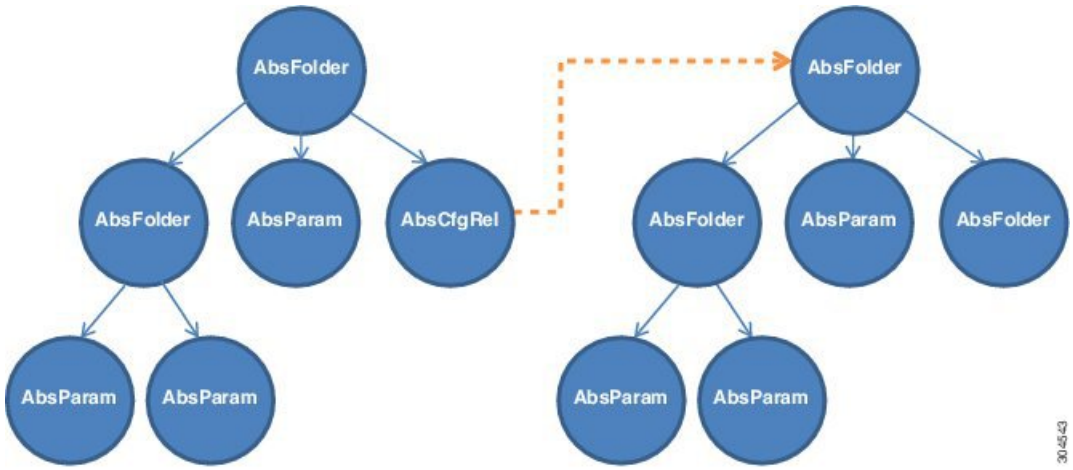
サービス グラフでの抽象機能ノード内のコンフィギュレーション パラメータ

サービス グラフ内の機能ノードを使用して、管理者はコンフィギュレーション パラメータの値を設定できます。これらの値は、グラフのインストール時に使用されます。

抽象機能ノードでは、コンフィギュレーションパラメータは抽象機能プロファイル内の場合と似た方法で構成されます。

次の図は、抽象機能ノード内のコンフィギュレーション パラメータ階層を示しています。

図 *3*:抽象機能ノード内のコンフィギュレーション パラメータ



AbsDevCfg

このセクションは、デバイスパッケージ内のデバイス レベル設定と宣言される、コンフィギュレーションアイテムのデフォルト値を提供するために使用されます。コンフィギュレーションアイテムは MDevCfg で指定されます。

これらの各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

AbsGrpCfg

このセクションは、デバイスパッケージ内のデバイス レベル設定と宣言される、コンフィギュレーションアイテムのデフォルト値を提供するために使用されます。コンフィギュレーションアイテムは MGrpCfg で指定されます。

これらの各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションで説明される設定は、デバイスを使用するサービス グラフのすべての機能で共有されます。Application Policy Infrastructure Controller（APIC）は、この項で説明されるコンフィギュレーション アイテムを使用して作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトは、サービス グラフからすべての機能が削除された後にサービス デバイスから削除されます。

AbsFuncCfg

このセクションは、デバイスパッケージ内の機能レベル設定と宣言される、コンフィギュレーション アイテムのデフォルト値を提供するために使用されます。コンフィギュレーション アイテムは MFuncCfg で指定されます。

これらの各コンフィギュレーションアイテムに対して、デバイスパッケージに同等のコンフィギュレーション アイテムが存在する必要があります。

このセクションは、サービス機能にローカルな設定を説明するために使用されます。このセクションで説明されている設定は、サービス機能に固有のものです。APIC は、このセクションで説明されるコンフィギュレーション アイテムによって作成されたコンフィギュレーション オブジェクトの参照カウントを実行します。オブジェクトが作成され、サービス機能がインスタンス化または削除されたときに削除されます。

AbsFolder

AbsFolder は、AbsParam および他のネストされた AbsFolder を含むことができるコンフィギュレーション アイテムのグループです。デバイス パッケージ内に各 AbsFolder の MFolder が必要です。APIC は、各 AbsFolder を検証して、パッケージ内に AbsFolder に対応する MFolder が存在することを確認します。AbsFolder には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。

