CHAPTER 3

トラブルシューティング用の API コールの使用方法

Cisco ISE トラブルシューティング API コールは、ステータス要求をターゲット モニタリング ノードに送信し、次の診断関連情報を取得します。

- ノードのバージョンおよびタイプ（Version API コールを使用）
- 障害理由（FailureReasons API コールを使用）
- 認証ステータス（AuthStatus API コールを使用）
- アカウンティング ステータス（AcctStatus API コールを使用）

次の項では、トラブルシューティング API コールの各タイプを説明するほか、ファイルの例、各コール発行の手順、および返されるデータのサンプルが記載されています。

Version API コールの使用方法

各ノードの Representational State Transfer（REST）プログラミング インターフェイス（PI）サービスとクレデンシャルをテストするには Version API コールを使用します。ここでは、スキーマ ファイルの出力例、この API コールを呼び出すことにより、Cisco ISE ソフトウェアのバージョンおよびノード タイプを要求する手順、この API コール発行後に返されるノードのバージョンとタイプのサンプルについて説明します。

各ノード タイプには関連付けられた値があり、次のいずれかを指定できます。

- STANDALONE_MNT_NODE = 0
- ACTIVE_MNT_NODE = 1
- BACKUP_MNT_NODE = 2
- NOT_AN_MNT_NODE = 3

Version API コールのスキーマ ファイル

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

```
  <xs:element name="product" type="product"/>

  <xs:complexType name="product">
    <xs:sequence>
      <xs:element name="version" type="xs:string" minOccurs="0"/>
  <xs:element name="type_of_node" type="xs:int"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

Version API コールの呼び出し

（注） ターゲット ノードが有効なモニタリング ノードであることを確認します。Cisco ISE ノードのペルソナを確認するには、「モニタリング ノードの確認」（P.1-2）を参照してください。

ステップ 1　ターゲット モニタリング ノードにログインします。

たとえば、ホスト名が acme123 のモニタリング ノードに最初にログインすると、次の URL アドレス フィールドが表示されます。

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

ステップ 2　「/admin/」コンポーネントを API コールのコンポーネント（/ise/mnt/<specific-api-call>）に置き換えて、ターゲット ノードの URL アドレス フィールドに Version API コールを入力します。

```
https://acme123/ise/mnt/Version
```

（注） これらのコールは、大文字小文字を区別するため、ターゲット ノードの URL アドレス フィールドに慎重に各 API コールを入力する必要があります。API コール規則での「mnt」の使用は、モニタリング ノードを表します。

ステップ 3　Enter キーを押して API コールを発行します。

Version API コール データ

次の例では、Version API コールはターゲット ノードに関する次の情報を返しました。

- ノードのバージョン：この例では、1.0.3.032 を表示します。
- モニタリング ノードのタイプ：この例では、1 を表示し、ノードがアクティブであることを意味します。

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<product name="Cisco Identity Services Engine">
<version>1.0.3.032</version>
<type_of_node>1</type_of_node>
</product>
```

FailureReasons API コールの使用方法

FailureReasons API コールを使用して、表 3-1 で説明されている失敗した操作と可能な解決策のリストを返します。

（注） Cisco ISE Failure Reasons Editor を使用して操作の失敗の完全なリストにアクセスする方法に関する詳細については、「障害理由レポートの使用方法」（P.A-1）を参照してください。

表 3-1　FailureReasons API コールから返されるデータ

要素	例
障害理由 ID	<failureReason id="11011">
コード	<11011 RADIUS listener failed>
原因	<Could not open one or more of the ports used to receive RADIUS requests>
解決策	<Ensure that ports 1812, 1813, 1645 and 1646 are not being used by another process on the system>

FailureReasons API コールのスキーマ ファイル

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="failureReasonList" type="failureReasonList"/>

  <xs:complexType name="failureReasonList">
    <xs:sequence>
      <xs:element name="failureReason" type="failureReason" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="failureReason">
    <xs:sequence>
      <xs:element name="code" type="xs:string" minOccurs="0"/>
      <xs:element name="cause" type="xs:string" minOccurs="0"/>
      <xs:element name="resolution" type="xs:string" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="xs:string"/>
  </xs:complexType>
</xs:schema>
```

FailureReasons API コールの呼び出し

（注） API コールを発行するターゲット ノードが、有効なモニタリング ノードであることを確認します。Cisco ISE ノードのペルソナを確認するには、「モニタリング ノードの確認」（P.1-2）を参照してください。

ステップ 1　ターゲット モニタリング ノードにログインします。

たとえば、ホスト名が acme123 のモニタリング ノードに最初にログインすると、次の URL アドレス フィールドが表示されます。

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

ステップ 2　「/admin/」コンポーネントを API コールのコンポーネント（/ise/mnt/<specific-api-call>）に置き換えて、ターゲット ノードの URL アドレス フィールドに FailureReasons API コールを入力します。

