CHAPTER 4

管理作業の実行

この章では、Microsoft 管理ツールを入手して、Active Directory 環境でユーザとコンピュータにワイヤレス プロファイルを配布する方法について説明します。また、EAP-FAST、LEAP、および PEAP-GTC の XML スキーマについても説明します。

次の項目について説明します。

Microsoft ツールを使用した管理作業の実行

管理者は、Microsoft のグループ ポリシー オブジェクト エディタで Microsoft のグループ ポリシー オブジェクトを作成して Active Directory 環境内のユーザやコンピュータにワイヤレス プロファイルを配布するなど、各種管理作業を実行する必要があります。これらの Microsoft ソリューションとその機能については、このシスコのマニュアルでは扱いません。

次の項では、Microsoft ツールを使用して管理作業を実行する際に役立つ予備的な情報や参考資料について取り上げます。

グループ ポリシー オブジェクトの概要

グループ ポリシーは、Active Directory ディレクトリ サービス環境で、ユーザおよびコンピュータに対して設定を指定し管理することができるインフラストラクチャです。グループ ポリシーの設定は、グループ ポリシー オブジェクト（GPO）に含まれています。GPO はドメインに存在し、サイト、ドメイン、組織ユニット（OU）などの Active Directory コンテナに割り当てることができます。

GPO と GPO エディタの詳細については、次の URL にある Microsoft Windows Server TechCenter を参照してください。

http://technet2.microsoft.com/windowsserver/en/technologies/featured/gp/faq.mspx

Microsoft は、Microsoft 管理コンソール（MMC）でグループ ポリシー オブジェクト エディタを使用できるプログラム スナップインを提供しています。

MMC の詳細は、次の URL にある Microsoft 管理コンソールのヘルプを参照してください。

http://www.microsoft.com/technet/WindowsVista/library/ops/06e1cb7b-19c9-4c49-9db8-a941f6f593c3.mspx

グループ ポリシー オブジェクト エディタの追加

グループ ポリシー オブジェクトを設定する前に、グループ ポリシー オブジェクト エディタのスナップインを追加する必要があります。スナップインを追加するには、次の手順を実行します。

ステップ 1　MMC を開きます。

a. デスクトップの左下にある **Start** ボタンをクリックします。

b. **検索**の**ボックス**に「**mmc**」と入力し、**Enter** キーを押します。

（注）　既存または保存済みの MMC コンソールを開くには、Windows エクスプローラでスナップイン コンソールまたはスナップイン コンソールへのショートカットを探してダブルクリックします。

使用している別のコンソールから既存の MMC コンソールを開くこともできます。これを行うには、**File** メニューをクリックし、**Open** をクリックします。

ステップ 2　グループ ポリシー オブジェクト エディタのスナップインを追加します。

a. **File**、**Add/Remove Snap-in...** の順にクリックします。

Add/Remove Snap-in... ダイアログ ボックスが表示されます。

b. **Add or Remove Snap-ins** ダイアログ ボックスで、**Available snap-ins** リストの **Group Policy Object Editor** を選択し、**Add** ボタンをクリックします。

Select Group Policy Object ダイアログ ボックスが表示されます。

c. **Select Group Policy Object** ダイアログ ボックスで、**Browse** をクリックします。

Browse for a Group Policy Object ダイアログ ボックスが表示されます。

d. **Browse for a Group Policy Object** ダイアログ ボックスで、**Domains/O Us** タブを選択します。

e. **Look in** ドロップダウン リストからドメイン コントローラを選択します。

f. **OK** をクリックします。

g. **Select Group Policy Object** ダイアログ ボックスで、**Finish** をクリックします。

h. **Add or Remove Snap-ins** ダイアログ ボックスで **OK** をクリックします。

これで、グループ ポリシー オブジェクト エディタを使用することができます。

Windows Vista での EAP グループ ポリシー オブジェクトの作成

新しい EAP グループ ポリシー オブジェクトを作成するには、次の手順を実行します。

ステップ 1　**Default Domain Policy** ペインで、**Windows Settings**、**Security Settings**、**Wireless Network Policies** の順に選択します。

ステップ 2　**Wireless Network Policies** を右クリックし、**Create a New Policy** を選択します。

ステップ 3　SSID、暗号化、認証方式など、ワイヤレス ネットワークのプロパティを設定します。

ステップ 4　EAP 方式を選択します。

ステップ 5　対象の EAP モジュールのプロパティを開き、設定を行います。

- EAP-FAST — **Advanced Security** 画面で、マシン認証、SSO など、詳細な設定を行うことができます。マシン認証の詳細は、「EAP-FAST のマシン認証の設定」（4-4 ページ）を参照してください。SSO の詳細は、「EAP-FAST のシングル サインオンの設定」（4-5 ページ）を参照してください。
- PEAP-GTC — **Advanced Security** 画面で、マシン認証、SSO など、詳細な設定を行うことができます。マシン認証の詳細は、「PEAP-GTC のマシン認証の設定」（4-5 ページ）を参照してください。SSO の詳細は、「PEAP-GTC および LEAP のシングル サインオンの設定」（4-5 ページ）を参照してください。
- LEAP — **Advanced Security** 画面で、SSO サプリカント設定を行うことができます。SSO の詳細は、「PEAP-GTC および LEAP のシングル サインオンの設定」（4-5 ページ）を参照してください。

（注）　**Wired Network Policy** オブジェクトを選択することで、有線ネットワークの設定を行うことができます。

ステップ 6　完了したら、GPO を保存します。「gpupdate /force」を実行して GPO の更新を強制することにより、Vista クライアントを更新できます。新しいプロファイルが Vista マシンに追加されたのが確認できます。

