



管理ツール

この章の内容は、次のとおりです。

- [管理ツール, 1 ページ](#)
- [管理 GUI について, 1 ページ](#)
- [CLI について, 2 ページ](#)
- [Visore 管理対象オブジェクト ビューア, 3 ページ](#)
- [管理情報モデルのリファレンス, 4 ページ](#)
- [API インスペクタ, 5 ページ](#)
- [ユーザ ログインのメニュー オプション, 6 ページ](#)
- [MIT 内のオブジェクトの検索, 7 ページ](#)
- [エクスポート/インポートの設定, 12 ページ](#)

管理ツール

Cisco アプリケーションセントリック インフラストラクチャ (ACI) のツールは、ファブリックの管理者、ネットワーク エンジニア、および開発者がテナントおよびアプリケーションの導入を開発、設定、デバッグおよび自動化するのに役立ちます。

管理 GUI について

次の管理 GUI の機能により、ファブリックおよびそのコンポーネント（リーフとスパイン）にアクセスできます。

- 世界共通の Web 標準 (HTML5) に基づく。インストーラまたはプラグインは必要ありません。

- モニタリング（統計情報、障害、イベント、監査ログ）、操作および設定データへのアクセス。
- シングル サインオン メカニズムによる APIC とスパインおよびリーフ スイッチへのアクセス。
- サードパーティが使用できる同じ RESTful API を使用した APIC との通信。

CLI について

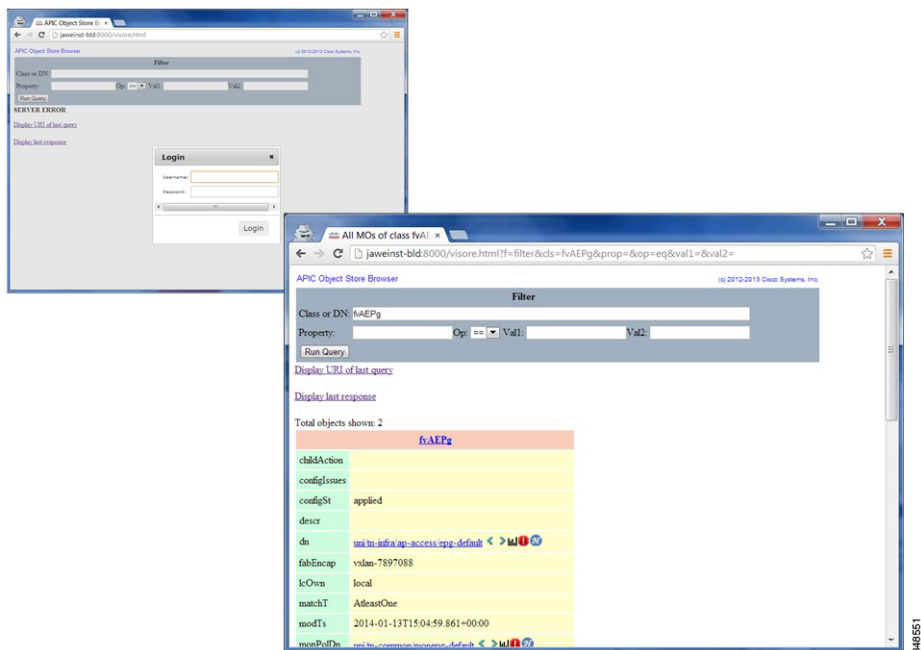
CLI は、APIC、リーフおよびスパイン スイッチへの操作インターフェイスおよび設定インターフェイスを特徴としています。

- Python で初めから実行され、Python インタープリタと CLI 間で切替えることができます。
- 拡張性のプラグイン アーキテクチャ
- 監視データ、操作データおよび構成データへのコンテキストベースのアクセス
- Python コマンドまたはバッチ スクリプティングによる自動化

Visore 管理対象オブジェクトビューア

Visore は、下の図に示すように、読み取り専用の管理情報ツリー（MIT）ブラウザです。これにより、オプションのフィルタを使用して、識別名（DN）とクラスのクエリーが可能になります。