```
https://acme123/ise/mnt/FailureReasons
```

（注）　コールは、大文字小文字を区別するため、ターゲット ノードの URL アドレス フィールドに慎重に各 API コールを入力する必要があります。API コール規則での「mnt」の使用は、モニタリング ノードを表します。

ステップ 3　Enter キーを押して API コールを発行します。

FailureReasons API コール データ

（注）　次の FailureReasons API コールの例は、返されるデータの小規模なサンプルを表示しています。

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<failureReasonList>
-
<failureReason id="100001">
-
<code>
100001 AUTHMGR-5-FAIL Authorization failed for client
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100002">
-
<code>
100002 AUTHMGR-5-SECURITY_VIOLATION Security violation on the interface
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100003">
-
<code>
100003 AUTHMGR-5-UNAUTHORIZED Interface unauthorized
```

```
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100004">
-
<code>
100004 DOT1X-5-FAIL Authentication failed for client
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100005">
<code>100005 MAB-5-FAIL Authentication failed for client</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100006">
-
<code>
100006 RADIUS-4-RADIUS_DEAD RADIUS server is not responding
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
-
<failureReason id="100007">
-
<code>
100007 EPM-6-POLICY_APP_FAILURE Interface ACL not configured
</code>
<cause>This may or may not be indicating a violation</cause>
-
<resolution>
Please review and resolve according to your organization's policy
</resolution>
</failureReason>
```

Cisco ISE Failure Reasons Editor の詳細については、付録 A「障害理由レポートの使用方法」を参照してください。

AuthStatus API コールの使用方法

ターゲット ノードにおけるセッションの認証ステータスをチェックするために AuthStatus API コールを使用します。クエリーに対して少なくとも 1 つの MAC アドレスを指定する必要があります。

ここでは、次の項目について説明します。

- 「AuthStatus API コールのスキーマ ファイル」（P.3-7）
- 「AuthStatus API コールの呼び出し」（P.3-9）
- 「AuthStatus API コール データ」（P.3-9）

次の検索関連パラメータを設定できます。

- 期間 : 指定した MAC アドレスに関連付けられた認証ステータス レコードの検索と取得が試行される秒数を定義します。ユーザが設定可能な値の有効範囲は 1 ～ 864000 秒（10 日）です。0 秒の値を入力した場合は、デフォルト期間の 10 日が指定されます。
- レコード : MAC アドレスごとに検索するセッションのレコード数を定義します。ユーザが設定可能な値の有効範囲は 1 ～ 500 レコードです。0 を入力した場合は、デフォルト設定の 200 レコードが指定されます。

  （注）　期間パラメータおよびレコード パラメータの両方に値 0 を指定すると、この AuthStatus API コールは、指定された MAC アドレスに関連付けられている最新の認証セッション レコードのみを返します。

  ここに、期間とレコードの属性を指定した URL の一般的な形式の例を示します。

  https://10.10.10.10/ise/mnt/AuthStatus/MACAddress/01:23:45:67:89:98/900000/2/All

- 属性 : AuthStatus API コールを使用して認証ステータスの検索で返された認証ステータスのテーブルの属性数を定義します。有効な値は 0（デフォルト）、All、または user_name+acs_timestamp です（AuthStatus スキーマの例「AuthStatus API コールのスキーマ ファイル」（P.3-7）を参照）。
  - 「0」を入力すると、表 3-2 で定義された属性が返されます。これらは出力スキーマの restAuthStatus のセクションに記載されています。
  - 「All」を入力すると、より詳しい属性セットが返されます。これらは出力スキーマの fullRESTAuthStatus のセクションに記載されています。
  - user_name+acs_timestamp のスキーマに示されている値を入力すると、それらの属性だけが返されます。user_name 属性と acs_timestamp 属性は、出力スキーマ restAuthStatus のセクションに記載されています。