属性	説明
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は `1` です。
ScopedBy	パラメータ解決の範囲を指定します。ScopedByは、APICがコンフィギュレーション MO からパラメータを解決する場合にパラメータ値を検索する場所を決定します。 デフォルト値は `Epg` です。サポートされる値は `Tenant`、`Ap`、`Bd`、および `Epg` です。
RsCfgToConn	コンフィギュレーション アイテムを AbsConn に関連付ける関係。
DevCtx	コンフィギュレーションアイテムをデバイス(LDev)内の特定の物理デバイス(CDev) に関連付けることができます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。

AbsParam

AbsParamはコンフィギュレーションパラメータの基本単位です。 AbsParamは単一のコンフィギュレーション パラメータを定義します。AbsFolder と同様、各 AbsParam に対してデバイス仕様内に対応する MFolder が存在する必要があります。APIC は仕様を検証して、パッケージ内に AbsParam に対応する MFolder が存在することを確認します。AbsParam の値は、MParam 内で指定される検証メソッドを使用して検証されます。AbsParam には次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は `1` です。
RsCfgToConn	コンフィギュレーション アイテムを MConn に関連付ける関係。
必須	コンフィギュレーション アイテムが必須としてマークされます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。
Validation	コンフィギュレーション パラメータの検証に使用する検証メカニズムを指定します。

AbsRel

AbsRel は 1 つの AbsFolder が別の AbsFolder を参照することを可能にします。AbsRel には次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。
説明	コンフィギュレーション アイテムを説明します。
Cardinality	コンフィギュレーション アイテムの濃度を指定します。デフォルト値は 1 です。
RsCfgToConn	コンフィギュレーション アイテムを MConn に関連付ける関係。
必須	コンフィギュレーション アイテムが必須としてマークされます。
Locked	コンフィギュレーション アイテム値がロックされます。一度ロックされると値は変更できません。

コンフィギュレーション パラメータを持つ抽象機能ノードに対する XML POST の例

次の XML POST の例は、コンフィギュレーション パラメータを持つ抽象機能ノードを示しています。

```
<vnsAbsNode name = "SLB" funcType="GoTo" >
   <vnsRsDefaultScopeToTerm tDn="uni/tn-tenant1/AbsGraph-G3/AbsTermNode-Output1/outtmnl"/>

    <vnsAbsFuncConn name = "C4" direction = "input">
        <vnsRsMConnAtt tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-external" />
    </vnsAbsFuncConn>
    <vnsAbsFuncConn name = "C5" direction = "output">
        <vnsRsMConnAtt tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB/mConn-internal" />
    </vnsAbsFuncConn>

    <vnsAbsDevCfg>
        <vnsAbsFolder key="Network" name="Network" scopedBy="epg">
            <!-- Following scopes this folder to input terminal or Src Epg -->
            <vnsRsScopeToTerm
tDn="uni/tn-tenant1/AbsGraph-G3/AbsTermNode-Output1/outtmnl"/>

            <!-- VIP address -->
            <vnsAbsFolder key="vip" name="vip" scopedBy="epg">
                <vnsAbsParam name="vipaddress" key="vipaddress" value=""/>
            </vnsAbsFolder>

            <!-- SNIP address -->
            <vnsAbsFolder key="snip" name="snip" scopedBy="epg">
                <vnsAbsParam name="snipaddress" key="snipaddress" value=""/>
```

```
            </vnsAbsFolder>

        </vnsAbsFolder>

        <vnsAbsFolder key="Service" name="Service" scopedBy="epg" cardinality="n">
            <vnsRsScopeToTerm
tDn="uni/tn-tenant1/AbsGraph-G3/AbsTermNode-Output1/outtmnl"/>
            <vnsAbsParam name="servicename" key="servicename" value=""/>
            <vnsAbsParam name="servername" key="servername" value=""/>
            <vnsAbsParam name="serveripaddress" key="serveripaddress" value=""/>
        </vnsAbsFolder>
    </vnsAbsDevCfg>

    <vnsAbsFuncCfg>
        <vnsAbsFolder key="VServer" name="VServer" scopedBy="epg">
            <vnsRsScopeToTerm
tDn="uni/tn-tenant1/AbsGraph-G3/AbsTermNode-Output1/outtmnl"/>
            <!-- Virtual Server Configuration -->
            <vnsAbsParam name="vip" key="vip" value=""/>
            <vnsAbsParam name="vservername" key="vservername" value=""/>
            <vnsAbsParam name="servicename" key="servicename"/>
            <vnsRsCfgToConn tDn="uni/tn-tenant1/AbsGraph-G3/AbsNode-Node2/AbsFConn-C4"
/>
        </vnsAbsFolder>
    </vnsAbsFuncCfg>
    <vnsRsNodeToMFunc tDn="uni/infra/mDev-Acme-ADC-1.0/mFunc-SLB"/>
</vnsAbsNode>
```