図 1: Visore MO ビューア



Visore 管理対象オブジェクトビューアは次の場所にあります。 `http(s)://host[:port]/visore.html`

管理情報モデルのリファレンス

管理情報モデル（MIM）には、システム内のすべての管理対象オブジェクトとそのプロパティが含まれます。MIT内のオブジェクトを検索するために管理者がどのようにMIMを使用できるかに関する例については、次の図を参照してください。

図 2: MIM リファレンス

The screenshot displays the Management Information Model Reference interface. On the left, a sidebar lists various classes under the 'All Packages' section, including 'aaa:Provider', 'aaa:ActEp', 'aaa:AuthMethod', 'aaa:AuthRealm', 'aaa:Banner', 'aaa:ChangePassword', 'aaa:ChangeSSHKey', 'aaa:ChangeSshKey', 'aaa:ChangeSshKey', 'aaa:Config', 'aaa:ConsoleAuth', 'aaa:DefaultAuth', 'aaa:Definition', 'aaa:Domain', 'aaa:DomainAuth', 'aaa:DomainRef', 'aaa:DomainRolesTuple', 'aaa:Ep', 'aaa:IBCEp', 'aaa:IdcEp', 'aaa:LocalProvider', 'aaa:LocalProviderGroup', 'aaa:LoginDomain', 'aaa:Modi', 'aaa:PrelogBanner', 'aaa:ProviderGroup', 'aaa:ProviderRef', 'aaa:RadProvId', 'aaa:RadiusEp', 'aaa:RadiusProvider', 'aaa:RadiusProviderGroup', 'aaa:Radius', 'aaa:RemoteUser', and 'aaa:Role'. Below the list are sections for 'Methods', 'Types', 'Events', 'Faults', 'FSMs', 'Errors', and 'Syslog Messages'.

The main content area is titled 'Management Information Model Reference' and shows details for the class 'aaa:Ep (ABSTRACT)'. The class ID is 766. Properties include: Enabled: false, Exportable: true, Persistent: true, Write Access: [aaa, admin, none], Read Access: [aaa, admin, none], Semantic Scope: None, Semantic Scope Evaluation Rule: Subclasses, Monitoring Policy Source: Parent, and Monitoring Flags: [IsObservable: false, HasStats: false, HasFaults: false, HasHealth: false]. A note states: 'The base class for a AAA endpoint. This is an abstract class and cannot be instantiated.'

Below the class details is the 'Naming Rules' section, which shows a FORGAT rule: [0] aaa/aaa:aaa/.