表 3-2　　認証ステータス テーブルの属性

属性	説明
name = "passed" または name = "failed"	認証ステータスの結果 : • 合格 • 不合格
name = "user_name"	ユーザ名
name = "nas_ip_address"	ネットワーク アクセス デバイスの IP アドレス / ホスト名
name = "failure_reason"	セッション認証に失敗した原因
name = "calling_station_id"	送信元 IP アドレス
name = "nas_port"	ネットワーク アクセス サーバ ポート

表 3-2　認証ステータス テーブルの属性 （続き）

属性	説明
name = "identity_group"	関連するユーザとホストで構成される論理グループ
name = "network_device_name"	ネットワーク デバイスの名前
name = "acs_server"	Cisco ISE アプライアンスの名前
name = "eap_authentication"	認証要求に使用する拡張認証プロトコル（EAP）方式
name = "framed_ip_address"	特定のユーザに設定されたアドレス
name = "network_device_groups"	関連するネットワーク デバイスで構成される論理グループ
name = "access_service"	アプリケーション アクセス サービス
name = "acs_timestamp"	Cisco ISE 認証要求に関連付けられたタイム スタンプ
name = "authentication_method"	認証で使用される方式を指定します
name = "execution_steps"	要求の処理中にログに記録された各診断メッセージのメッセージ コードのリスト
name = "radius_response"	RADIUS 応答のタイプ（例：VLAN、ACL）
name = "audit_session_id"	認証セッションの ID
name = "nas_identifier"	特定のリソースに関連づけられているネットワーク アクセス サーバ（NAS）
name = "nas_port_id"	使用される NAS ポート ID
name = "nac_policy_compliance"	ポスチャ ステータスを示します（準拠または非準拠）
name = "selected_azn_profiles"	認証に使用されるプロファイルを指定します
name = "service_type"	フレームド ユーザを示します
name = "eap_tunnel"	EAP 認証に使用されるトンネルまたは外部方式
name = "message_code"	処理された要求の結果を定義する監査のメッセージを識別します
name = "destination_ip_address"	宛先 IP アドレスを指定します

AuthStatus API コールのスキーマ ファイル

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="authStatusOutputList" type="fullRESTAuthStatusOutputList"/>

  <xs:complexType name="fullRESTAuthStatusOutputList">
    <xs:sequence>
      <xs:element name="authStatusList" type="fullRESTAuthStatusList" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="fullRESTAuthStatusList">
    <xs:sequence>
      <xs:element name="authStatusElements" type="fullRESTAuthStatus" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="key" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="fullRESTAuthStatus">
    <xs:complexContent>
```

```
      <xs:extension base="restAuthStatus">
        <xs:sequence>
          <xs:element name="id" type="xs:long" minOccurs="0"/>
          <xs:element name="acsview_timestamp" type="xs:dateTime" minOccurs="0"/>
          <xs:element name="acs_session_id" type="xs:string" minOccurs="0"/>
          <xs:element name="service_selection_policy" type="xs:string" minOccurs="0"/>
          <xs:element name="authorization_policy" type="xs:string" minOccurs="0"/>
          <xs:element name="identity_store" type="xs:string" minOccurs="0"/>
          <xs:element name="response" type="xs:string" minOccurs="0"/>
          <xs:element name="cts_security_group" type="xs:string" minOccurs="0"/>
          <xs:element name="use_case" type="xs:string" minOccurs="0"/>
          <xs:element name="cisco_av_pair" type="xs:string" minOccurs="0"/>
          <xs:element name="ad_domain" type="xs:string" minOccurs="0"/>
          <xs:element name="acs_username" type="xs:string" minOccurs="0"/>
          <xs:element name="radius_username" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_role" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_username" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_posture_token" type="xs:string" minOccurs="0"/>
          <xs:element name="nac_radius_is_user_auth" type="xs:string" minOccurs="0"/>
          <xs:element name="selected_posture_server" type="xs:string" minOccurs="0"/>
          <xs:element name="selected_identity_store" type="xs:string" minOccurs="0"/>
          <xs:element name="authentication_identity_store" type="xs:string"