GPO ネットワーク プロファイルの作成後、Vista マシンでそのプロファイルを変更することはできません。

ワイヤレス ネットワーク ポリシーの General タブでは、ポリシーの名前と説明の入力、WLAN 自動構成サービスを有効にするかどうかの指定、ワイヤレス ネットワーク ポリシーのリストと優先順位の設定を行うことができます。プロファイルを XML ファイルとしてエクスポートしたり、ワイヤレス プロファイルとして XML ファイルをインポートしたりすることもできます。

ポリシーの設定、プロファイルのエクスポート、プロファイルのインポートの詳細は、次のドキュメントを参照してください。

- *Windows Vista Wireless Networking Evaluation Guide*

http://technet2.microsoft.com/WindowsVista/en/library/f0b0d1fd-6dff-46a2-8e6a-bdd152d2337f1033.mspx?mfr=true

- *Wireless Group Policy Settings for Windows Vista（Windows Vista 用ワイヤレス グループ ポリシーの設定）*

http://www.microsoft.com/technet/technetmag/issues/2007/04/CableGuy/default.asp

EAP-FAST のマシン認証の設定

グループ ポリシー オブジェクトを作成するときに、Advanced Security 画面でマシン認証を有効にすることができます。

EAPHost は、EAP-FAST モジュールに、現在の認証がマシン認証であることを通知します。

マシン認証は、次のいずれかによって実行できます。

- マシン PAC
- マシン証明書
- マシン パスワード

EAP-FAST モジュールは、最初に、マシン PAC を取得しようとします。マシン PAC を取得できない場合、EAP-FAST モジュールはマシン証明書を取得しようとします。マシン証明書を取得できない場合、EAP-FAST モジュールは Active Directory 内のマシン アカウントのマシン パスワードを取得しようとします。

マシンがマシン証明書またはマシン パスワードで認証されると、EAP-FAST モジュールは、以降の使用のために、マシン PAC のプロビジョニングを要求します。マシン証明書もマシン パスワードも取得できない場合、EAP-FAST モジュールは、ユーザがログインした後、次回のユーザ認証成功時に、マシン PAC を要求します。既存のマシン PAC が無効か、期限切れの場合、EAP-FAST モジュールはこのプロセスを使用して新しいマシン PAC を要求します。

マシン認証は Windows 802.1X サプリカントで統合およびサポートされているため、EAP-FAST モジュールが担当するのは、ネットワークへのアクセス権を取得するための認証のみとなります。マシン認証をサポートするためのその他のネットワーク管理（DHCP、マシンレベルの GPO、その他の関連するネットワーク サービスなど）は、オペレーティング システムと 802.1X サプリカントの担当になります。

EAP-FAST のシングル サインオンの設定

SSO は、以下のように、Microsoft Windows Vista でサポートされます。

- Windows ユーザ クレデンシャルは、EAPHost インターフェイスを通じて EAP-FAST モジュールに渡されます。EAP-FAST モジュールが、ネットワーク認証で Windows ユーザ クレデンシャルを使用するように設定されており、かつ、ネットワーク プロファイルが、シングル サインオンを実行するように設定されている場合、システムはユーザに追加のクレデンシャルを求めません。
- Windows 以外のネットワーク クレデンシャルは、Microsoft Windows Vista のログオン プロセスで収集されます。EAP-FAST モジュールは、ユーザにこれらのネットワーク クレデンシャルを求めるよう、ログオン モジュールに要求します。
- 必要に応じて、EAP-FAST モジュールは、ユーザが Microsoft Windows Vista にログインする前に、ユーザに追加のネットワーク クレデンシャルを求めることができます。

ネットワーク クレデンシャルが設定に保存されている場合、EAP-FAST モジュールは、ユーザが Microsoft Windows Vista にログインする前に、これらのクレデンシャルにアクセスすることができます。

PEAP-GTC のマシン認証の設定

PEAP-GTC モジュールは、マシン パスワードを使用する場合に限り、マシン認証をサポートします。PEAP-GTC モジュールは、Microsoft のローカル セキュリティ機関（LSA）API 経由で Windows からマシン パスワードを取得します。この場合、ユーザはパスワードの入力を求められません。

マシン認証は、サプリカントで有効になり、設定されます。

PEAP-GTC および LEAP のシングル サインオンの設定

PEAP-GTC モジュールと LEAP モジュールでは、Microsoft Windows Vista により、次のようにシングル サインオン（SSO）がサポートされます。

- Windows ユーザ クレデンシャルは、EAPHost インターフェイスを通じてモジュールに渡されます。モジュールが、ネットワーク認証で Windows ユーザ クレデンシャルを使用するように設定されており、かつ、ネットワーク プロファイルが、シングル サインオンを実行するように設定されている場合、システムはユーザに追加のクレデンシャルを求めません。
- Windows 以外のネットワーク クレデンシャルは、Microsoft Windows Vista のログオン プロセスで収集されます。モジュールは、ユーザにこれらのネットワーク クレデンシャルを求めるよう、ログオン モジュールに要求します。
- Windows 802.1X サプリカントは、グループ ポリシー プロセスを処理し、それが Window のログイン プロセスと同期され、実行されるようにします。
- 必要に応じて、モジュールは、ユーザが Microsoft Windows Vista にログインする前に、ユーザに追加のネットワーク クレデンシャルを求めることができます。
- ネットワーク クレデンシャルが設定に保存されている場合、モジュールは、ユーザが Microsoft Windows Vista にログインする前に、これらのクレデンシャルにアクセスすることができます。