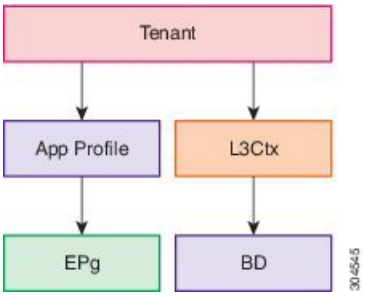
各種の設定 MO 内のコンフィギュレーション パラメータ

管理者は EPG、テナント、BD、または AP などの各種の Application Policy Infrastructure Controller（APIC） MO の一部としてサービス機能に対するコンフィギュレーション パラメータを指定できます。グラフがインスタンス化されると、APIC は各種の場所からパラメータを検索することでグラフに必要な設定を解決します。インスタンス化では、パラメータ値はデバイス スクリプトに解決され、渡されます。

各種の MO 内でコンフィギュレーション パラメータを保持できることの柔軟性により、管理者は単一のサービス グラフを設定し、グラフを異なるテナントまたはエンド ポイント グループ（EPG）に対して異なる設定で使用できます。

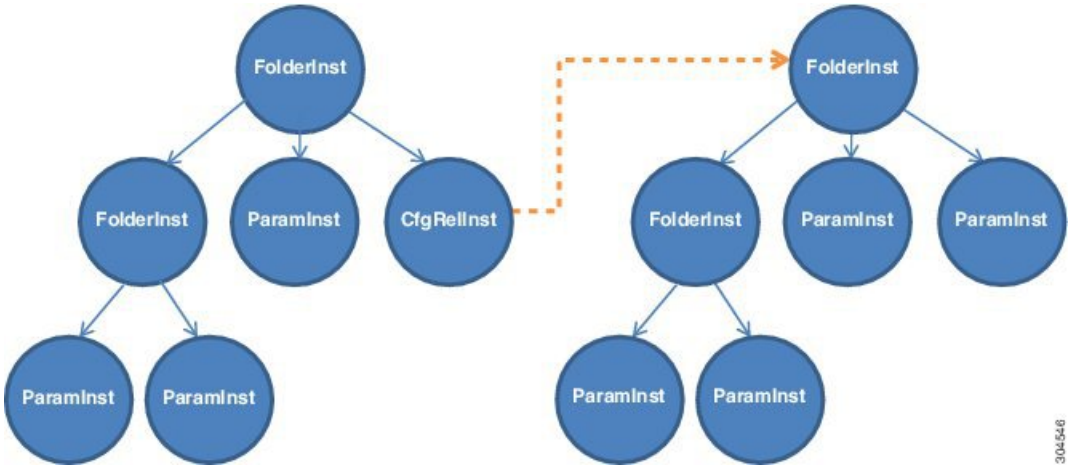
次の図は、APIC MO の階層を示しています。

図 *4 : APIC MO* の階層



次の図は、各種のコンフィギュレーション MO 内のコンフィギュレーション パラメータを示しています。

図 *5*: 各種の設定 *MO* 内のコンフィギュレーション パラメータ



FolderInst

FolderInst は、ParamInst および他のネストされた FolderInst を含むことができるコンフィギュレーション アイテムのグループです。FolderInst は次の属性を持ちます。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
ctrctNameOrLbl	パラメータの解決時に一致する FolderInst を検索します。FolderInst をパラメータ解決で使用するには、このプロパティはサービス グラフと関連付けられたコントラクト名と一致する必要があります。一致していない場合、FolderInst はスキップされ、値はこの FolderInst から使用されません。 このフィールドの値を [any] にして、この FolderInst がすべてのコントラクトで使用されるようにできます。
graphNameOrLbl	パラメータの解決時に一致する FolderInst を検索します。FolderInst をパラメータ解決で使用するには、このプロパティはサービス グラフ名と一致する必要があります。一致していない場合、FolderInst はスキップされ、値はこの FolderInst から使用されません。 この FolderInst がすべてのサービス グラフで使用されるようにするには、このフィールドの値を [any] にできます。

属性	説明
nodeNameOrLbl	パラメータの解決時に一致する FolderInst を検索します。FolderInst をパラメータ解決で使用するには、このプロパティはノード名と一致する必要があります。一致していない場合、FolderInst はスキップされ、値はこの FolderInst から使用されません。 このフィールドの値を [any] にして、この FolderInst がサービス グラフ内のすべてのノードで使用されるようにできます。

ParamInst

ParamInst はコンフィギュレーション パラメータの基本単位です。ParamInst は単一のコンフィギュレーション パラメータを定義します。FolderInst と同様、各 ParamInst に対してデバイス仕様内に対応する MParam が存在する必要があります。APIC は仕様を検証して、パッケージ内に ParamInst に対応する MParam が存在することを確認します。ParamInst の値は、対応する MParam 内で指定される検証メソッドを使用して検証されます。ParamInst には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	特定のコンフィギュレーション アイテムの値を保持します。値は MParam ではサポートされません。

CfgRelInst

CfgRelInst には、次の属性があります。

属性	説明
Key	コンフィギュレーション アイテムのタイプを定義します。キーは、デバイス パッケージで定義されており、上書きすることはできません。キーは、検証だけでなく一致基準として使用されます。
値	ターゲット FolderInst のパスを保持します。

コンフィギュレーション パラメータを持つアプリケーション EPG の XML POST の例

次の XML の例は、デバイス パッケージ内のコンフィギュレーション パラメータを示しています。

```
<fvAEPg dn="uni/tn-acme/ap-myApp/epg-app"  name="app">

    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"
key="Monitor"
```

```
      name="monitor1">
        <vnsRsFolderInstToMFolder
tDn="uni/infra/mDev-Acme-ADC-1.0/mDevCfg/mFolder-Monitor"/>
        <vnsParamInst name="weight" key="weight" value="10"/>
    </vnsFolderInst>

    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"
key="Service"
      name="Service1">
        <vnsParamInst name="servicename" key="servicename" value="crpvgrtst02-8010"/>
        <vnsParamInst name="servicetype" key="servicetype" value="TCP"/>