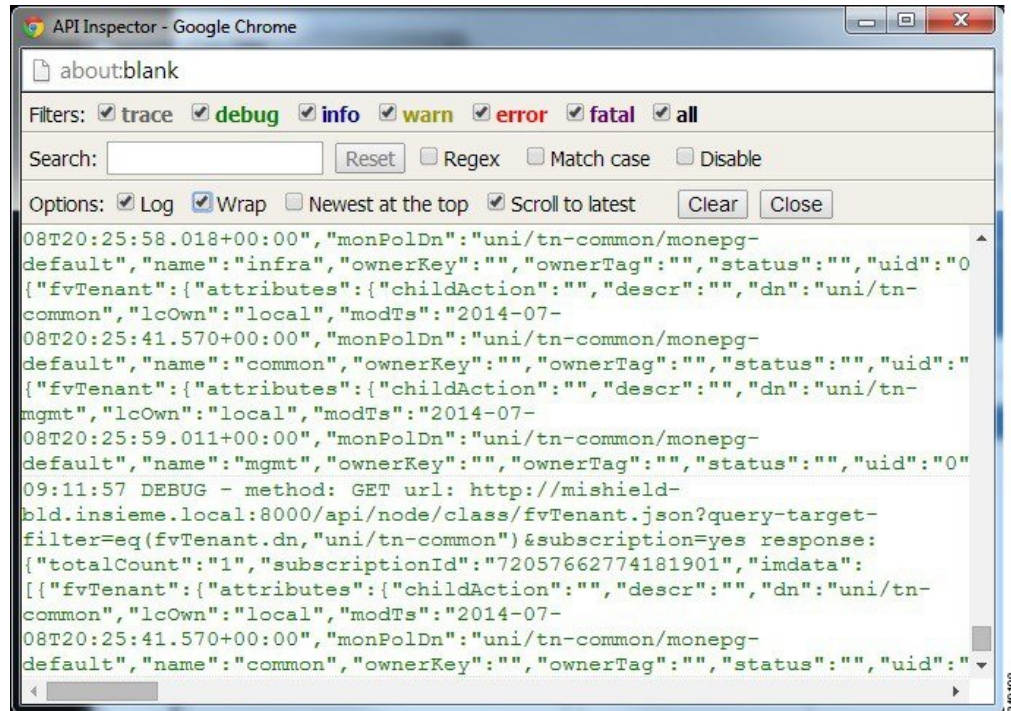
The 'Diagram' section shows an inheritance diagram. A legend on the left explains the symbols: a circle with 'C' for Concrete Model A, a circle with 'A' for Abstract Model B, and a circle with 'R' for Relation Model. The diagram shows 'Ep' (Abstract Model B) as the base class. It has four subclasses: 'UserEp' (Concrete Model A), 'Definition' (Abstract Model B), 'LocalEp' (Concrete Model A), and 'RadiusEp' (Concrete Model A). 'LocalEp' and 'RadiusEp' have their own subclasses: 'LocalEp' has 'aaa:LocalAuth' and 'aaa:LocalAuth', and 'RadiusEp' has 'aaa:RadiusAuth' and 'aaa:RadiusAuth'. 'Definition' has a subclass 'aaa:Definition'. 'UserEp' has a subclass 'aaa:UserAuth'. The diagram also shows 'TacacsPlusEp' (Concrete Model A) as a subclass of 'Ep'.

348553

API インспекタ

API インспекタでは、APIC が GUI インタラクションを実行するために処理する REST API コマンドのリアルタイム表示が提供されます。下の図は、API インспекタが GUI の主要テナントのセクションに移動する場合に表示する REST API コマンドを示します。

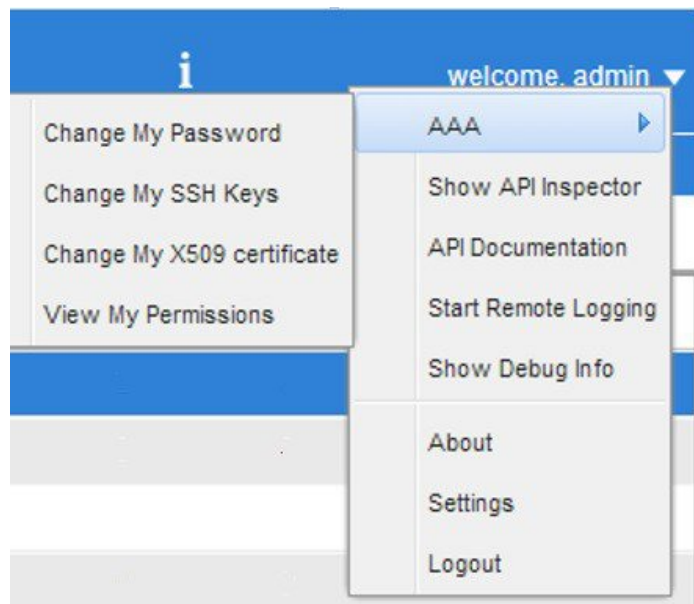
図 3: API インспекタ



ユーザ ログインのメニューオプション

ユーザログインのドロップダウンメニューにより、複数の設定、診断、参照およびプリファレンスのオプションが提供されます。次の図は、このドロップダウンメニューを示します。