minOccurs="0"/>
          <xs:element name="azn_exp_pol_matched_rule" type="xs:string" minOccurs="0"/>
          <xs:element name="ext_pol_server_matched_rule" type="xs:string" minOccurs="0"/>
          <xs:element name="grp_mapping_pol_matched_rule" type="xs:string" minOccurs="0"/>
          <xs:element name="identity_policy_matched_rule" type="xs:string" minOccurs="0"/>
          <xs:element name="nas_port_type" type="xs:string" minOccurs="0"/>
          <xs:element name="query_identity_stores" type="xs:string" minOccurs="0"/>
          <xs:element name="sel_exp_azn_profiles" type="xs:string" minOccurs="0"/>
          <xs:element name="selected_query_identity_stores" type="xs:string"
minOccurs="0"/>
          <xs:element name="tunnel_details" type="xs:string" minOccurs="0"/>
          <xs:element name="cisco_h323_attributes" type="xs:string" minOccurs="0"/>
          <xs:element name="cisco_ssg_attributes" type="xs:string" minOccurs="0"/>
          <xs:element name="other_attributes" type="xs:string" minOccurs="0"/>
          <xs:element name="response_time" type="xs:long" minOccurs="0"/>
          <xs:element name="nad_failure" type="xs:anyType" minOccurs="0"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="restAuthStatus">
    <xs:sequence>
      <xs:element name="passed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="failed" type="xs:anyType" minOccurs="0"/>
      <xs:element name="user_name" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="failure_reason" type="xs:string" minOccurs="0"/>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port" type="xs:string" minOccurs="0"/>
      <xs:element name="identity_group" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_name" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_server" type="xs:string" minOccurs="0"/>
      <xs:element name="eap_authentication" type="xs:string" minOccurs="0"/>
      <xs:element name="framed_ip_address" type="xs:string" minOccurs="0"/>
      <xs:element name="network_device_groups" type="xs:string" minOccurs="0"/>
      <xs:element name="access_service" type="xs:string" minOccurs="0"/>
      <xs:element name="acs_timestamp" type="xs:dateTime" minOccurs="0"/>
      <xs:element name="authentication_method" type="xs:string" minOccurs="0"/>
      <xs:element name="execution_steps" type="xs:string" minOccurs="0"/>
      <xs:element name="radius_response" type="xs:string" minOccurs="0"/>
```

```
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_identifier" type="xs:string" minOccurs="0"/>
      <xs:element name="nas_port_id" type="xs:string" minOccurs="0"/>
      <xs:element name="nac_policy_compliance" type="xs:string" minOccurs="0"/>
      <xs:element name="selected_azn_profiles" type="xs:string" minOccurs="0"/>
      <xs:element name="service_type" type="xs:string" minOccurs="0"/>
      <xs:element name="eap_tunnel" type="xs:string" minOccurs="0"/>
      <xs:element name="message_code" type="xs:string" minOccurs="0"/>
      <xs:element name="destination_ip_address" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

AuthStatus API コールの呼び出し

（注） API コールを発行するターゲット ノードが、有効なモニタリング ノードであることを確認します。Cisco ISE ノードのペルソナを確認するには、「モニタリング ノードの確認」（P.1-2）を参照してください。

ステップ 1　ターゲット モニタリング ノードにログインします。

たとえば、ホスト名が acme123 のモニタリング ノードに最初にログインすると、次の URL アドレス フィールドが表示されます。

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

ステップ 2　「/admin/」コンポーネントを API コールのコンポーネント（/ise/mnt/<specific-api-call>/MACAddress/<macaddress>/<seconds>/<numberofrecordspermacaddress>/All）に置き換えて、ターゲット ノードの URL アドレス フィールドに AuthStatus API コールを入力します。

```
https://acme123/ise/mnt/AuthStatus/MACAddress/00:50:56:10:13:02/120/100/All
```

（注） API コールは大文字と小文字が区別されます。API コール規則での「mnt」の使用は、ターゲット モニタリング ノードを表します。

ステップ 3　Enter キーを押して API コールを発行します。

AuthStatus API コール データ

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<authStatusOutputList>
-
<authStatusList key="00:0C:29:46:F3:B8"><authStatusElements>
-
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>suser77</user_name>
<nas_ip_address>10.77.152.209</nas_ip_address>
<calling_station_id>00:0C:29:46:F3:B8</calling_station_id>
<identity_group>User Identity Groups:Guest</identity_group>
```