EAP-FAST XML スキーマ

EAP-FAST モジュールは、次のスキーマを使用することで、ネットワーク プロファイルのネイティブ EAP 方式セクションのすべての設定を XML として保存します。

```
<?xml version="1.0"?>

<!--
*******************************************************************************
                    Cisco EAP-FAST スキーマ          (1.0.40)
Copyright 2006-2007, Cisco Systems, Inc.              All rights reserved.
*******************************************************************************
-->

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="http://www.cisco.com/CCX"
 targetNamespace="http://www.cisco.com/CCX"
 elementFormDefault="qualified"
 attributeFormDefault="unqualified">

 <xs:element name="eapFast" type="EapFast"/>

 <xs:complexType name="EapFast">
  <xs:complexContent>
   <xs:extension base="TunnelMethods">
    <xs:sequence>
     <xs:choice>
      <xs:element name="usePac">
       <xs:complexType>
        <xs:sequence>
         <xs:element name="allowUnauthPacProvisioning" type="xs:boolean" default="true">
          <xs:annotation>
                    <xs:documentation> 認証されていないサーバからの PAC を受け入れます。
</xs:documentation>
          </xs:annotation>
```

```
</xs:element>
<xs:element name="autoGrouping" type="xs:boolean" default="true">
 <xs:annotation>
  <xs:documentation>
```

aid グループは、すべて等しく信頼されている A-ID のセットです。グループ内の A-ID はどれでも使用できます。自動グループ化とは、信頼されていない A-ID がエンドユーザによって受け入れられた場合に、その A-ID が、そのプロファイルですでに信頼されている A-ID と一緒にグループ化されること、つまり、ユーザのアクションに基づいて、A-ID グループを自動的に作成および拡張することを意味します。A-ID グループの利点は、プロファイルで最初に信頼されている A-ID(1) があり、その後、そのプロファイルの使用時にエンドユーザが新しい A-ID(2) の使用を許可した場合に、再度エンドユーザに尋ねることなく、A-ID(2) が自動的に受け入れられることです。
</xs:documentation>

```
  </xs:annotation>
</xs:element>
<xs:element name="userValidatesServerIdFromUnauthProv" type="xs:boolean" default="true">
 <xs:annotation>
  <xs:documentation>
```

true の場合、クライアントが認証されていないプロビジョニングを実行する前に、ユーザに、認証されていないプロビジョニングを許可するかどうかを尋ねるメッセージが表示されます。
</xs:documentation>

```
  </xs:annotation>
</xs:element>
<xs:element name="unauthProvAllowedTilPacReceived" type="xs:boolean" default="false">
 <xs:annotation>
  <xs:documentation>true の場合、認証されていないプロビジョニングが成功し、PAC が取得されるまで、認証されていないプロビジョニングの実行が許可され、その後、認証されているプロビジョニングのみが許可されるようになります。</xs:documentation>
 </xs:annotation>
</xs:element>
<xs:choice>
 <xs:element name="validateWithSpecificPacs" type="ValidateWithSpecificPacs">
  <xs:annotation>
   <xs:documentation> これは、この要素で参照されている PAC（およびこのプロファイルの使用時にこのプロファイルに自動的にプロビジョニングされた PAC）のみを検証で使用することを示します。</xs:documentation>
  </xs:annotation>
 </xs:element>
</xs:choice>
</xs:sequence>
```

```
    </xs:complexType>
   </xs:element>
   <xs:element name="doNotUsePac" type="Empty">
    <xs:annotation>
     <xs:documentation> 認証で PAC を使用しません。</xs:documentation>
    </xs:annotation>
   </xs:element>
  </xs:choice>
  <xs:element name="enablePosture" type="xs:boolean" default="false">
   <xs:annotation>
    <xs:documentation> ポスチャ情報の処理を許可します。</xs:documentation>
   </xs:annotation>
  </xs:element>
  <xs:element name="authMethods">
   <xs:complexType>
    <xs:choice>
     <xs:element name="builtinMethods">
      <xs:complexType>
       <xs:choice>
        <xs:element name="authenticateWithPassword">
         <xs:complexType>
          <xs:sequence>
           <xs:element name="protectedIdentityPattern" type="IdentityPattern" minOccurs="0">
            <xs:annotation>
             <xs:documentation> 形式の規則は unprotectedIdentityPattern と同じです。通常のパターンは [username]@[domain] です。パスワードのソースがこのプロファイルの場合は、ユーザ名として送信する実際の文字列になります。</xs:documentation>
            </xs:annotation>
           </xs:element>
           <xs:element name="passwordSource" type="PasswordSource"/>
           <xs:element name="methods">
            <xs:annotation>
             <xs:documentation> 少なくとも 1 つの子要素が必要です。</xs:documentation>
            </xs:annotation>
            <xs:complexType>
```

```
            <xs:all>
              <xs:element name="eapMschapv2" type="Empty" minOccurs="0"/>
              <xs:element name="eapGtc" type="Empty" minOccurs="0"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="authenticateWithToken">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="protectedIdentityPattern" type="IdentityPattern" minOccurs="0">
          <xs:annotation>
            <xs:documentation> 形式の規則は unprotectedIdentityPattern と同じです。通常のパターンは [username]@[domain] です。</xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="tokenSource" type="TokenSource"/>
        <xs:element name="methods">
          <xs:complexType>
            <xs:all>
              <xs:element name="eapGtc" type="Empty"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="authenticateWithCertificate">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="protectedIdentityPattern" type="IdentityPattern" minOccurs="0">
          <xs:annotation>
```