        <vnsParamInst name="servername" key="servername" value="s192.168.100.100"/>
        <vnsParamInst name="serveripaddress" key="serveripaddress"
value="192.168.100.100"/>
        <vnsParamInst name="serviceport" key="serviceport" value="8080"/>
        <vnsParamInst name="svrtimeout" key="svrtimeout" value="9000" />
        <vnsParamInst name="clttimeout" key="clttimeout" value="9000" />
        <vnsParamInst name="usip" key="usip" value="NO" />
        <vnsParamInst name="useproxyport" key="useproxyport" value="" />
        <vnsParamInst name="cip" key="cip" value="ENABLED" />
        <vnsParamInst name="cka" key="cka" value="NO" />
        <vnsParamInst name="sp" key="sp" value="OFF" />
        <vnsParamInst name="cmp" key="cmp" value="NO" />
        <vnsParamInst name="maxclient" key="maxclient" value="0" />
        <vnsParamInst name="maxreq" key="maxreq" value="0" />
        <vnsParamInst name="tcpb" key="tcpb" value="NO" />
        <vnsCfgRelInst name="MonitorConfig" key="MonitorConfig" targetName="monitor1"/>
    </vnsFolderInst>

    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G2" nodeNameOrLbl="any"
key="Network"
      name="Network">
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G2" nodeNameOrLbl="any"
key="vip"
          name="vip">
            <vnsParamInst name="vipaddress1" key="vipaddress" value="10.10.10.200"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G2" nodeNameOrLbl="any"
          devCtxLbl="C1" key="snip" name="snip1">
            <vnsParamInst name="snipaddress" key="snipaddress" value="192.168.1.200"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G2" nodeNameOrLbl="any"
          devCtxLbl="C2" key="snip" name="snip2">
    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
key="Network"
      name="Network">
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
key="vip"
          name="vip">
            <vnsParamInst name="vipaddress1" key="vipaddress" value="10.10.10.100"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
          devCtxLbl="C1" key="snip" name="snip1">
            <vnsParamInst name="snipaddress" key="snipaddress" value="192.168.1.100"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
          devCtxLbl="C2" key="snip" name="snip2">
            <vnsParamInst name="snipaddress" key="snipaddress" value="192.168.1.101"/>
        </vnsFolderInst>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="G1" nodeNameOrLbl="any"
          devCtxLbl="C3" key="snip" name="snip3">
            <vnsParamInst name="snipaddress" key="snipaddress" value="192.168.1.102"/>
        </vnsFolderInst>
    </vnsFolderInst>
```

