



Autoconf の設定

ここでは、Autoconf に関する情報と Autoconf の設定方法について説明します。

- [Autoconf の前提条件](#) (1 ページ)
- [Autoconf の制約事項](#) (1 ページ)
- [Autoconf に関する情報](#) (2 ページ)
- [Autoconf の設定方法](#) (8 ページ)
- [Autoconf の設定例](#) (15 ページ)
- [Autoconf のその他の参考資料](#) (16 ページ)
- [Autoconf の機能履歴](#) (16 ページ)

Autoconf の前提条件

- Autoconf を有効にする前に、Auto SmartPort (ASP) マクロ、デバイス分類子を無効にしてから、セッションモニターにアクセスします。

Autoconf の制約事項

- ASP マクロと Autoconf は、同じインターフェイスでは同時にサポートされません。Autoconf または ASP のいずれかをインターフェイスごとのレベルで無効にする必要があります。
- インターフェイス テンプレートは、ワイヤレスセッションには適用されません。
- **autoconf enable** コマンドを使用して Autoconf 機能を有効にすると、デフォルトの Autoconf サービスポリシーがすべてのインターフェイスに適用されます。**service-policy** コマンドを使用して他のサービスポリシーをグローバルに適用することはできません。別のサービスポリシーを適用するには、そのインターフェイスで Autoconf を無効にする必要があります。サービスポリシーをグローバルに適用する場合は、Autoconf 機能を無効にしてから有効にする必要があります。
- ローカル (インターフェイスレベル) ポリシーとグローバルサービスポリシーの両方が存在する場合、ローカルポリシーが優先されます。ローカルサービスポリシー内のイベント

が処理され、グローバルサービスポリシーは適用されません。グローバルサービスポリシーは、ローカルポリシーが削除された場合にのみ有効になります。

- サービステンプレートはインターフェイスに適用できません。また、インターフェイステンプレートはサービスインスタンスに適用できません。
- インターフェイステンプレート内にネストできるサービステンプレートは1つだけです。

Autoconf に関する情報

ここでは、Autoconf について説明します。

Autoconf の利点

Autoconf機能により、エンドデバイスとインターフェイス間のハードバインドが可能になります。Autoconf は、Smart Operations ソリューションに含まれます。Smart Operations は、LAN スイッチの導入を簡素化し改善できる包括的な機能セットです。Smart Operations は、組織が優れた運用を実現し、ネットワーク上でサービスを拡張できるように支援します。

Autoconf 機能は、デバイスポートに必要な設定を自動的に適用し、インターフェイス テンプレート内で設定された一連のインターフェイス設定を使用して、直接接続された各エンドデバイスの効率的なパフォーマンスを実現します。

- Autoconf は、パーサーが毎回各コマンドを解析する必要がないため、コマンドをインターフェイスに効率的に適用します。
- Autoconf機能を使用して適用された設定は、ポートの以前の設定または後続の設定に影響を与えることなく、確実にポートから削除できます。
- Autoconf機能は、インターフェイスおよびサービステンプレートを使用して、組み込みの設定およびユーザー定義の設定を提供します。テンプレートを使用して適用された設定は、1回の操作で一元的に更新できます。
- Autoconf 機能を使用すると、ポートおよびアクセスセッションに設定を適用できます。
- Autoconf機能は、デバイスと接続されたエンドデバイスを直感的で自動設定可能にすることで、継続的なメンテナンスを削減します。これにより、運用コスト（OPEX）が削減され、総所有コスト（TCO）が削減されます。

アイデンティティセッション管理とテンプレート

Autoconf機能の主な利点は、コアセッション管理機能がアプリケーション固有のロジックから分離されていることです。これにより、ポリシー決定の基準や適用されるポリシーの性質に関係なく、同じフレームワークを使用できます。

アイデンティティセッション管理インフラストラクチャを使用すると、設定やポリシーをテンプレートとして適用できます。

サービステンプレートとインターフェイステンプレートは両方とも、設定とポリシーの名前付きコンテナです。サービステンプレートはアクセスセッションにのみ適用でき、インターフェイステンプレートはポートにのみ適用できます。サービステンプレートがアクセスセッションに適用されると、含まれる設定/ポリシーはターゲットセッションにのみ適用され、同じアクセスポートでホストされる可能性のある他のセッションには影響しません。同様に、インターフェイステンプレートがアクセスポートに適用されると、そのポートで交換されるすべてのトラフィックに影響します。