<xs:documentation> 形式の規則は unprotectedIdentityPattern と同じです。通常のパターンは [username]@[domain] です。</xs:documentation>

```
        </xs:annotation>
       </xs:element>
       <xs:element name="certificateSource" type="CertificateSource"/>
       <xs:choice>
        <xs:element name="doNotUseInnerMethod">
         <xs:complexType>
          <xs:choice>
           <xs:element name="sendWheneverRequested" type="Empty"/>
           <xs:element name="sendSecurelyOnly" type="Empty"/>
          </xs:choice>
         </xs:complexType>
        </xs:element>
        <xs:element name="sendViaInnerMethod">
         <xs:complexType>
          <xs:all>
           <xs:element name="eapTls" type="Empty"/>
          </xs:all>
         </xs:complexType>
        </xs:element>
       </xs:choice>
      </xs:sequence>
     </xs:complexType>
    </xs:element>
   </xs:choice>
  </xs:complexType>
 </xs:element>
</xs:sequence>
</xs:extension>
```

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

 <xs:complexType name="IdentityPattern">
  <xs:simpleContent>
   <xs:extension base="NonEmptyString">
    <xs:attribute name="encryptContent" type="xs:boolean" use="optional" default="true">
     <xs:annotation>
      <xs:documentation> デフォルトは 'true' です。これは、この要素を暗号化する必要があること
を後処理ツールに示しています。</xs:documentation>
     </xs:annotation>
    </xs:attribute>
   </xs:extension>
  </xs:simpleContent>
 </xs:complexType>

 <xs:complexType name="PasswordFromProfile">
  <xs:simpleContent>
   <xs:extension base="xs:string">
    <xs:attribute name="encryptContent" type="xs:boolean" use="optional" default="true">
     <xs:annotation>
      <xs:documentation> デフォルトは 'true' です。これは、この要素を暗号化する必要があること
を後処理ツールに示しています。</xs:documentation>
     </xs:annotation>
    </xs:attribute>
   </xs:extension>
  </xs:simpleContent>
 </xs:complexType>

 <xs:complexType name="PasswordSource">
  <xs:choice>
   <xs:element name="passwordFromLogon" type="Empty"/>
   <xs:element name="passwordFromUser" type="Empty"/>
   <xs:element name="passwordFromProfile" type="PasswordFromProfile"/>
  </xs:choice>
```

```
 </xs:complexType>

 <xs:complexType name="TokenSource">
  <xs:choice>
   <xs:element name="passwordFromOtherToken" type="Empty">
    <xs:annotation>
     <xs:documentation> これにより、トークンから ID と OTP を取得するためのプロンプトがユーザに表示されます。</xs:documentation>
    </xs:annotation>
   </xs:element>
  </xs:choice>
 </xs:complexType>

 <xs:complexType name="CertificateSource">
  <xs:choice>
   <xs:element name="certificateFromUser" type="Empty">
    <xs:annotation>
     <xs:documentation>
認証時に使用するクライアント証明書は、表示されたリストからエンドユーザが選択したものです。</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="certificateFromLogon" type="Empty">
    <xs:annotation>
     <xs:documentation> 認証時に使用するクライアント証明書は、Windows へのログオンでエンドユーザが使用したものです。</xs:documentation>
    </xs:annotation>
   </xs:element>
   <xs:element name="certificateFromProfile" type="ClientCertificate">
    <xs:annotation>
     <xs:documentation> 認証時に使用するクライアントのユーザ証明書が、ここに示されます。
</xs:documentation>
    </xs:annotation>
   </xs:element>
  </xs:choice>
 </xs:complexType>
```

```
<xs:complexType name="ExtendedInnerEapMethod">
 <xs:sequence>
  <xs:element name="methodName" type="xs:string"/>
  <xs:element name="methodEapId" type="xs:unsignedInt"/>
  <xs:element name="vendorId" type="xs:integer" default="0"/>
  <xs:element name="AuthorName" type="xs:string"/>
  <xs:element name="AuthorId" type="xs:unsignedInt"/>
  <xs:any namespace="##any" processContents="lax" minOccurs="0"/>
 </xs:sequence>
</xs:complexType>

<xs:complexType name="TunnelMethods">
 <xs:sequence>
  <xs:choice>
   <xs:element name="validateServerCertificate" type="serverCertificateValidationParameters"/>
   <xs:element name="doNotValidateServerCertificate" type="Empty"/>
  </xs:choice>
  <xs:element name="unprotectedIdentityPattern" type="IdentityPattern" minOccurs="0">
   <xs:annotation>
    <xs:documentation>[username] および [domain]、またはそのどちらかのプレースホルダが使用
される場合：認証でクライアント証明書が使用される場合、プレースホルダの値はクライアント証
明書の CN フィールドから取得されます。クレデンシャルがエンドユーザから取得される場合、プ
レースホルダの値はユーザが入力した情報から取得されます。クレデンシャルがオペレーティング
システムから取得される場合、プレースホルダの値はログオン情報から取得されます。通常のパ
ターン：anonymous@[domain]（トンネルされた方式の場合）または [username]@[domain]（トンネ
ルされていない方式の場合）。クレデンシャルのソースがこのプロファイルの場合は、ユーザ名と
して送信する実際の文字列になります（プレースホルダなし）。</xs:documentation>
   </xs:annotation>
  </xs:element>
  <xs:choice>
   <xs:element name="enableFastReconnect">
    <xs:complexType>
     <xs:complexContent>
      <xs:extension base="Empty">
       <xs:choice>
        <xs:element name="alwaysAttempt" type="Empty"/>
```