図 4: ユーザ ログインのメニューオプション



オプションには次のものが含まれます。

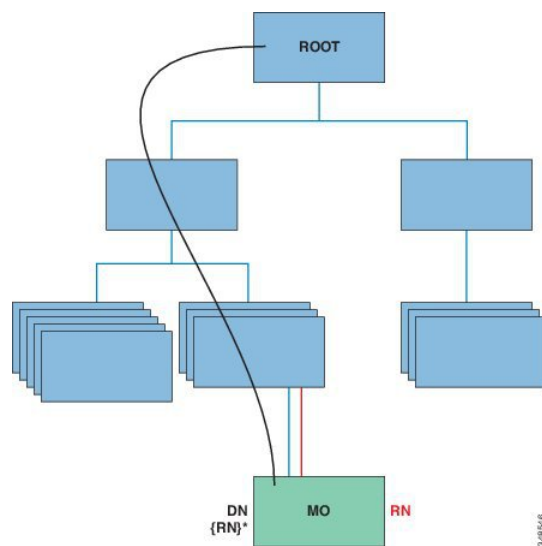
- ユーザパスワード、SSH キー、X509 証明書を変更、およびログインしたユーザの権限を表示するための AAA オプション。
- [Show API Inspector] では、API インスペクタが開きます。
- [API Documentation] では、管理情報モデルの参照を開きます。
- リモート ロギング。
- デバッグ情報。
- ソフトウェアの現在のバージョン番号について。
- GUI を使用するためのプリファレンスの設定。
- システムを終了するためのログアウト。

MIT内のオブジェクトの検索

Cisco ACIは情報モデルベースのアーキテクチャ（管理情報ツリー（MIT））を使用しており、管理プロセスによって制御できるすべての情報がモデルによって説明されます。オブジェクトインスタンスは管理対象オブジェクト（MO）と呼ばれます。

次の図は、任意のMOインスタンスを一意的に表す識別名と、親MOの下にあるMOをローカル的に表す相対名を示します。MIT内のオブジェクトはすべて、ルートオブジェクトの下に存在します。

図 5: MOの識別名と相対名



システム内のすべてのMOは固有の識別名（DN）によって識別されます。このアプローチにより、グローバルにオブジェクトを参照できます。またオブジェクトの識別名のほか、各オブジェクトを相対名（RN）で参照することもできます。相対名は、親オブジェクトに対して相対的にオブジェクトを識別します。指定されたオブジェクトの識別名は、親オブジェクトの識別名に相対名を加えることで取得できます。

DNは、オブジェクトを一意的に識別する一連の相対名です。

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

```
dn = "sys/ch/lcslot-1/lc/leafport-1"
```

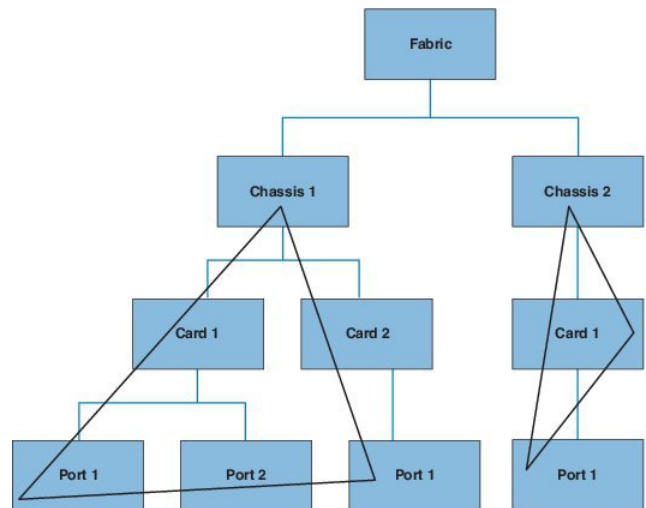
識別名はURLに直接マッピングされます。MIT内におけるオブジェクトの現在位置に応じて、相対名または識別名のいずれかを使用してオブジェクトにアクセスできます。