Autoconf機能は、一連の組み込みマップと組み込みテンプレートを使用します。組み込みテンプレートは、インターフェイス設定のベストプラクティスに基づいて設計されています。組み込みテンプレートは、カスタマイズされた設定を含めるようにユーザーが変更できるため、新しいテンプレートを作成する必要がありません。

ユーザーが作成したテンプレートは、ユーザー定義テンプレートと呼ばれます。ユーザー定義のテンプレートはデバイス上で定義でき、任意の組み込みトリガーまたはユーザー定義トリガーにマッピングできます。

Autoconfテンプレートと手動設定によって適用される全体的な適用設定を表示するには、**show derived-config** コマンドを使用します。**show running-config interface type number** コマンドの出力に表示されるインターフェイスコマンドは、必ずしも動作設定ではありません。Autoconf機能は、インターフェイスにテンプレートを動的に適用し、すでに適用されている競合する静的設定を上書きします。

Autoconf の動作

Autoconfは、デバイス分類子を使用して、ポートに接続されているエンドデバイスを識別します。

Autoconf機能は、Cisco Discovery Protocol、LLDP、DHCP、MACアドレスから収集したデバイス分類情報、およびデバイス分類子によって識別される組織固有識別子（OUI）を使用します。

デバイス分類子は、改善されたデバイス分類機能および精度、拡張デバイス可視性、および拡張設定管理を提供します。

グローバル コンフィギュレーション モードで **autoconf enable** コマンドを使用して Autoconf 機能を有効にすると、デバイス分類が有効になります。

デバイス検出はイベントトリガーとして機能し、適切な自動テンプレートをインターフェイスに適用します。

Autoconf 機能は、3 層階層に基づいています。

- ポリシーマップは、Autoconf 機能を適用するためのトリガータイプを識別します。
- パラメータマップは、エンドデバイスに基づいて、適用する必要がある適切なテンプレートを識別します。
- テンプレートには、適用する設定が含まれています。

Autoconf の組み込みテンプレートとトリガーは、これら 3 つの手順を自動的に実行します。

Autoconf 機能は、次の組み込みテンプレートを提供します。

- AP_INTERFACE_TEMPLATE
- DMP_INTERFACE_TEMPLATE
- IP_CAMERA_INTERFACE_TEMPLATE
- IP_PHONE_INTERFACE_TEMPLATE
- LAP_INTERFACE_TEMPLATE
- MSP_CAMERA_INTERFACE_TEMPLATE
- MSP_VC_INTERFACE_TEMPLATE
- PRINTER_INTERFACE_TEMPLATE
- ROUTER_INTERFACE_TEMPLATE
- SWITCH_INTERFACE_TEMPLATE
- TP_INTERFACE_TEMPLATE



(注) デフォルトでは、組み込みテンプレートは実行中の設定では表示されません。組み込みテンプレートは、編集した場合にのみ実行中のコンフィギュレーションに表示されます。

