



フレキシブルコマンドラインインターフェイスの設定

このモジュールでは、フレキシブル コマンドライン インターフェイス (CLI) の設定グループを構成および使用する方法について説明します。

- [フレキシブル CLI 設定グループ \(1 ページ\)](#)
- [フレキシブル設定の制約事項 \(1 ページ\)](#)
- [設定グループの構成 \(3 ページ\)](#)
- [設定グループの構成の確認 \(6 ページ\)](#)
- [設定グループの正規表現 \(7 ページ\)](#)

フレキシブル CLI 設定グループ

フレキシブル コマンドライン インターフェイス (CLI) の設定グループは、設定グループ内に一連の構成ステートメントを定義し、このグループをルータ設定ツリーの複数の階層レベルに適用することにより、反復構成を最小限に抑える機能を提供します。

フレキシブル CLI 設定グループは、階層内でグループが適用されている場所に基づいて、設定ツリーの複数のサブモードで一致が確認される正規表現を使用します。設定サブモードで一致が見つかった場合、そのグループに定義されている対応する構成が、一致するサブモード内で継承されます。

フレキシブル CLI 設定グループには、自動継承機能もあります。自動継承とは、CLI 設定グループに対して行われた変更が、その階層レベルで適用グループを持つ一致するサブモードの構成に自動的に適用されることを意味します。これにより、フレキシブル CLI 設定グループを適用した場所に応じて、構成の変更または追加を一度行い、複数の場所に自動的に適用することができます。

フレキシブル設定の制約事項

フレキシブル設定グループを使用する場合は、次の制約事項に注意してください。

- フレキシブル CLI 設定グループは管理構成ではサポートされず、対応する適用グループは管理構成ではサポートされません。
- 設定グループ内の事前設定されたインターフェイスの使用はサポートされていません。
- 設定グループをサポートしているイメージからサポートしていないイメージにダウンロードすることはサポートされていません。
- アクセスリスト、QoS およびルート ポリシー設定は、設定グループの使用をサポートしていません。次のような構成は有効ではありません。

```
group g-not-supported
  ipv4 access-list ...
  !
  ipv6 access-list ...
  !
  ethernet-service access-list ...
  !
  class-map ...
  !
  policy-map ...
  !
  route-policy ...
  !
end-group
```

ただし、次の例に示すような構成を参照できます。

```
group g-reference-ok
  router bgp 6500
  neighbor 7::7
    remote-as 65000
    bfd fast-detect
    update-source Loopback300
    graceful-restart disable
    address-family ipv6 unicast
      route-policy test1 in
      route-policy test2 out
      soft-reconfiguration inbound always
    !
  !
  interface Bundle-Ether1005
    bandwidth 10000000
    mtu 9188
    service-policy output input_1
    load-interval 30
  !
end-group
```

- 一部の正規表現はグループ内ではサポートされていません。たとえば、‘?’、‘|’ および ‘\$’ はグループ内ではサポートされていません。また、/d や /w などの文字はサポートされていません。
 - 正規表現内で複数の一致式を表す選択演算子「|」はサポートされていません。たとえば、次の式はサポートされていません。

Gig.*|Gig.*\.* : ギガビットイーサネット インターフェイスまたはギガビットイーサネット サブインターフェイスでの照合時。

Gig.*0/0/0/[1-5]|Gig.*0/0/0/[10-20] : Gig.*0/0/0/[1-5] または Gig.*0/0/0/[10-20] での照合時。

'TenGigE.*|HundredGigE.* : TenGigE.* または HundredGigE.* での照合時。

- **location** キーワードのノード識別子を必要とするコマンドはサポートされていません。たとえば、次の構成はサポートされません。

```
lpts pifib hardware police location 0/RP0/CPU0
```

- 同じ構成に対する設定グループ内の重複する正規表現はサポートされていません。次に例を示します。

```
group G-INTERFACE
interface 'gig.*a.*'
    mtu 1500
!
interface 'gig.*e.* '
    mtu 2000
!
end-group

interface gigabitethernet0/0/0/* ---- where * is 0 to 31
    apply-group G-INTERFACE
```

interface GigabitEthernet0/0/0/* 設定が mtu 1500 または mtu 2000 を継承するかどうかを判別できないため、この設定は許可されません。設定グループの両方の式は、GigabitEthernet0/0/0/* と一致します。