ツリーは階層型で構成され、属性システムを使用してオブジェクトクラスを識別できるため、さまざまな方法で管理対象オブジェクトの情報を取得するためにツリー内を照会できます。クエリは、識別名を使用してオブジェクト自体に対して実行するか、スイッチシャーシなどのオブジェクトのクラスに対して実行するか、ツリーレベルで実行してオブジェクトのすべてのメンバーを検出できます。

ツリーレベルのクエリ

次の図は、クエリー対象の2つのシャーシをツリーレベルで示しています。

図 6: ツリーレベルのクエリ

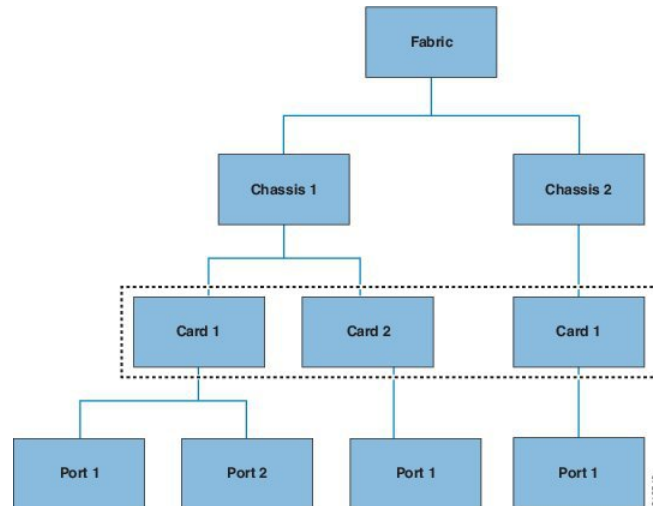


どちらのクエリも、参照されたオブジェクトと、その子オブジェクトを返します。このアプローチは、大規模なシステムのコンポーネントを検出するために役立ちます。この例では、クエリーにより指定されたスイッチシャーシのカードとポートが検出されます。

クラスレベルクエリー

次の図は、2 番目のクエリー タイプ、クラスレベルクエリーを示します。

図 7: クラスレベルクエリー



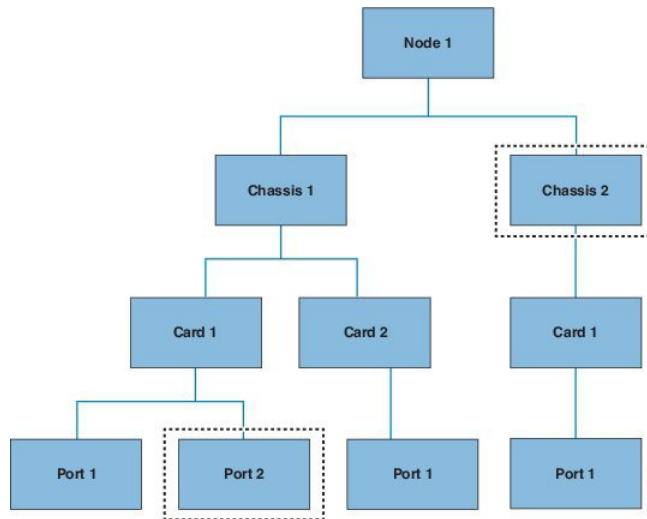
クラスレベルクエリーは、任意のクラスのオブジェクトをすべて返します。このアプローチは、MITで使用できる特定のタイプのオブジェクトをすべて検出する場合に役立ちます。この例で使用しているクラスはカードで、カードタイプのすべてのオブジェクトを返します。

オブジェクトレベルクエリー

3 つ目のクエリ タイプはオブジェクトレベルクエリです。オブジェクトレベルクエリでは、識別名を使用して特定のオブジェクトを返します。次の図は、2 つのオブジェクトレベルクエリー