選択されるテンプレートは、インターフェイスに適用されるパラメータマップ情報に基づいています。この情報は、次の基準に基づく場合があります。

- エンドデバイスタイプ
- MAC アドレス
- OUI
- ユーザー ロール
- ユーザー名

Autoconf 機能は、次の設定の 1 つの組み込みパラメータマップ (BUILTIN_DEVICE_TO_TEMPLATE) を提供します。

```
Parameter-map name: BUILTIN_DEVICE_TO_TEMPLATE
Map: 10 map device-type regex "Cisco-IP-Phone"
Action(s):
  20 interface-template IP_PHONE_INTERFACE_TEMPLATE
Map: 20 map device-type regex "Cisco-IP-Camera"
Action(s):
  20 interface-template IP_CAMERA_INTERFACE_TEMPLATE
Map: 30 map device-type regex "Cisco-DMP"
Action(s):
  20 interface-template DMP_INTERFACE_TEMPLATE
Map: 40 map oui eq "00.0f.44"
Action(s):
  20 interface-template DMP_INTERFACE_TEMPLATE
Map: 50 map oui eq "00.23.ac"
```

```

Action(s):
  20 interface-template DMP_INTERFACE_TEMPLATE
Map: 60 map device-type regex "Cisco-AIR-AP"
Action(s):
  20 interface-template AP_INTERFACE_TEMPLATE
Map: 70 map device-type regex "Cisco-AIR-LAP"
Action(s):
  20 interface-template LAP_INTERFACE_TEMPLATE
Map: 80 map device-type regex "Cisco-TelePresence"
Action(s):
  20 interface-template TP_INTERFACE_TEMPLATE
Map: 90 map device-type regex "Surveillance-Camera"
Action(s):
  10 interface-template MSP_CAMERA_INTERFACE_TEMPLATE
Map: 100 map device-type regex "Video-Conference"
Action(s):
  10 interface-template MSP_VC_INTERFACE_TEMPLATE

```



- (注) 組み込みパラメータマップの設定を表示するには、**show parameter-map type subscriber attribute-to-service All** コマンドを使用します。

Autoconf機能は、次の設定の1つの組み込みポリシーマップ (BUILTIN_AUTOCONF_POLICY) を提供します。

```

BUILTIN_AUTOCONF_POLICY
event identity-update match-all
  10 class always do-until-failure
    10 map attribute-to-service table BUILTIN_DEVICE_TO_TEMPLATE

```



- (注) 組み込みポリシーマップの設定を表示するには、**show policy-map type control subscriber BUILTIN_AUTOCONF_POLICY** コマンドを使用します。

ポリシーマップ、パラメータマップ、およびテンプレートを手動で作成することもできます。

特定のユーザー情報に基づいてトリガーが作成されると、ローカル 802.1X Cisco Identity Services Engine (ISE) サーバーがトリガーを認証し、操作のセキュリティを確保します。

インターフェイステンプレートは、次のいずれかの方法を使用して (インターフェイス上で) 動的にアクティブ化できます。

- **RADIUS CoA** : 認可変更 (CoA) コマンドは 1 つ以上のアクセスセッションを対象としていますが、参照されるテンプレートは、参照されるセッションをホストするインターフェイスに適用される必要があります。
- **RADIUS Access-Accept** (クライアントの認証または認可用) : Access-Accept で返される参照先インターフェイステンプレートは、認可されたアクセスセッションをホストしているポートに適用する必要があります。
- **サービステンプレート** : ローカルに定義されたサービステンプレートまたは AAA サーバーから取得されたサービステンプレートでインターフェイステンプレートが参照されている場合、そのインターフェイステンプレートは、サービステンプレートが適用されるアクセスセッションをホストするインターフェイスに適用する必要があります (ローカルに定義

されたサービステンプレート内からインターフェイステンプレートを参照するための新しいコマンドを追加します)。

- 加入者制御ポリシーアクション：加入者制御ポリシーに基づくマッピングアクションは、フィルタのタイプに基づいてサービステンプレートやインターフェイステンプレート（パラメータマップで参照される）をアクティブ化し、以前のポリシーに関連付けられたテンプレートを削除します。
- デバイスからテンプレートへのパラメータマップ：サービスおよびインターフェイステンプレートへのフィルタタイプのマッピングを効率的かつ読み取り可能な方法で指定できる加入者パラメータマップ。