- 1 つの apply-group コマンドでは、最大 8 つの設定グループが許可されます。

設定グループの構成

設定グループには、ルータ設定ツリーの複数の階層レベルで使用できる一連の構成ステートメントが含まれています。設定グループで正規表現を使用すると、複数のインスタンスに適用できる汎用コマンドを作成できます。

設定グループを作成して使用するには、このタスクを使用します。



(注) フレキシブル CLI 設定は、XML インターフェイスでは利用できません。

手順

ステップ 1 configure

ステップ 2 group group-name

例：

```
RP/0/RP0/cpu 0: router(config)# group g-interf
```

設定グループの名前を指定し、グループを定義するためのグループ コンフィギュレーション モードを開始します。`group-name` 引数は最大 32 文字で、特殊文字は使用できません。

ステップ 3 グローバル コンフィギュレーション モードから開始して、設定コマンドを入力します。インターフェイス名やその他の変数インスタンスには正規表現を使用します。

例：

```
RP/0/RP0/cpu 0: router(config)# group g-interf
RP/0/RP0/cpu 0: router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/cpu 0: router(config-GRP-if)# mtu 1500
```

この設定グループに含める構成ステートメントを指定します。

正規表現の使用に関する詳細については、[正規表現を使用した設定グループの継承：例（16 ページ）](#) を参照してください。この例は、すべてのギガビットイーサネットインターフェイスに適用されます。

ステップ 4 end-group

例：

```
RP/0/RP0/cpu 0: router(config-GRP-if)# end-group
```

設定グループの設定を完了し、グローバル コンフィギュレーション モードを終了します。

ステップ 5 apply-group

例：

```
RP/0/RP0/cpu 0: router(config)# interface GigabitEthernet0/2/0/0
RP/0/RP0/cpu 0: router(config-if)# apply-group g-interf
```

グループが適用されている場所に適用可能なルータ構成に設定グループの構成を追加します。グループは複数の場所に適用することができ、その影響は場所とコンテキストによって異なります。

グループ `g-interf` からの MTU 値は、インターフェイス `TenGigE0/11/0/0` に適用されます。このグループがグローバル コンフィギュレーション モードで適用される場合、この MTU 値は、MTU 値が設定されていないすべてのギガビットイーサネットインターフェイスによって継承されます。

シンプルな設定グループ：例

次に、設定グループを使用してグローバル設定をシステムに追加する例を示します。

```
RP/0/RP0/cpu 0: router(config)# group g-logging
RP/0/RP0/cpu 0: router(config-GRP)# logging trap notifications
RP/0/RP0/cpu 0: router(config-GRP)# logging console debugging
RP/0/RP0/cpu 0: router(config-GRP)# logging monitor debugging
RP/0/RP0/cpu 0: router(config-GRP)# logging buffered 10000000
RP/0/RP0/cpu 0: router(config-GRP)# end-group

RP/0/RP0/cpu 0: router(config)# apply-group g-logging
```

この設定がコミットされると、g-logging 設定グループに含まれるすべてのコマンドがコミットされます。

異なる場所に適用された設定グループ：例

設定グループは異なる場所に適用でき、その影響は適用されるコンテキストによって異なります。次の設定グループを考えてみましょう。

```
RP/0/RP0/cpu 0: router(config)# group g-interfaces
RP/0/RP0/cpu 0: router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/cpu 0: router(config-GRP-if)# mtu 1500
RP/0/RP0/cpu 0: router(config-GRP-if)# exit
RP/0/RP0/cpu 0: router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/cpu 0: router(config-GRP-if)# mtu 1000
RP/0/RP0/cpu 0: router(config-GRP-if)# exit
RP/0/RP0/cpu 0: router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/cpu 0: router(config-GRP-if)# mtu 2000
RP/0/RP0/cpu 0: router(config-GRP-if)# end-group
```

このグループはギガビットイーサネットインターフェイスに適用でき、各インスタンスでは、適用可能な MTU が適用されます。たとえば、次の例では、ギガビットイーサネットインターフェイスの MTU が 1000 に設定されています。

```
RP/0/RP0/cpu 0: router(config)# interface TenGigE0/11/0/0
RP/0/RP0/cpu 0: router(config-if)# apply-group g-interfaces
RP/0/RP0/cpu 0: router(config-if)# ipv4 address 2.2.2.2 255.255.255.0
```