を示しており、1つはノード1/シャーシ2、もう1つはノード1/シャーシ1/カード1/ポート2を照会しています。

図 8: オブジェクトレベルクエリー



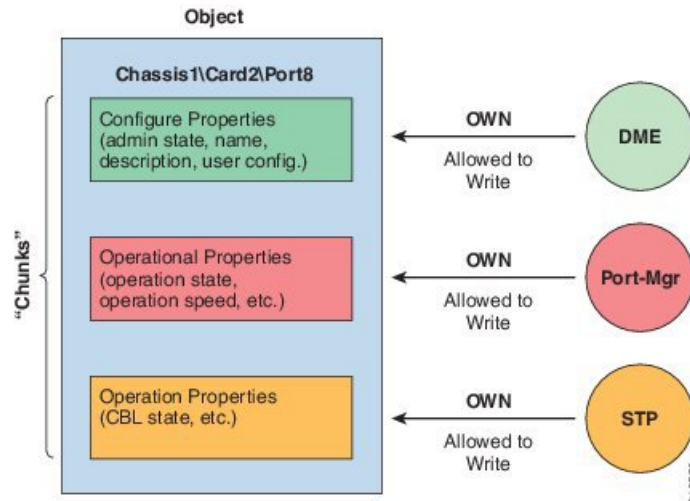
すべての MIT クエリーで、管理者はサブツリー全体またはサブツリーの一部を返すよう選択できます。また、システム内のロールベースアクセスコントロール (RBAC) メカニズムによって、返されるオブジェクトが決まります。必ず、ユーザが表示権限を持つオブジェクトのみが返されます。

管理対象オブジェクトのプロパティ

Cisco ACI の管理対象オブジェクトには、管理対象オブジェクトを定義するプロパティが含まれています。管理対象オブジェクトのプロパティはチャンクに分割され、オペレーティングシステム内でプロセスによって管理されます。オブジェクトには、複数のプロセスがアクセスする場合は

あります。これらのプロパティはすべて実行時にまとめてコンパイルされ、単一のオブジェクトとしてユーザに提示されます。次の図は、この関係の例を示します。

図 9: 管理対象オブジェクトのプロパティ



オブジェクトの例には、オブジェクト内のプロパティ チャンクに書き込むプロセスが 3 つあります。Cisco APIC (つまりユーザ) とオブジェクトとの間のインターフェイスとなるデータ管理エンジン (DME)、ポートの構成を処理するポート マネージャ、およびスパンニング ツリー プロトコル (STP) のすべてが、このオブジェクトのチャンクとやりとりします。APIC は、実行時にコンパイルされる単一のエンティティとしてオブジェクトをユーザに提示します。

REST インターフェイスによるオブジェクト データへのアクセス

REST は、World Wide Web などの分散型システム用ソフトウェア アーキテクチャの形式で、形式がシンプルであるため、Simple Object Access Protocol (SOAP) や Web サービス記述言語 (WSDL) など、その他の設計モデルに代わって採用される機会が増えています。Cisco APIC は REST インターフェイスをサポートしており、Cisco ACI ソリューション全体へのプログラムを通じたアクセスを実現します。

Cisco ACI のオブジェクトベース情報モデルは、REST インターフェイスに非常にうまく適合しています。URL と URI は識別名に直接マッピングされ、MIT 上のオブジェクトを識別でき、MIT 上のデータを XML または JSON 形式でエンコードされた自己完結型の構造化テキスト ツリー ドキュメントとして記述できます。オブジェクトには、識別名とプロパティを使用して識別される親子関係があり、この関係は一連の作成、読み取り、更新、および削除 (CRUD) 操作によって読み取りと変更が可能です。

オブジェクトにアクセスするには、明確に定義されたアドレスである REST URL を使用します。Cisco APIC オブジェクト データを取得および操作するには標準の HTTP コマンドを使用します。使用できる URL の形式は次のとおりです。

```
<system>/api/[mo|class]/[dn|class][:method].[xml|json]?{options}
```