テンプレートを使用する利点

自動構成にテンプレートを使用すると、次の利点があります。

- テンプレートは、定義時に一度解析されます。これにより、テンプレートの動的な適用が非常に効率的になります。
- テンプレートは、エンドデバイスのタイプに基づいて、エンドデバイスに接続されているイーサネット インターフェイスに適用できます。
- サービステンプレートを使用すると、セッション指向機能をアクティブ化できます。一方、インターフェイステンプレートは、セッションをホストしているインターフェイスに設定を適用します。
- サービステンプレートはアクセスセッションに適用されるため、ポート上の単一のエンドポイントと交換されるトラフィックにのみ影響します。
- デバイスのスタートアップ設定と実行中の設定は、テンプレートの動的な適用によって変更されません。
- ポリシーの適用は、アクセスセッションのライフサイクルと同期されます。アクセスセッションのライフサイクルは、リンクアップ/リンクダウンだけを含む、利用可能なすべての手法を使用してフレームワークによって追跡されます。
- テンプレートは、1回の操作で更新できます。テンプレートの適用されたすべてのインスタンスが更新されます。
- テンプレートの構成コマンドは、実行中の設定には表示されません。
- 以前の設定や後続の設定に影響を与えることなく、テンプレートを削除できます。
- テンプレートアプリケーションが認識され、同期が可能になり、障害が発生した場合は修復アクションが実行されます。
- データ VLAN、Quality of Service (QoS) パラメータ、ストーム制御、および MAC ベースのポートセキュリティは、スイッチに接続されているエンドデバイスに基づいて自動的に設定されます。
- スイッチポートは、デバイスがポートから切断されたときに設定を削除することで完全にクリーンアップされます。

- インストールと設定のプロセスにおける人的エラーが減少します。

Autoconf の機能

Autoconf 機能は、デフォルトではグローバル コンフィギュレーション モードで無効になっています。Autoconf 機能をグローバル コンフィギュレーション モードで有効にすると、デフォルトでインターフェイスレベルで有効になります。組み込みテンプレート設定は、すべてのインターフェイスで検出されたエンドデバイスに基づいて適用されます。

Autoconf がグローバルレベルで有効になっている場合でも、インターフェイスレベルで手動で Autoconf を無効にするには、**access-session inherit disable autoconf** コマンドを使用します。

Autoconf をグローバルレベルで無効にすると、すべてのインターフェイスレベルの設定が無効になります。

グローバル	インターフェイスレベル	AutoConf ステータス
無効	無効	エンドデバイスが接続されている場合、自動設定は適用されません。
有効	デフォルトで有効	Autoconf がグローバルレベルで有効になっている場合は、デフォルトではインターフェイスレベルで有効です。組み込みテンプレート設定は、すべてのインターフェイスで検出されたエンドデバイスに基づいて適用されます。
有効	無効	グローバルレベルで有効です。インターフェイスレベルで無効です。Autoconf が無効になっているインターフェイスにエンドデバイスが接続されている場合、自動設定は適用されません。

Autoconf では、Autoconf スティッキ機能を設定することで、エンドデバイスへのリンクがダウンしている場合やエンドデバイスが切断されている場合でも、テンプレートを保持できます。**access-session interface-template sticky** コマンドを使用して、グローバルコンフィギュレーションモードで Autoconf スティッキ機能を設定します。Autoconf スティッキ機能により、エンドデバイスを検出し、リンクフラップまたはデバイスが取り外されて接続し直されるたびにテンプレートを適用する必要がなくなります。

access-session interface-template sticky コマンドは、**access-session** コマンドを含む組み込みテンプレートをインターフェイスに適用するために必須です。サービスポリシーを使用してポートにインターフェイステンプレートを適用するには **access-session interface-template sticky** コマンドを設定します。

特定のインターフェイスで Autoconf 機能を無効にするには、インターフェイス コンフィギュレーションモードで **access-session inherit disable interface-template-sticky** コマンドを使用します。

Autoconf の設定方法

ここでは、Autoconf の設定方法について説明します。

エンドデバイスへの組み込みテンプレートの適用

次のタスクでは、Cisco IP 電話などのエンドデバイスに接続されているインターフェイスに組み込みテンプレートを適用する方法を示します。

始める前に

Cisco IP 電話などのエンドデバイスがスイッチポートに接続されていることを確認します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device(config)# configure terminal	グローバル コンフィギュレーションモードを開始します。
ステップ 3	autoconf enable 例： Device(config)# autoconf enable	Autoconf 機能を有効にします。
ステップ 4	end 例： Device(config)# end	グローバル コンフィギュレーションモードを終了し、特権 EXEC モードを開始します。
ステップ 5	(任意) show device classifier attached interface interface-type interface-number 例： Device# show device classifier attached interface Gi3/0/26	エンドデバイスが正しい属性を持つデバイス分類子によって分類されているかどうかを表示します。
ステップ 6	show template binding target interface-type interface-number 例： Device# show template binding target gi3/0/26	インターフェイス上のテンプレートを介して適用された設定を表示します。