```
<acs_server>guest-240</acs_server>
<acs_timestamp>2012-10-05T10:50:56.515Z</acs_timestamp>
<execution_steps>5231</execution_steps>
<message_code>5231</message_code>
<id>1349422277270561</id>
<acsview_timestamp>2012-10-05T10:50:56.517Z</acsview_timestamp>
<identity_store>Internal Users</identity_store>
<response_time>146</response_time>
<other_attributes>ConfigVersionId=81,EndPointMACAddress=00-0C-29-46-F3-B8,PortalName=Defau
ltGuestPortal,
CPMSessionID=0A4D98D1000001F26F0C04D9,CiscoAVPair=</other_attributes>
</authStatusElements>
-
<authStatusElements>
<passed xsi:type="xs:boolean">true</passed>
<failed xsi:type="xs:boolean">false</failed>
<user_name>00:0C:29:46:F3:B8</user_name>
<nas_ip_address>10.77.152.209</nas_ip_address>
<calling_station_id>00:0C:29:46:F3:B8</calling_station_id>
<identity_group>Guest_IDG</identity_group>
<network_device_name>switch</network_device_name>
<acs_server>guest-240</acs_server>
<authentication_method>mab</authentication_method>
<authentication_protocol>Lookup</authentication_protocol>
<acs_timestamp>2012-10-05T10:49:47.915Z</acs_timestamp>
<execution_steps>11001,11017,11027,15049,15008,15048,15048,15004,15041,15006,15013,24209,2
421
1,22037,15036,15048,15004,15016,11022,11002</execution_steps>
<response>{UserName
=00:0C:29:46:F3:B8; User-Name=00-0C-29-46-F3-B8;
State=ReauthSession:0A4D98D1000001F26F0C04D9;
Class=CACS:0A4D98D1000001F26F0C04D9:guest-240/138796808/76;
Termination-Action=RADIUS-Request; Tunnel-Type=(tag=1) VLAN;
Tunnel-Medium-Type=(tag=1) 802; Tunnel-Private-Group-ID=(tag=1) 2;
cisco-av-pair=url-redirect-acl=ACL-WEBAUTH-REDIRECT;
cisco-av-pair=url-redirect=https://guest-240.cisco.com:8443/guestportal/gateway?
sessionId=0A4D98D1000001F26F0C04D9&action=cwa;
cisco-av-pair=ACS:CiscoSecure-Defined-ACL=#ACSACL#-IP-pre-posture-506e980a;
cisco-av-pair=profile-name=WindowsXP-Workstation;}</response
><audit_session_id>0A4D98D1000001F26F0C04D9</audit_session_id><nas_po
rt_id>GigabitEthernet1/0/17</nas_port_id><posture_status>Pending</posture_status>
<selected_azn_profiles>CWA_Redirect</selected_azn_profiles>
<service_type>Call Check</service_type>
<message_code>5200</message_code>
<nac_policy_compliance>Pending</nac_policy_compliance>
<id>1349422277270556</id>
<acsview_timestamp>2012-10-05T10:49:47.915Z</acsview_timestamp>
<identity_store>Internal Endpoints</identity_store>
<response_time>13</response_time>
<other_attributes>ConfigVersionId=81,DestinationPort=1812,Protocol=Radius,AuthorizationPol
icyMatchedRule=CWA_Redirect,
NAS-Port=50117,Framed-MTU=1500,NAS-Port-Type=Ethernet,EAP-Key-N
ame=,cisco-nas-port=GigabitEthernet1/0/17,AcsSessionID=guest-240/138796808/76,Us
eCase=Host Lookup,SelectedAuthenticationIdentityStores=Internal
Endpoints,ServiceSelectionMatchedRule=MAB,IdentityPolicyMatchedRule=Default,CPMS
essionID=0A4D98D1000001F26F0C04D9,EndPointMACAddress=00-0C-29-46-F3-B8,EndPointM
atchedProfile=WindowsXP-Workstation,ISEPolicySetName=Default,HostIdentityGroup=E
ndpoint Identity Groups:Guest_IDG,Device Type=Device Type#All Device
Types,Location=Location#All Locations,Device IP
Address=10.77.152.209,Called-Station-ID=00:24:F7:73:9A:91,CiscoAVPair=audit-sess
ion-id=0A4D98D1000001F26F0C04D9</other_attributes>
-
</authStatusElements>
-
```