次の例では、ギガビットイーサネットインターフェイスの MTU が 1500 に設定されています。

```
RP/0/RP0/cpu 0: router(config)# interface TenGigE0/11/0/0
RP/0/RP0/cpu 0: router(config-if)# apply-group g-interfaces
RP/0/RP0/cpu 0: router(config-if)# ipv4 address 3.3.3.3 255.255.255.0
```

どちらの場合も同じ設定グループが使用されていますが、適用可能な構成ステートメントのみが使用されています。

設定グループの構成の確認

このタスクを使用して、設定グループを使用したルータの設定を確認します。

手順

	コマンドまたはアクション	目的
ステップ 1	show running-config group [group-name] 例 : <pre>RP/0/RP0/cpu 0: router# show running-config group group g-int-ge interface 'GigabitEthernet.*' mtu 1000 negotiation auto ! end-group</pre>	特定のまたはすべての構成済み設定グループの内容を表示します。
ステップ 2	show running-config 例 : <pre>RP/0/RP0/cpu 0: router# show running-config group G-INTERFACE-MTU interface 'GigabitEthernet.*' mtu 1500 ! end-group interface interface TenGigE0/11/0/0 apply-group G-INTERFACE-MTU ! interface interface TenGigE0/11/0/1 apply-group G-INTERFACE-MTU mtu 2000 !</pre>	実行コンフィギュレーションを表示します。適用されているグループが表示されます。これらの設定グループが実際の構成に影響を与えるかどうかに関しては示されません。この例では、グループ G-INTERFACE-MTU は interface TenGigE0/11/0/1 に適用されますが、設定された MTU 値は 1500 ではなく 2000 です。これは、コマンド mtu 2000 がインターフェイス上で直接設定されている場合に発生します。実際の構成と設定グループの構成が同じであれば、実際の構成が優先されます。
ステップ 3	show running-config inheritance 例 : <pre>RP/0/RP0/cpu 0: router# show running-config inheritance . . group G-INTERFACE-MTU interface 'GigabitEthernet.*' mtu 1500 ! end-group</pre>	設定グループが適用されている継承された構成を表示します。

	コマンドまたはアクション	目的
	<pre> . . interface interface TenGigE0/11/0/0 ## Inherited from group G-INTERFACE-MTU mtu 1500 ! interface interface TenGigE0/11/0/1 mtu 2000 ! . . </pre>	
ステップ 4	<p>show running-config interface x/y/z inheritance detail</p> <p>例 :</p> <pre> RP/0/RP0/cpu 0: router# show running-config interface interface TenGigE0/11/0/0 inheritance detail interface interface TenGigE0/11/0/0 ## Inherited from group G-INTERFACE-MTU mtu 1500 </pre>	特定のコンフィギュレーション コマンドに対する継承された構成を表示します。

設定グループの正規表現

正規表現は、設定グループを幅広く適用可能にするために設定グループで使用されます。Portable Operating System Interface for UNIX (POSIX) 1003.2 正規表現は、構成ステートメントの名前でサポートされています。正規表現を区切るには、単一引用符を使用する必要があります。



(注) すべての POSIX 正規表現がサポートされているわけではありません。

インターフェイス識別子の正規表現

設定グループは、正確なインターフェイス識別子を受け入れません。設定グループに適用可能なインターフェイスのグループを識別するには、正規表現を使用する必要があります。正規表現 `.*` は使用できません。インターフェイス識別子の正規表現は、一義的な単語で始まり、その後正規表現を続けなければなりません。たとえば、ギガビットイーサネットインターフェイスを設定するには、正規表現 `'GigabitEthernet.*'` を使用します。