エンドデバイスのデバイス分類の確認

インターフェイスのインターフェイス テンプレートの確認

インターフェイス コンフィギュレーションの確認

Autoconf 適用後のグローバル設定の確認

次の例は、IP 電話が正しい属性を持つデバイス分類子によって分類されていることを示しています。

```
Device# show device classifier attached interface GigabitEthernet 3/0/26
```

Summary:

MAC_Address	Port_Id	Profile Name	Device Name
0026.0bd9.7bbb	Gi3/0/26	Cisco-IP-Phone-7962	Cisco IP Phone 7962

次の例は、組み込みインターフェイス テンプレートがインターフェイスに適用されることを示しています。

```
Device# show template binding target GigabitEthernet 3/0/26
```

```
Interface Templates
=====
Interface: Gi4/0/11
Method          Source          Template-Name
-----
dynamic         Built-in        IP_PHONE_INTERFACE_TEMPLATE
```

次の例は、インターフェイス テンプレートが GigabitEthernet インターフェイス 3/0/26 に接続された IP 電話に適用された後にインターフェイス設定を確認する方法を示しています。

```
Device# show running-config interface GigabitEthernet 3/0/26
Building configuration...
```

```
Current configuration : 624 bytes
!
interface GigabitEthernet3/0/26
!
End
```

```
Device# show derived-config interface GigabitEthernet 3/0/26
```

```
Building configuration...

Derived configuration : 649 bytes
!
interface GigabitEthernet3/0/26
 switchport mode access
 switchport block unicast
 switchport port-security maximum 3
 switchport port-security maximum 2 vlan access
 switchport port-security violation restrict
 switchport port-security aging time 2
 switchport port-security aging type inactivity
 switchport port-security
```

```

load-interval 30
storm-control broadcast level pps 1k
storm-control multicast level pps 2k
storm-control action trap
spanning-tree portfast
spanning-tree bpduguard enable
service-policy input AutoConf-4.0-CiscoPhone-Input-Policy
service-policy output AutoConf-4.0-Output-Policy
ip dhcp snooping limit rate 15
end

Device# show running config
class-map match-any AutoConf-4.0-Scavenger-Queue
  match dscp cs1
  match cos 1
  match access-group name AutoConf-4.0-ACL-Scavenger
class-map match-any AutoConf-4.0-VoIP
  match dscp ef
  match cos 5
class-map match-any AutoConf-4.0-Control-Mgmt-Queue
  match cos 3
  match dscp cs7
  match dscp cs6
  match dscp cs3
  match dscp cs2
  match access-group name AutoConf-4.0-ACL-Signaling
class-map match-any AutoConf-4.0-Multimedia-Conf
  match dscp af41
  match dscp af42
  match dscp af43
class-map match-all AutoConf-4.0-Broadcast-Vid
  match dscp cs5
class-map match-any AutoConf-4.0-Bulk-Data
  match dscp af11
  match dscp af12
  match dscp af13
class-map match-all AutoConf-4.0-Realtime-Interact
  match dscp cs4
class-map match-any AutoConf-4.0-VoIP-Signal
  match dscp cs3
  match cos 3
class-map match-any AutoConf-4.0-Trans-Data-Queue
  match cos 2
  match dscp af21
  match dscp af22
  match dscp af23
  match access-group name AutoConf-4.0-ACL-Transactional-Data
class-map match-any AutoConf-4.0-VoIP-Data
  match dscp ef
  match cos 5
class-map match-any AutoConf-4.0-Multimedia-Stream
  match dscp af31
  match dscp af32
  match dscp af33
class-map match-all AutoConf-4.0-Internetwork-Ctrl
  match dscp cs6
class-map match-all AutoConf-4.0-VoIP-Signal-Cos
  match cos 3
class-map match-any AutoConf-4.0-Multimedia-Stream-Queue
  match dscp af31
  match dscp af32
  match dscp af33
class-map match-all AutoConf-4.0-Network-Mgmt
  match dscp cs2
class-map match-all AutoConf-4.0-VoIP-Data-Cos