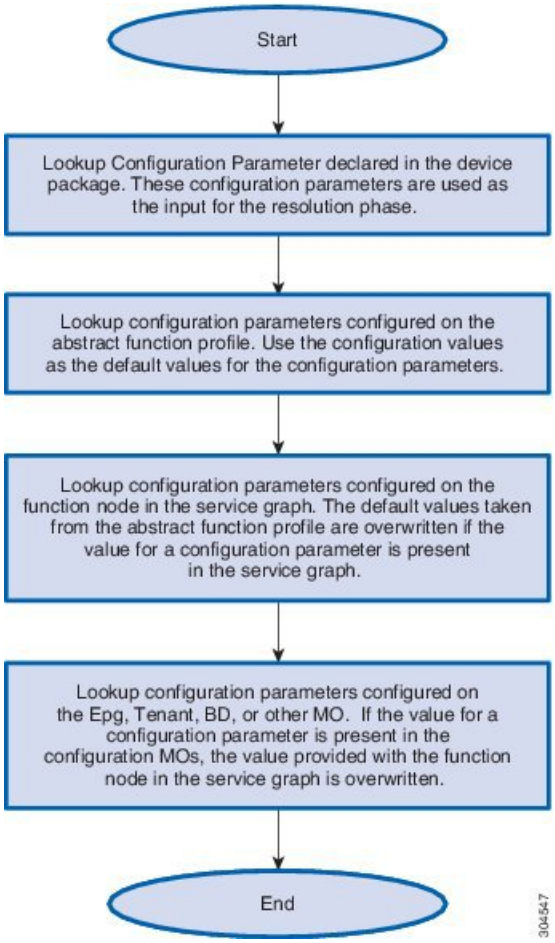
```
    <!-- SLB Configuration -->
    <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"
key="VServer"
      name="VServer">
        <!-- Virtual Server Configuration -->
        <vnsParamInst name="port" key="port" value="8010"/>
        <vnsParamInst name="vip" key="vip" value="10.10.10.100"/>
       <vnsParamInst name="vservername" key="vservername" value="crpvgrtst02-vip-8010"/>

        <vnsParamInst name="servicename" key="servicename" value="crpvgrtst02-8010"/>
        <vnsFolderInst ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any"
          key="VServerGlobalConfig" name="VServerGlobalConfig">
          <vnsCfgRelInst name="ServiceConfig" key="ServiceConfig" targetName="Service1"/>

            <vnsCfgRelInst name="VipConfig" key="VipConfig" targetName="Network/vip"/>
        </vnsFolderInst>
    </vnsFolderInst>
</fvAEPg>
```

パラメータ解決

グラフ インスタンス作成時に、Application Policy Infrastructure Controller（APIC）はサービス グラフの各機能に対してコンフィギュレーション パラメータを解決します。解決が完了すると、パラメータ値がデバイス スクリプトに渡されます。デバイス スクリプトはこれらのパラメータ値を使用してサービス アプライアンス上でサービスを設定します。