```
            </xs:choice>
           </xs:extension>
          </xs:complexContent>
         </xs:complexType>
        </xs:element>
        <xs:element name="disableFastReconnect" type="Empty"/>
       </xs:choice>
      </xs:sequence>
     </xs:complexType>

     <xs:complexType name="ClientCertificate">
      <xs:choice>
       <xs:element name="certificateId" type="CertificateIdentifier">
        <xs:annotation>
         <xs:documentation> これは、OS にあらかじめ保存されている証明書への参照です。
</xs:documentation>
        </xs:annotation>
       </xs:element>
      </xs:choice>
     </xs:complexType>

     <xs:complexType name="CertificateContainer">
      <xs:choice minOccurs="0" maxOccurs="unbounded">
       <xs:element name="certificateId" type="CertificateIdentifier">
        <xs:annotation>
         <xs:documentation> これは、OS にあらかじめ保存されている証明書への参照です。
</xs:documentation>
        </xs:annotation>
       </xs:element>
      </xs:choice>
     </xs:complexType>

     <xs:complexType name="CertificateIdentifier">
      <xs:simpleContent>
       <xs:annotation>
```

<xs:documentation>X509 形式のバイナリ証明書全体にわたる SHA 1 のハッシュ値。マシンの信頼済み CA のグローバル リストで証明書を一意に識別します（Windows の OS で管理されるストア）。</xs:documentation>

</xs:annotation>

<xs:extension base="NonEmptyString">

<xs:attribute name="reference" type="xs:boolean">

<xs:annotation>

<xs:documentation>true は、要素値が PEM 形式の証明書へのファイル参照であることを示します。後処理ツールが、その証明書ファイルを取得し、ハッシュ値に変換し、certificateId 要素に値を投入し、これがその証明書にわたる SHA1 のハッシュ値であることを示すために、参照を false に設定します。</xs:documentation>

</xs:annotation>

</xs:attribute>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

<xs:complexType name="Empty"/>

<xs:simpleType name="NonEmptyString">

<xs:restriction base="xs:string">

<xs:minLength value="1"/>

</xs:restriction>

</xs:simpleType>

<xs:complexType name="ServerRuleFormat">

<xs:simpleContent>

<xs:extension base="NonEmptyString">

<xs:attribute name="match" use="required">

<xs:simpleType>

<xs:restriction base="xs:string">

<xs:enumeration value="exactly"/>

<xs:enumeration value="endsWith"/>

</xs:restriction>

</xs:simpleType>

</xs:attribute>

```
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="ServerValidationRules">
   <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:annotation>
     <xs:documentation>
```

ユーザがサーバを信頼できる場合のみのオプションです。サーバ検証ルールを持たないプロファイルの開始が許可されると、ユーザが信頼されていないサーバを検証するときに、検証プロセスによってそのサーバ名が検証されます。</xs:documentation>

```
    </xs:annotation>
    <xs:element name="matchSubjectAlternativeName" type="ServerRuleFormat">
     <xs:annotation>
      <xs:documentation>DNSName：通常は、完全修飾ドメイン名（FQDN）の形式になります。
</xs:documentation>
     </xs:annotation>
    </xs:element>
    <xs:element name="matchSubject" type="ServerRuleFormat">
     <xs:annotation>
      <xs:documentation> サブジェクト：CN（共通名）- 通常は、単純な ASCII 文字列です。または、
サブジェクト：DN（ドメイン名）- 一連の DC ( ドメイン コンポーネント）属性で構成されます。
</xs:documentation>
     </xs:annotation>
    </xs:element>
   </xs:choice>
  </xs:complexType>

  <xs:complexType name="serverCertificateValidationParameters">
   <xs:sequence>
    <xs:choice>
     <xs:element name="serverNameValidationRules" type="ServerValidationRules"/>
     <xs:element name="anyServerName" type="Empty">
      <xs:annotation>
       <xs:documentation> 証明書内のサーバ名はテストされません。</xs:documentation>
      </xs:annotation>
```

```
    </xs:element>
   </xs:choice>
   <xs:choice>
    <xs:element name="validateChainWithSpecificCa">
     <xs:complexType>
      <xs:complexContent>
       <xs:extension base="CertificateContainer"/>
      </xs:complexContent>
     </xs:complexType>
    </xs:element>
    <xs:element name="validateChainWithAnyCaFromOs" type="Empty">
     <xs:annotation>
      <xs:documentation> グローバル CA 証明書ストアの CA 証明書で終わっている場合、その証明書チェーンは信頼されます。</xs:documentation>
     </xs:annotation>
    </xs:element>
   </xs:choice>
   <xs:element name="userValidatesUntrustedServerCertificate" type="xs:boolean">
    <xs:annotation>
     <xs:documentation> サーバ証明書の検証に失敗した場合、true に設定されていると、エンドユーザはサーバを検証するように求められます。検証すると、適切な trustedCaCerts およびサーバ名フィールドが記憶され、次回から自動的に信頼されるようになります。</xs:documentation>
    </xs:annotation>
   </xs:element>
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="ValidateWithSpecificPacs">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
   <xs:annotation>
    <xs:documentation> これはオプションです。これにより、エンジンでサーバ PAC を検証する必要があるが、PAC が、プロファイル内のこの場所で静的に定義されるのではなく、エンドユーザのアクションまたは認証されていないプロビジョニングによって動的に追加されるということをプロファイルで示すことができます。</xs:documentation>
   </xs:annotation>
   <xs:element name="trustPacFromGlobalPacStoreWithThisId" type="xs:string">
    <xs:annotation>
```