```

```
match cos 5
class-map match-any AutoConf-4.0-Priority-Queue
match cos 5
match dscp ef
match dscp cs5
match dscp cs4
class-map match-any AutoConf-4.0-Bulk-Data-Queue
match cos 1
match dscp af11
match dscp af12
match dscp af13
match access-group name AutoConf-4.0-ACL-Bulk-Data
class-map match-any AutoConf-4.0-Transaction-Data
match dscp af21
match dscp af22
match dscp af23
class-map match-any AutoConf-4.0-Multimedia-Conf-Queue
match cos 4
match dscp af41
match dscp af42
match dscp af43
match access-group name AutoConf-4.0-ACL-Multimedia-Conf
class-map match-all AutoConf-4.0-Network-Ctrl
match dscp cs7
class-map match-all AutoConf-4.0-Scavenger
match dscp cs1
class-map match-any AutoConf-4.0-Signaling
match dscp cs3
match cos 3
!
!
policy-map AutoConf-4.0-Cisco-Phone-Input-Policy
class AutoConf-4.0-VoIP-Data-Cos
set dscp ef
police cir 128000 bc 8000
exceed-action set-dscp-transmit cs1
exceed-action set-cos-transmit 1
class AutoConf-4.0-VoIP-Signal-Cos
set dscp cs3
police cir 32000 bc 8000
exceed-action set-dscp-transmit cs1
exceed-action set-cos-transmit 1
class class-default
set dscp default
set cos 0
policy-map AutoConf-4.0-Output-Policy
class AutoConf-4.0-Scavenger-Queue
bandwidth remaining percent 1
class AutoConf-4.0-Priority-Queue
priority
police cir percent 30 bc 33 ms
class AutoConf-4.0-Control-Mgmt-Queue
bandwidth remaining percent 10
class AutoConf-4.0-Multimedia-Conf-Queue
bandwidth remaining percent 10
class AutoConf-4.0-Multimedia-Stream-Queue
bandwidth remaining percent 10
class AutoConf-4.0-Trans-Data-Queue
bandwidth remaining percent 10
db1
class AutoConf-4.0-Bulk-Data-Queue
bandwidth remaining percent 4
db1
class class-default
```

```

bandwidth remaining percent 25
    dbl
policy-map AutoConf-DMP
  class class-default
    set dscp cs2
policy-map AutoConf-IPVSC
  class class-default
    set cos dscp table AutoConf-DscpToCos
policy-map AutoConf-4.0-Input-Policy
  class AutoConf-4.0-VoIP
  class AutoConf-4.0-Broadcast-Vid
  class AutoConf-4.0-Realtime-Interact
  class AutoConf-4.0-Network-Ctrl
  class AutoConf-4.0-Internetwork-Ctrl
  class AutoConf-4.0-Signaling
  class AutoConf-4.0-Network-Mgmt
  class AutoConf-4.0-Multimedia-Conf
  class AutoConf-4.0-Multimedia-Stream
  class AutoConf-4.0-Transaction-Data
  class AutoConf-4.0-Bulk-Data
  class AutoConf-4.0-Scavenger

```

エンドデバイスへの変更された組み込みテンプレートの適用

次のタスクは、複数のワイヤレスアクセスポイントと IP カメラがスイッチに接続されている場合に、組み込みテンプレートを変更する方法を示しています。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none">パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device(config)# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	template <i>template-name</i> 例： Device(config)# template AP_INTERFACE_TEMPLATE	組み込みテンプレートのテンプレート コンフィギュレーション モードを開始します。
ステップ 4	switchport access vlan <i>vlan-id</i> 例： Device(config-template)# switchport access vlan 20	インターフェイスがアクセス モードのときに VLAN を設定します。

	コマンドまたはアクション	目的
ステップ 5	description <i>description</i> 例： Device(config-template)# description modifiedAP	組み込みテンプレートの説明を変更します。
ステップ 6	exit 例： Device(config-template)# exit	テンプレート コンフィギュレーション モードを終了し、グローバル コンフィギュレーション モードを開始します。
ステップ 7	autoconf enable 例： Device(config)# autoconf enable	Autoconf 機能を有効にします。
ステップ 8	end 例： Device(config)# end	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードを開始します。
ステップ 9	show template interface binding all 例： Device# show template interface binding all	テンプレートがインターフェイスに適用されているかどうかを表示します。