次のフロー チャートは、パラメータの解決手順について説明しています。

図 *6:* パラメータ解決



パラメータ解決時の MO の検索

Application Policy Infrastructure Controller（APIC）は、コンフィギュレーション パラメータを取得する適切なコンフィギュレーション MO の検出に 2 つの主なコンストラクトを使用します。

RsScopeToTerm

機能ノードまたは AbsFolder に対する RsScopeToTerm 関係は、グラフに対するパラメータを持つコンフィギュレーション MO と接続されるサービス グラフの端末ノードを示します。APIC は、グラフ コンフィギュレーション パラメータを検出するために、RsScopeToTerm 内の指定の端末ノードと接続されたコンフィギュレーション MO を使用します。

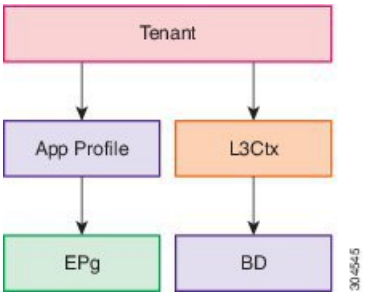
指定された RsScopeToTerm コンフィギュレーションがない場合、APIC はデフォルトでプロバイダー EPG に接続された端末を使用します。

ScopedBy 属性

ScopedBy 属性はパラメータの解決に使用する開始 MO の検出に使用されます。たとえば、scopedBy に「EPG」の値がある場合、APIC はエンドポイント グループからパラメータ解決を開始します。APIC は、階層を上ってパラメータを解決し、アプリケーション プロファイルの次にテナントに上ってコンフィギュレーション パラメータを解決します。

次の図は、APIC MO の階層を示しています。

図 *7 : APIC MO* の階層



ロールベースのアクセス コントロール ルールの拡張について

マルチテナント環境でのレイヤ 4 ～ レイヤ 7 設定では、従来のロールベースのアクセス コントロール（RBAC）ドメインとロール モデルの定義を使用してテナント管理者が作成できない特定のオブジェクトを作成するには、管理者が介入する必要がありました。Application Policy Infrastructure Controller（APIC）では、オブジェクトの作成に必要な権限をテナント管理者に付与できるように、管理情報ツリー（MIT）でRBAC権限をより詳細に指定できます。また、テナント管理者は、管理者の介入なしに、セルフサービスを介してRBACルールを作成し、テナントサブツリーの下にあるリソースの権限をシステム内の他のテナントやユーザに付与することもできます。

ロールベースのアクセス コントロール ルールのアーキテクチャ

ロールベースのアクセス コントロール（RBAC）ルールには、ロールベースのアクセス コントロール（RBAC）モデルを強化して加筆性ルールを許可するブール型の `allowWrites` フィールドがあります。`allowWrites` フィールドがない場合に定義できるのは、読み取り RBAC ルールのみになります。

`RbacRule` クラスは次のように定義します。

```
Class aaa:RbacRule (CONCRETE)
Encrypted: false
Exportable: true
Persistent: true
Configurable: true
Write Access: [aaa, admin]
Read Access: [aaa, admin]
```

RBACルールにより、ユーザはセキュリティ ドメインから特定のオブジェクトで始まるサブツリーを読み取ることができます。

```
DN FORMAT:   [1] uni/rbacdb/rule-{[objectDn]}-dom-{domain}
```

表 *1 : aaa:RbacRule* プロパティの概要

プロパティ	タイプ	クラス	説明
aaa:Boolean	scalar:Enum8	allowWrites（aaa:RbacRule:allowWrites）	読み取り/書き込みまたは読み取りルール。
naming:Name	string:Basic	domain（aaa:RbacRule:domain）	カウントオブジェクトのドメイン。`aaa:ARbacRule:domain` を無効にします。
reference:BinRef		objectDn（aaa:RbacRule:objectDn）	`aaa:ARbacRule:objectDn` を無効にします。