URL の前に指定する各構成要素は、次のとおりです。

- `system` : システム識別子、IP アドレスまたは DNS で解決可能なホスト名
- `mo | class` : これが MIT 内の MO かまたはクラスレベルのクエリーかどうかの表示
- `class` : 照会するオブジェクトの MO クラス（情報モデルでの指定に従う）。クラス名は、`<pkgName><ManagedObjectClassName>` で表されます。
- `dn` : 照会するオブジェクトの識別名（MIT 内のオブジェクトの一意の階層名）
- `method` : オブジェクトに対して呼び出すメソッドの指定（オプション）。HTTP POST リクエストにのみ適用されます。
- `xml | json` : エンコード形式
- `options` : クエリー オプション、フィルタ、引数

REST URL で個々のオブジェクトまたはオブジェクト クラスのアドレスを指定してアクセスできる機能により、管理者はオブジェクトツリー全体、つまりシステム全体にプログラムを通じて完全にアクセスできます。

次に、REST クエリーの例を示します。

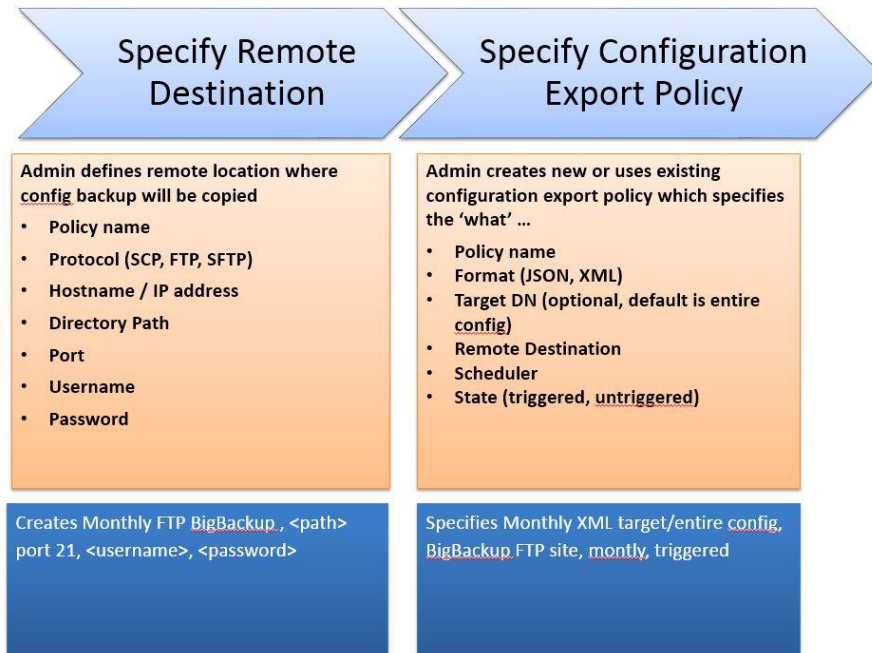
- テナント `solar` 下のすべての EPG と障害を検索する。
<http://192.168.10.1:7580/api/no/uni/tn-solar.xml?query-target=subtree&target-subtree-class=fvAEPg&rsp-subtree-include=faults>
- フィルタされた EPG クエリー
[http://192.168.10.1:7580/api/class/fvAEPg.xml?query-target-filter=eq\(fvAEPg.fabEncap,%20"vxlan-12780288"\)](http://192.168.10.1:7580/api/class/fvAEPg.xml?query-target-filter=eq(fvAEPg.fabEncap,%20)

エクスポート/インポートの設定

すべての APIC ポリシーおよび設定データは、バックアップの作成のためにエクスポートできます。これは、エクスポート ポリシーを使用して設定でき、リモート サーバにスケジュールバックアップまたは即時バックアップできます。スケジュールバックアップは、定期バックアップジョブまたは繰り返しバックアップジョブを実行するように設定できます。デフォルトでは、すべてのポリシーおよびテナントがバックアップされますが、管理者は任意に管理情報ツリーの特定のサブツリーのみを指定できます。バックアップは、インポート ポリシーによって APIC にインポートでき、システムを以前の設定に復元できます。