エンドデバイスのデバイス分類の確認

インターフェイスのインターフェイス テンプレートの確認

次の例は、IP カメラとアクセスポイントが正しい属性を持つデバイス分類子によって分類されていることを示しています。

```
Device# show device classifier attached detail
```

```
DC default profile file version supported = 1
```

```
Detail:
```

MAC_Address	Port_Id	Cert	Parent	Proto	ProfileType	Profile Name
001d.a1ef.23a8	Gi1/0/7	30	3	C	M	Cisco-AIR-AP-1130
cisco AIR-AP1131AG-A-K9						
001e.7a26.eb05	Gi1/0/30	70	2	C	M	Cisco-IP-Camera
Cisco IP Camera						

次の例は、組み込みインターフェイス テンプレートがインターフェイスに適用されることを示しています。

```
Device# show template interface binding all
```

Template-Name	Source	Method	Interface
---------------	--------	--------	-----------

```

-----
IP_CAMERA_INTERFACE_TEMPLATE      Built-in      dynamic      Gi1/0/30
AP_INTERFACE_TEMPLATE             Modified-Built-in  dynamic      Gi1/0/7

```

ASP から Autoconf への移行

始める前に

show running-config | include macro auto global コマンドを使用して、AutoSmart ポート (ASP) マクロが実行されていることを確認します。

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 3	no macro auto global processing 例： Device(config)# no macro auto global processing	グローバルレベルで ASP を無効にします。
ステップ 4	exit 例： Device(config)# exit	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 5	clear macro auto configuration all 例： Device# clear macro auto configuration all	すべてのインターフェイスのマクロ設定をクリアします。
ステップ 6	configure terminal 例： Device# configure terminal	グローバル コンフィギュレーション モードを開始します。
ステップ 7	autoconf enable 例： Device(config)# autoconf enable	Autoconf 機能を有効にします。

	コマンドまたはアクション	目的
ステップ 8	end 例 : Device(config)# end	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

Autoconf の設定例

次のセクションに Autoconf の設定例を示します。

例：エンドデバイスへの組み込みテンプレートの適用

次に、インターフェイスに接続されたエンドデバイスに組み込みテンプレートを適用する例を示します。

```
Device> enable
Device(config)# configure terminal
Device(config)# autoconf enable
Device(config)# end
Device# show device classifier attached interface Gi3/0/26
Device# show template binding target GigabitEthernet 3/0/26
```

例：エンドデバイスへの変更された組み込みテンプレートの適用

次の例は、組み込みテンプレートを変更して構成を確認する方法を示しています。

```
Device> enable
Device(config)# configure terminal
Device(config)# template AP_INTERFACE_TEMPLATE
Device(config-template)# switchport access vlan 20
Device(config-template)# description modifiedAP
Device(config-template)# exit
Device(config)# autoconf enable
Device(config)# end
Device# show template interface binding all
```

例：ASP マクロから Autoconf への移行

次に、ASP から Autoconf に移行する例を示します。

```
Device> enable
Device# configure terminal
Device(config)# no macro auto global processing
Device(config)# exit
Device# clear macro auto configuration all
Device# configure terminal
Device(config)# autoconf enable
Device(config)# end
```

Autoconf のその他の参考資料

関連資料

関連項目	マニュアル タイトル
Cisco identity-based networking services コマンド	『Cisco IOS Identity-Based Networking Services Command Reference』
インターフェイス テンプレート	『Identity-Based Networking Services Configuration Guide』の「Interface Templates」の章。

標準および RFC

標準/RFC	タイトル
IEEE 802.1X	「Port Based Network Access Control」

シスコのテクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	http://www.cisco.com/cisco/web/support/index.html

Autoconf の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.6.1	[Autoconf]	この機能が導入されました。 Autoconf 機能では、エンドデバイスとインターフェイス間のハードバインディングが可能です。 autoconf enable 、 map attribute-to-service (autoconf) 、 map device-type (service-template) 、 parameter-map type subscriber (service-template) 、 show parameter-map type subscriber attribute-to-service all 、 show template interface の各コマンドが追加または変更されました。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator にアクセスするには、<https://cfngng.cisco.com> に進みます。

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。