```
</authStatusList>
-
</authStatusOutputList>
```

AcctStatus API コール データの使用方法

ターゲット ノードの最新のデバイスおよびセッションのアカウント情報を取得するために AcctStatus API コールを使用できます。ここでは、次の項目について説明します。

- 「AcctStatus API コールのスキーマ ファイル」（P.3-11）
- 「AcctStatus API コールの呼び出し」（P.3-12）
- 「AcctStatus API コール データ」（P.3-12）

次の時間関連パラメータを設定できます。

- 期間 : 指定された MAC アドレスに関連付けられた最新アカウントのデバイス レコードの検索と取得が試行される秒数を定義します。ユーザが設定可能な値の有効範囲は 1 ～ 432000 秒（5 日）です。
  - 2400 秒（40 分）の値を入力した場合は、指定 MAC アドレスの過去 40 分間のデバイス レコードが必要であることを意味します。
  - 0 秒の値を入力した場合は、デフォルト期間の 15 分（900 秒）を指定します。これは、指定 MAC アドレスの過去 15 分間のデバイス レコードが必要であることを意味します。

AcctStatus API コールは、API 出力として、次のアカウント ステータスのデータ フィールドを提供します。

表 3-3　アカウト ステータスのデータ フィールド

データ フィールド	説明
MAC アドレス	クライアントの MAC アドレス
audit-session-id	認証セッション ID
パケット入力	受信されたパケットの総数
パケット出力	送信されたパケットの総数
バイト入力	受信された合計バイト数
バイト出力	送信された合計バイト数
セッション時間	現在のセッションの期間

AcctStatus API コールのスキーマ ファイル

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<xs:schema version="1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="acctStatusOutputList" type="restAcctStatusOutputList"/>

  <xs:complexType name="restAcctStatusOutputList">
    <xs:sequence>
      <xs:element name="acctStatusList" type="restAcctStatusList" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="restAcctStatusList">
    <xs:sequence>
```

```
      <xs:element name="acctStatusElements" type="restAcctStatus" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="macAddress" type="xs:string"/>
    <xs:attribute name="username" type="xs:string"/>
  </xs:complexType>

  <xs:complexType name="restAcctStatus">
    <xs:sequence>
      <xs:element name="calling_station_id" type="xs:string" minOccurs="0"/>
      <xs:element name="audit_session_id" type="xs:string" minOccurs="0"/>
      <xs:element name="paks_in" type="xs:long" minOccurs="0"/>
      <xs:element name="paks_out" type="xs:long" minOccurs="0"/>
      <xs:element name="bytes_in" type="xs:long" minOccurs="0"/>
      <xs:element name="bytes_out" type="xs:long" minOccurs="0"/>
      <xs:element name="session_time" type="xs:long" minOccurs="0"/>
      <xs:element name="username" type="xs:string" minOccurs="0"/>
  <xs:element name="server" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

AcctStatus API コールの呼び出し

（注） API コールを発行するターゲット ノードが、有効なモニタリング ノードであることを確認します。Cisco ISE ノードのペルソナを確認するには、「モニタリング ノードの確認」（P.1-2）を参照してください。

ステップ 1　ターゲット モニタリング ノードにログインします。

たとえば、ホスト名が acme123 のモニタリング ノードに最初にログインすると、次の URL アドレス フィールドが表示されます。

```
https://acme123/admin/LoginAction.do#pageId=com_cisco_xmp_web_page_tmpdash
```

ステップ 2　「/admin/」コンポーネントを API コールのコンポーネント（/ise/mnt/<specific-api-call>/MACAddress/<macaddress>/ <durationofcurrenttime>）に置き換えて、ターゲット ノードの URL アドレス フィールドに AcctStatus API コールを入力します。

```
https://acme123/ise/mnt/AcctStatus/MACAddress/00:26:82:7B:D2:51/1200
```

（注） これらのコールは、大文字小文字を区別するため、ターゲット ノードの URL アドレス フィールドに慎重に各 API コールを入力する必要があります。API コール規則での「mnt」の使用は、モニタリング ノードを表します。

ステップ 3　Enter キーを押して API コールを発行します。

AcctStatus API コール データ

```
This XML file does not appear to have any style information associated with it. The
document tree is shown below.

-
<acctStatusOutputList>
```

```
-
<acctStatusList macAddress="00:25:9C:A3:7D:48">
-
<acctStatusElements>
<calling_station_id>00:25:9C:A3:7D:48</calling_station_id>
<audit_session_id>0acb6b0b0000000B4D0C0DBD</audit_session_id>
<paks_in>0</paks_in>
<paks_out>0</paks_out>
<bytes_in>0</bytes_in>
<bytes_out>0</bytes_out>
<session_time>240243</session_time>
<server>HAREESH-R6-1-PDP1</server>
</acctStatusElements>
</acctStatusList>
</acctStatusOutputList>
```