`PartialRbacRule` クラスは `fvTenant` クラスの下に定義され、テナントがRBACルール（セルフサービス）を作成できるようにします。`PartialRbacRule` クラスは次のように定義されます。

```
Class aaa:PartialRbacRule (CONCRETE)
Encrypted: false
Exportable: true
Persistent: true
Configurable: true
Write Access: [aaa, admin]
Read Access: [aaa, admin]
```

表 *2 : aaa:PartialRbacRule* プロパティの概要

プロパティ	タイプ	クラス	説明
aaa:Boolean	scalar:Enum8	allowWrites（aaa:PartialRbacRule:allowWrites）	読み取り/書き込みまたは読み取りルール。
naming:Name	string:Basic	domain（aaa:PartialRbacRule:domain）	カウントオブジェクトのドメイン。
reference:BinRef		monPolDn（aaa:PartialRbacRule:monPolDn）	この監視可能なオブジェクトにアタッチするモニタリングポリシー。
reference:BinRef		partialObjectDn（aaa:PartialRbacRule:partialObjectDn）	

テナントによる `PartialRbacRule` クラスの作成では、`partialObjectDn` の正当性を確認する必要があります。`partialObjectDn` がテナント サブツリーの下にあれば有効です。親テナント サブツリー外の識別名は許可されていません。

管理者は、システム内の識別名を指す RbacRule を作成できます。テナント管理者が作成できるのは、テナント管理者のテナント サブツリー内にある識別名を指す PartialRbacRule のみです。

ロールベース アクセス コントロール ルールのシステム フロー

レイヤ4～レイヤ7ポリシーの設定前、設定中、または設定後に、テナント管理者は、特定のファイアウォールとロード バランサ デバイスへのアクセス権を自分のテナント ユーザに付与する PartialRbacRule の作成を選択することができます。各リソース グループを表す aaaDomain を作成し、個々に割り当てることでアクセスが実現します。次にセットアップの例を示します。

テナント	Acme
ユーザ	acme-admin acme-firewall-1-admin acme-firewall-2-admin acme-loadbalancer-1-admin acme-loadbalancer-2-admin
ファイアウォール デバイス	Firewall1 Firewall2
ロード バランサ デバイス	LB1 LB2

テナント管理者ユーザの acme-admin は、デバイスの Firewall1、Firewall2、LB1、および LB2 を作成したいと考えています。各デバイスに対する完全な書き込みアクセス許可をユーザごとに割り当てる必要があります。たとえば、ユーザ acme-firewall-1-admin にはデバイス Firewall1 ポリシーへの書き込み権限のみが必要ですが、ユーザ acme-loadbalancer-1-admin にはデバイス LB1 ポリシーへの書き込み権限のみが必要です。これを実現するには、acme-admin ユーザが、次のアクセス権を付与する 4 つの PartialRbacRule を作成する必要があります。

- Firewall1 識別名：ドメイン acme-firewall1 による書き込みが可能
- Firewall2 識別名：ドメイン acme-firewall2 による書き込みが可能
- LB1 識別名：ドメイン acme-lb1 による書き込みが可能
- LB2 識別名：ドメイン acme-lb2 による書き込みが可能

ユーザには次の権限が割り当てられます。

- ユーザ：acme-firewall-1-admin
  - ドメイン acme：read-all 権限
  - ドメイン acme-firewall1：テナント管理/書き込み

- ユーザ：acme-firewall-2-admin
  - ドメイン acme：read-all 権限
  - ドメイン acme-firewall2：テナント管理/書き込み
- ユーザ：acme-lb-1-admin
  - ドメイン acme：read-all 権限
  - ドメイン acme-lb1：テナント管理/書き込み
- ユーザ：acme-lb-2-admin
  - ドメイン acme：read-all 権限
  - ドメイン acme-lb2：テナント管理/書き込み

上記 4 人のユーザのいずれも、ドメイン acme の権限によって、acme テナント サブツリーを読み取れますが、どのノードにも書き込めません。テナント acme-lb2 のテナント管理/書き込みの権限によって、ユーザは LB2 ポリシー サブツリーのみに書き込むことができます。

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。