ルータ設定で使用可能なインターフェイス タイプのリストを表示するには、設定グループプロンプトで **interface ?** と入力します。

```
RP/0/RP0/cpu 0: router(config-GRP)# interface ?
```

```

ATM          'RegExp': ATM Network Interface(s)
BVI          'RegExp': Bridge-Group Virtual Interface
Bundle-Ether 'RegExp': Aggregated Ethernet interface(s)
GigabitEthernet 'RegExp': GigabitEthernet/IEEE 802.3 interface(s)
IMA         'RegExp': ATM Network Interface(s)
Loopback    'RegExp': Loopback interface(s)
MgmtEth     'RegExp': Ethernet/IEEE 802.3 interface(s)
Multilink   'RegExp': Multilink network interface(s)
Null        'RegExp': Null interface
PW-Ether    'RegExp': PWHE Ethernet Interface
PW-IW       'RegExp': PWHE VC11 IP Interworking Interface
Serial      'RegExp': Serial network interface(s)
tunnel-ip   'RegExp': GRE/IPinIP Tunnel Interface(s)
tunnel-mte  'RegExp': MPLS Traffic Engineering P2MP Tunnel interface(s)
tunnel-te   'RegExp': MPLS Traffic Engineering Tunnel interface(s)
tunnel-tp   'RegExp': MPLS Transport Protocol Tunnel interface

```



- (注) インターフェイスのタイプを一意なものにするのに十分な文字数のみを入力するよう要求されますが、フレーズ全体を入力することをお勧めします。正規表現で使用されるすべてのインターフェイスタイプでは、大文字と小文字が区別されます。

サブインターフェイスを指定するには、式の前に\という文字を付けます（バックスラッシュピリオド）。たとえば、すべてのギガビットイーサネットサブインターフェイスを設定するには、`interface 'GigabitEthernet.*\..*'` を使用します。

次の例に示すように、レイヤ2トランスポートインターフェイスまたはポイントツーポイントインターフェイスを指定できます。

```

group g-l2t
  interface 'Gi.*\..*' l2transport
  .
  .
end-group
group g-ptp
  interface 'Gi.*\..*' point-to-point
  .
  .
end-group

```

OSPF 設定の正規表現

正確なルータ プロセス名と OSPF 領域は使用できません。プロセス名または OSPF 領域のグループを指定するには、正規表現を使用する必要があります。OSPF 領域がスカラー値または IP アドレスのいずれかになるように指定するには、次の例のように正規表現 `'.*'` を使用します。

```

group g-ospf
router ospf '.*'
area '.*'
mtu-ignore enable
!
!

```

```
end-group
```

OSPF 領域を IP アドレスにする必要があることを指定するには、次の例のように式 '\!' を使用します。

```
group g-ospf-ipaddress
router ospf '\.*\...\*\.\.\.*\.\.\.*'
area '\.*'
passive enable
!
!
end-group
```

OSPF 領域をスカラー値にする必要があることを指定するには、次の例のように式 '1.*' を使用します。

```
group g-ospf-match-number
router ospf '1.*'
area '1.*'
passive enable
!
!
end-group
```

BGP AS の正規表現

正確な BGP AS 値は、設定グループでは使用できません。形式 X.Y など AS プレーン形式または AS ドット形式を指定するには正規表現を使用します。AS プレーン形式のインスタンスを照合するには、単純な正規表現を使用します。AS ドット形式のインスタンスを照合するには、次の例に示すように、ドットで区切られた 2 つの正規表現を使用します。

```
group g-bgp
router bgp '*'.*'
address-family ipv4 unicast
!
!
end-group
```

ANCP の正規表現

正確なアクセスノード制御プロトコル (ANCP) 送信者名識別子は設定グループで使用できません。送信者名の引数は IP アドレスまたは MAC アドレスのどちらにでもできるので、正規表現ではどちらが使用されているかを指定する必要があります。IP アドレスは '\.*\...*\.\.\.*\.\.\.*' と指定します。MAC アドレスは '\.*\...*\.\.\.*' と指定します。

ユニフォームタイプへの解決

正規表現は、ユニフォームタイプに解決されなければなりません。次は、不正な正規表現の例です。

```
group g-invalid
```

```

interface \.*'
  bundle port-priority 10
!
interface \.*Ethernet.*'
  bundle port-priority 10
!
end-group

```

この例では、**bundle** コマンドはインターフェイス タイプ **GigabitEthernet** ではサポートされていますが、インターフェイス タイプ **FastEthernet** ではサポートされていません。正規表現 **'.*'** および **'.*Ethernet.*'** は、**GigabitEthernet** タイプと **FastEthernet** タイプの両方に一致します。**bundle** コマンドはこれらの両方のインターフェイス タイプには適用できず、ユニフォーム タイプに解決しないので、システムはこの設定を許可しません。