```
        <xs:documentation>
```

PAC 用のグローバル ストア（プロファイルごとのストアではない）が存在する場合に使用されます。</xs:documentation>

```
      </xs:annotation>
    </xs:element>
  </xs:choice>
 </xs:complexType>


</xs:schema>
```

PEAP-GTC XML スキーマ

PEAP-GTC モジュールは、次のスキーマを使用することで、ネットワーク プロファイルのネイティブ EAP 方式セクションのすべての設定を XML として保存します。

```
<?xml version="1.0"?>

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="http://www.cisco.com/CCX"
 targetNamespace="http://www.cisco.com/CCX"
 elementFormDefault="qualified"
 attributeFormDefault="unqualified">

 <xs:element name="eapPeap" type="EapPeap"/>

 <xs:complexType name="EapPeap">
  <xs:complexContent>
   <xs:extension base="TunnelMethods">
    <xs:sequence>
     <xs:element name="authMethods">
      <xs:complexType>
       <xs:choice>
        <xs:element name="builtinMethods">
         <xs:complexType>
          <xs:choice>
           <xs:element name="authenticateWithPassword">
            <xs:complexType>
             <xs:sequence>
              <xs:element name="protectedIdentityPattern" type="IdentityPattern" minOccurs="0"/>
              <xs:element name="passwordSource" type="PasswordSource"/>
              <xs:element name="methods">
               <xs:complexType>
                <xs:all>
                 <xs:element name="eapGtc" type="Empty" minOccurs="0"/>
                </xs:all>
```

```
                </xs:complexType>
               </xs:element>
              </xs:sequence>
             </xs:complexType>
            </xs:element>
            <xs:element name="authenticateWithToken">
             <xs:complexType>
              <xs:sequence>
               <xs:element name="protectedIdentityPattern" type="IdentityPattern" minOccurs="0"/>
               <xs:element name="tokenSource" type="TokenSource"/>
               <xs:element name="methods">
                <xs:complexType>
                 <xs:all>
                  <xs:element name="eapGtc" type="Empty"/>
                 </xs:all>
                </xs:complexType>
               </xs:element>
              </xs:sequence>
             </xs:complexType>
            </xs:element>
           </xs:choice>
          </xs:complexType>
         </xs:element>
        </xs:choice>
       </xs:complexType>
      </xs:element>
     </xs:sequence>
    </xs:extension>
   </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="IdentityPattern">
   <xs:simpleContent>
    <xs:extension base="NonEmptyString">
     <xs:attribute name="encryptContent" type="xs:boolean" use="optional" default="true">
```

```
      <xs:annotation>
       <xs:documentation> これは 'true' の場合のデフォルトです。要素が（XML セキュリティ エンベロープ内で）まだ暗号化されていない場合に、この要素を暗号化する後処理ツールを示します。
</xs:documentation>
      </xs:annotation>
     </xs:attribute>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="PasswordFromProfile">
   <xs:simpleContent>
    <xs:extension base="xs:string">
     <xs:attribute name="encryptContent" type="xs:boolean" use="optional" default="true">
      <xs:annotation>
       <xs:documentation> これは 'true' の場合のデフォルトです。要素が（XML セキュリティ エンベロープ内で）まだ暗号化されていない場合に、この要素を暗号化する後処理ツールを示します。
</xs:documentation>
      </xs:annotation>
     </xs:attribute>
    </xs:extension>
   </xs:simpleContent>
  </xs:complexType>

  <xs:complexType name="PasswordSource">
   <xs:choice>
    <xs:element name="passwordFromLogon" type="Empty"/>
    <xs:element name="passwordFromUser" type="Empty"/>
    <xs:element name="passwordFromProfile" type="PasswordFromProfile"/>
   </xs:choice>
  </xs:complexType>