次の図は、エクスポート ポリシーを設定するプロセスがどのように動作するかを示します。

図 10: エクスポート ポリシーを設定するワークフロー

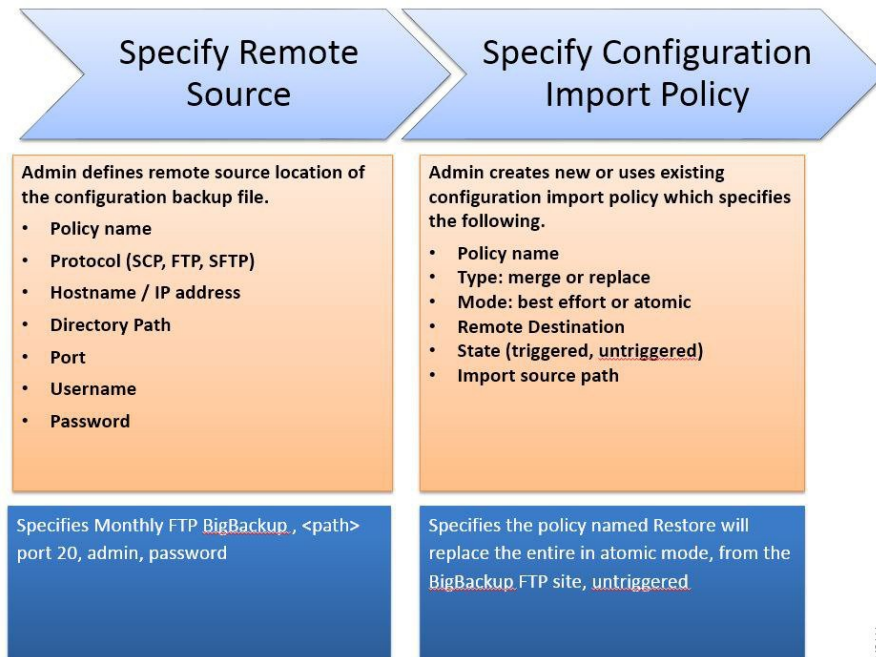


APIC はこのポリシーを次のように適用します。

- 完全なシステム構成のバックアップは月に一度実行されます。
- バックアップは BigBackup FTP サイトに XML 形式で保存されます。
- ポリシーがトリガーされます（有効です）。

次の図は、インポートポリシーを設定するプロセスがどのように動作するかを示します。

図 11: インポートポリシーを設定するワークフロー



APICはこのポリシーを次のように適用します。

- 毎月のバックアップから完全なシステム構成の復元を実行するためのポリシーが作成されます。
- 復元アトミックモードは、無効な設定をシャードにインポートしようとするときシャード全体をスキップします。
- ポリシーはトリガーされません（使用できますが、アクティブ化されていません）。

インポートポリシーは次のオプションをサポートしています。

アクション

- [Merge] : インポートされた設定は、既存の設定とマージされます。
- [Replace] : インポートされた設定は、既存の設定と置き換わります。以前にインポートされたファイルに存在しない既存の設定データが削除されます。

モード

- [Atomic] : すべての設定データをインポートしようとします。インポートできないオブジェクトがある場合、シャードへのインポート操作が失敗します。
- [Best-effort] : すべての設定をインポートしようとしますが、インポートできないオブジェクトは無視します。

テクニカル サポート、統計情報、コア

管理者は、APIC 内で、コア ファイルとデバッグ データを処理するために、統計情報、テクニカル サポートの収集、障害およびイベントをファブリック（APIC およびスイッチ）から外部ホストにエクスポートするようエクスポートポリシーを設定できます。エクスポートはXML、JSON、Web ソケット、SCP、HTTP などのさまざまな形式にできます。エクスポートはサブスクライブでき、定期的またはオンデマンドでストリーミングできます。

管理者は、転送プロトコル、圧縮アルゴリズム、転送の頻度などポリシーの詳細を設定できます。ポリシーは、AAA を使用して認証されたユーザによって設定できます。実際の転送のセキュリティメカニズムは、ユーザ名とパスワードに基づいています。内部的に、ポリシー要素はデータのトリガーを処理します。