(注) システムが正規表現からどの構成をすべきかを判別できない場合、その式は有効であると見なされません。



(注) インターフェイス識別子の参照時に正規表現 **'.*'** は使用できません。インターフェイス識別子の正規表現は、一義的な単語で始まり、その後に正規表現を続けなければなりません。詳細については、この項の「インターフェイス識別子の正規表現」を参照してください。

重複する正規表現

正規表現は、設定グループ内の構成ステートメントの名前で使用されます。これにより、一致する名前に適用された場合に設定による継承が可能になります。正規表現を区切るには、単一引用符が使用されます。同じ構成に対し設定グループ内の重複する正規表現は許可されません。

以下に示す例は、複数の設定グループを作成して適用するプロセスを示しています。

```

RP/0//CPU0:router(config)#group FB_flexi_snmp
RP/0//CPU0:router(config-GRP)# snmp-server vrf \.*'
RP/0//CPU0:router(config-GRP-snmp-vrf)# host 1.1.1.1 traps version 2c group_1
RP/0//CPU0:router(config-GRP-snmp-vrf)# host 1.1.1.1 informs version 2c group_1
RP/0//CPU0:router(config-GRP-snmp-vrf)# context group_1

RP/0//CPU0:router(config-GRP-snmp-vrf)#
RP/0//CPU0:router(config-GRP-snmp-vrf)#commit

RP/0//CPU0:router(config-GRP-snmp-vrf)#root
RP/0//CPU0:router(config)#
RP/0//CPU0:router(config)#snmp-server vrf vrf1
RP/0//CPU0:router(config-snmp-vrf)#snmp-server vrf vrf10
RP/0//CPU0:router(config-snmp-vrf)#!
RP/0//CPU0:router(config-snmp-vrf)#snmp-server vrf vrf100
RP/0//CPU0:router(config-snmp-vrf)#
RP/0//CPU0:router(config-snmp-vrf)#commit

RP/0//CPU0:router(config-snmp-vrf)#root

```

```

RP/0//CPU0:router(config)#
RP/0//CPU0:router(config)#apply-group FB_flexi_snmp
RP/0//CPU0:router(config)#do sh running-config group
group FB_flexi_snmp
  snmp-server vrf '.*'
    host 1.1.1.1 traps version 2c group_1
    host 1.1.1.1 informs version 2c group_1
    context group_1
  !
end-group
apply-group FB_flexi_snmp
snmp-server vrf vrf1
!
snmp-server vrf vrf10
!
snmp-server vrf vrf100
!
RP/0//CPU0:ios#show running-config inheritance detail

```

```

group FB_flexi_snmp
  snmp-server vrf '.*'
    host 1.1.1.1 traps version 2c group_1
    host 1.1.1.1 informs version 2c group_1
    context group_1
  !
end-group
snmp-server vrf vrf1
## Inherited from group FB_flexi_snmp
host 1.1.1.1 traps version 2c group_1
## Inherited from group FB_flexi_snmp
host 1.1.1.1 informs version 2c group_1
## Inherited from group FB_flexi_snmp
context group_1
!
snmp-server vrf vrf10
## Inherited from group FB_flexi_snmp
host 1.1.1.1 traps version 2c group_1
## Inherited from group FB_flexi_snmp
host 1.1.1.1 informs version 2c group_1
## Inherited from group FB_flexi_snmp
context group_1
!
snmp-server vrf vrf100
## Inherited from group FB_flexi_snmp
host 1.1.1.1 traps version 2c group_1
## Inherited from group FB_flexi_snmp
host 1.1.1.1 informs version 2c group_1
## Inherited from group FB_flexi_snmp
context group_1

```

次の例は、正規表現を示しています。この例では、`snmp-server vrf '.*'` および `snmp-server vrf '[\w]+'` は 2 つの異なる正規表現です。

```

group FB_flexi_snmp
snmp-server vrf '.*'
host 1.1.1.1 traps version 2c group_1
host 1.1.1.1 informs version 2c group_1
context group_1
!