  <xs:complexType name="TokenSource">
   <xs:choice>
    <xs:element name="passwordFromOtherToken" type="Empty">
     <xs:annotation>
```

<xs:documentation> これにより、トークンから ID と OTP を取得するためのプロンプトがユーザに表示されます。</xs:documentation>

</xs:annotation>

</xs:element>

</xs:choice>

</xs:complexType>

<xs:complexType name="TunnelMethods">

<xs:sequence>

<xs:choice>

<xs:element name="validateServerCertificate" type="serverCertificateValidationParameters"/>

<xs:element name="doNotValidateServerCertificate" type="Empty"/>

</xs:choice>

<xs:element name="unprotectedIdentityPattern" type="IdentityPattern" minOccurs="0">

<xs:annotation>

<xs:documentation>[username] および [domain]、またはそのどちらかのプレースホルダが使用される場合：認証でクライアント証明書が使用される場合、プレースホルダの値はクライアント証明書の CN フィールドから取得されます。クレデンシャルがエンドユーザから取得される場合、プレースホルダの値はユーザが入力した情報から取得されます。クレデンシャルがオペレーティングシステムから取得される場合、プレースホルダの値はログオン情報から取得されます。
</xs:documentation>

</xs:annotation>

</xs:element>

<xs:choice>

<xs:element name="enableFastReconnect">

<xs:complexType>

<xs:complexContent>

<xs:extension base="Empty">

<xs:choice>

<xs:element name="alwaysAttempt" type="Empty"/>

</xs:choice>

</xs:extension>

</xs:complexContent>

</xs:complexType>

</xs:element>

<xs:element name="disableFastReconnect" type="Empty"/>

</xs:choice>

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

 <xs:complexType name="CertificateContainer">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
   <xs:element name="certificateId" type="CertificateIdentifier"/>
  </xs:choice>
 </xs:complexType>

 <xs:complexType name="CertificateIdentifier">
  <xs:simpleContent>
   <xs:annotation>
    <xs:documentation>X509 形式のバイナリ証明書全体にわたる SHA 1 のハッシュ値。マシンの信頼済み CA のグローバル リストで証明書を一意に識別します（Windows の OS で管理されるストア）。</xs:documentation>
   </xs:annotation>
   <xs:extension base="NonEmptyString">
    <xs:attribute name="reference" type="xs:boolean">
     <xs:annotation>
      <xs:documentation>true は、これが PEM 形式の証明書へのファイル参照であることを示します。false は、これがその証明書にわたる SHA1 のハッシュ値であることを示します。このため、管理者はハッシュの検索、切り取り、貼り付けを行う必要はありません。ファイルを指定するだけで、後処理ツールがファイルをハッシュに変換します。</xs:documentation>
     </xs:annotation>
    </xs:attribute>
   </xs:extension>
  </xs:simpleContent>
 </xs:complexType>

 <xs:complexType name="Empty"/>

 <xs:simpleType name="NonEmptyString">
  <xs:restriction base="xs:string">
   <xs:minLength value="1"/>
  </xs:restriction>
 </xs:simpleType>
```

```
<xs:complexType name="ServerRuleFormat">
 <xs:simpleContent>
  <xs:extension base="NonEmptyString">
   <xs:attribute name="match" use="required">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:enumeration value="exactly"/>
      <xs:enumeration value="endsWith"/>
     </xs:restriction>
    </xs:simpleType>
   </xs:attribute>
  </xs:extension>
 </xs:simpleContent>
</xs:complexType>

<xs:complexType name="ServerValidationRules">
 <xs:choice minOccurs="0" maxOccurs="unbounded">
  <xs:annotation>
   <xs:documentation>
```

これはオプションです。Vista 製品により、サーバ検証ルールを持たないプロファイルの開始が許可されると、ユーザが信頼されていないサーバを検証するときに、検証プロセスによってそのサーバ名が検証されます。</xs:documentation>

```
  </xs:annotation>
  <xs:element name="matchSubjectAlternativeName" type="ServerRuleFormat"/>
  <xs:element name="matchSubject" type="ServerRuleFormat"/>
 </xs:choice>
</xs:complexType>

<xs:complexType name="serverCertificateValidationParameters">
 <xs:sequence>
  <xs:choice>
   <xs:element name="serverNameValidationRules" type="ServerValidationRules"/>
   <xs:element name="anyServerName" type="Empty">
    <xs:annotation>
```

```
        <xs:documentation> 証明書内のサーバ名はテストされません。</xs:documentation>
       </xs:annotation>
      </xs:element>
     </xs:choice>
     <xs:choice>
      <xs:element name="validateChainWithSpecificCa">
       <xs:complexType>
        <xs:complexContent>
         <xs:extension base="CertificateContainer"/>
        </xs:complexContent>
       </xs:complexType>
      </xs:element>
      <xs:element name="validateChainWithAnyCaFromOs" type="Empty">
       <xs:annotation>
        <xs:documentation> グローバル CA 証明書ストアの CA 証明書で終わっている場合、その証明書チェーンは信頼されます。</xs:documentation>
       </xs:annotation>
      </xs:element>
     </xs:choice>
     <xs:element name="userValidatesUntrustedServerCertificate" type="xs:boolean">
      <xs:annotation>
       <xs:documentation> サーバ証明書の検証に失敗した場合、true に設定されていると、エンドユーザはサーバを検証するように求められます。検証すると、適切な trustedCaCerts およびサーバ名フィールドが記憶され、次回から自動的に信頼されるようになります。</xs:documentation>
      </xs:annotation>
     </xs:element>
    </xs:sequence>
   </xs:complexType>

</xs:schema>
```

LEAP XML スキーマ

LEAP モジュールは、次のスキーマを使用することで、ネットワーク プロファイルのネイティブ EAP 方式セクションのすべての設定を XML として保存します。

```
<?xml version="1.0"?>

<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns="http://www.cisco.com/CCX"
 targetNamespace="http://www.cisco.com/CCX"
 elementFormDefault="qualified"
 attributeFormDefault="unqualified">

 <xs:element name="eapLeap" type="EapLeap"/>

 <xs:complexType name="EapLeap">
  <xs:complexContent>
   <xs:extension base="PasswordMethods"/>
  </xs:complexContent>
 </xs:complexType>

 <xs:complexType name="IdentityPattern">
  <xs:simpleContent>
   <xs:extension base="NonEmptyString">
    <xs:attribute name="encryptContent" type="xs:boolean" use="optional" default="true">
     <xs:annotation>
      <xs:documentation> これは 'true' の場合のデフォルトです。要素が（XML セキュリティ エンベロープ内で）まだ暗号化されていない場合に、この要素を暗号化する後処理ツールを示します。
</xs:documentation>
     </xs:annotation>
    </xs:attribute>
   </xs:extension>
  </xs:simpleContent>
 </xs:complexType>
```

```
 <xs:complexType name="PasswordFromProfile">
  <xs:simpleContent>
   <xs:extension base="xs:string">
    <xs:attribute name="encryptContent" type="xs:boolean" use="optional" default="true">
     <xs:annotation>
      <xs:documentation> これは 'true' の場合のデフォルトです。要素が（XML セキュリティ エンベロープ内で）まだ暗号化されていない場合に、この要素を暗号化する後処理ツールを示します。
</xs:documentation>
     </xs:annotation>
    </xs:attribute>
   </xs:extension>
  </xs:simpleContent>
 </xs:complexType>

 <xs:complexType name="PasswordSource">
  <xs:choice>
   <xs:element name="passwordFromLogon" type="Empty"/>
   <xs:element name="passwordFromUser" type="Empty"/>
   <xs:element name="passwordFromProfile" type="PasswordFromProfile"/>
  </xs:choice>
 </xs:complexType>

 <xs:complexType name="PasswordMethods">
  <xs:sequence>
   <xs:element name="unprotectedIdentityPattern" type="IdentityPattern" minOccurs="0"/>
   <xs:element name="passwordSource" type="PasswordSource"/>
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="Empty"/>

 <xs:simpleType name="NonEmptyString">
  <xs:restriction base="xs:string">
   <xs:minLength value="1"/>
  </xs:restriction>
```

```
</xs:simpleType>

</xs:schema>
```

EAP モジュールのロギング

トラブルシューティングに役立つログを生成するために、EAP-FAST、LEAP、PEAP-GTC モジュールは、Windows イベント ログ サービスを使用します。ログには、イベントのタイプ、イベントの発生場所、イベントの影響を受けた機能、イベントの発生日時などの情報が記録されます。

次の項で、ロギングの詳細について説明します。

ロギングの設定と開始

管理者のコマンド プロンプトにアクセスし、ロギングを設定および開始するには、次の手順を実行します。

ステップ 1　**Start**、**All Programs**、**Accessories** の順にクリックします。

ステップ 2　**Command Prompt** を右クリックし、**Run as administrator** を選択します。

ステップ 3　プロンプトで、次のコマンドを入力し、ロギングを設定および開始します。

- EAP-FAST の場合
  wevtutil sl Cisco-EAP-FAST/Debug /e:true /k:*category_mask* **/l:***log_level*
- PEAP-GTC の場合
  wevtutil sl Cisco-EAP-PEAP/Debug /e:true /k:*category_mask* **/l:***log_level*
- LEAP の場合
  wevtutil sl Cisco-EAP-LEAP/Debug /e:true /k:*category_mask* **/l:***log_level*

シンタックスの説明

category_mask	有効にするロギングのカテゴリのビットマスク。有効な値は次のとおりです。 • **0** — すべてのカテゴリをロギングします。 • **1** — 次の 2 つのカテゴリに該当しないすべてのメッセージをロギングします。 • **2** — 詳細ログ レベルでのみ、戻りコードとともに、機能のエントリ ポイントおよびエグジット ポイントのフローをロギングします。 • **4** — 詳細ログ レベルでのみ、パケット ダンプをロギングします。 デフォルト値は 0 です。
log_level	有効にするロギングのレベル。有効な値は次のとおりです。 • **0** — すべてのログ レベル • **1** — 重大 • **2** — エラー • **3** — 警告 • **4** — 情報 • **5** — 詳細 デフォルト値は 0 です。

（注） ロギングが終了する前に、ロギングを実行しているデバイスをシャットダウンする必要がある場合、ロギングはリブート後に再開されます。ただし、ロギングが自動または手動で開始されると、ログはクリアされます。

ロギングの無効化と内部バッファのフラッシュ

必要な情報を収集した後、次のコマンドを使用すると、ロギングを停止し、すべての内部バッファをフラッシュすることができます。

- EAP-FAST の場合
  wevtutil sl Cisco-EAP-FAST/Debug /e:false
- PEAP-GTC の場合
  wevtutil sl Cisco-EAP-PEAP/Debug /e:false
- LEAP の場合
  wevtutil sl Cisco-EAP-LEAP/Debug /e:false

（注） .etl ファイルを分析するには、このコマンドを入力する必要があります。

ログ ファイルの場所

デフォルトでは、分析とデバッグに使用できる .etl ファイルは次の場所に作成されます。

C:¥Windows¥System32¥Winevt¥Logs¥Cisco-EAP-FAST%4Debug.etl

この場所を変更する場合は、管理者のコマンド プロンプトで次のコマンドを入力します。

- EAP-FAST の場合
  wevtutil sl Cisco-EAP-FAST/Debug /lfn:"*path_to_etl_log_file***"**
- PEAP-GTC の場合
  wevtutil sl Cisco-EAP-PEAP/Debug /lfn:"*path_to_etl_log_file***"**
- LEAP の場合
  wevtutil sl Cisco-EAP-LEAP/Debug /lfn:"*path_to_etl_log_file***"**

（注） ログ ファイルへのパスを変更するコマンドは、ロギングの実行中に入力しないでください。

また、ロギングを開始するときに、.etl ファイルへのパスを変更することもできます。.etl ファイルの場所を指定してロギングを開始するには、管理者のコマンド プロンプトで次のコマンドを入力します。

- EAP-FAST の場合
  wevtutil sl Cisco-EAP-FAST/Debug /e:true /lfn:"*path_to_etl_log_file***"**
- PEAP-GTC の場合
  wevtutil sl Cisco-EAP-PEAP/Debug /e:true /lfn:"*path_to_etl_log_file***"**
- LEAP の場合
  wevtutil sl Cisco-EAP-LEAP/Debug /e:true /lfn:"*path_to_etl_log_file***"**