```

```
snmp-server vrf '[\w]+'
host 2.2.2.2 traps version 2c group_2
host 2.2.2.2 informs version 2c group_2
context group_2
!
end-group
```

この個々の正規表現は、次に示すように3つの式 `snmp-server vrf vrf1`、`snmp-server vrf vrf10` および `snmp-server vrf vrf100` すべてに結合されます。

```
apply-group FB_flexi_snmp
snmp-server vrf vrf1
!
snmp-server vrf vrf10
!
snmp-server vrf vrf100
!
```

設定グループには、正規表現の重複のインスタンスが存在することがあります。このような場合、適用されたときに最も優先順位の高い正規表現がアクティブ化され、継承されます。その正規表現は、最も優先順位の高い辞書式順で最初に来ます。

次の例は、重複する正規表現を使用する方法と、優先順位の高い式がどのように適用されるかを示しています。

```
group FB_flexi_snmp
snmp-server vrf '.*'

host 1.1.1.1 traps version 2c group_1

host 1.1.1.1 informs version 2c group_1

context group_1

!

snmp-server vrf '[\w]+'

host 2.2.2.2 traps version 2c group_2
host 2.2.2.2 informs version 2c group_2
context group_2
!
end-group
```

次に示す式は最も優先順位が高いです。

```
group FB_flexi_snmp
snmp-server vrf '.*'

host 1.1.1.1 traps version 2c group_1

host 1.1.1.1 informs version 2c group_1
```

```
context group_1
```

上記の例は、2つの異なる正規表現 `snmp-server vrf '.*'` と `snmp-server vrf '[\w]+'` を示しています。

次の式は、これら2つの式がどのようにマージされるかを示しています。

```
apply-group FB_flexi_snmp

snmp-server vrf vrf1
!
snmp-server vrf vrf10
!
snmp-server vrf vrf100
!
```

優先順位の低い正規表現の変更は、継承に影響しません。

優先順位が低い（上位ではない）既存の正規表現に加えられた変更は、継承には影響しません。

```
snmp-server vrf '[\w]+'

host 2.2.2.2 traps version 2c group_2
host 2.2.2.2 informs version 2c group_2
context group_2
```

次に示すように、優先順位の高い式が継承されます。

```
group FB_flexi_snmp

snmp-server vrf '.*'

host 1.1.1.1 traps version 2c group_1

host 1.1.1.1 informs version 2c group_1

context group_1
```

グループの優先順位継承の適用

優先順位によって継承が制御されます。



- (注) Cisco IOS XR リリース 6.3.1 以降では、コミット全体に必要なすべてのグループ定義があれば、フレキシブル CLI 設定グループ定義、**apply-group** および **exclude-group** コマンドを任意の順序で入力できます。

グループの優先順位の継承を適用すると、フレキシブルな設定グループがグループ間の共通の構成ステートメントを処理できるようになります。複数の設定グループに共通の構成ステートメントがある場合、継承の優先順位は、内部グループに存在する構成ステートメントが外部グループに存在する構成ステートメントよりも優先されます。タイブレーカーの場合、優先順位

は正規表現の辞書式順序に従って割り当てられます。ユーザが定義したコマンドの順序は受け入れられません。

たとえば、設定グループ ONE の構成ステートメントは、別のグループよりも優先されます。設定グループ SEVEN の構成ステートメントは、他のグループに存在しない場合にのみ使用されます。設定グループ内では、継承の優先順位は最長一致です。

```

apply-group SIX SEVEN
router ospf 0
apply-group FOUR FIVE
area 0
apply-group THREE
interface TenGigE0/11/0/0
apply-group ONE TWO

!
!
!
```

上記の例は、2つのシナリオを示しています。最も内側のグループ (**apply-group ONE TWO**) が最も優先順位が高くなります。ケース 1

最初のシナリオは、どのグループが優先順位を得るかを示しています。この例では、どのグループが異なる設定グループ (共通のものがない異なるグループ) 間で適用されるかを示しています。グループ 1 (**ONE TWO**) を適用すると、7つのグループすべてがインターフェイス `interfaceTenGigE0/11/0/0` に一致し、適用されます。

ケース 2

ここで、すべてが同じ (共通の) 構成を持つ場合、グループ 1 がアクティブになります。つまり、`apply-group ONE TWO` がアクティブになります。グループ 1 が削除されると、グループ 2 がアクティブになります。

正規表現を使用した設定例

正規表現を使用した設定グループ : 例

次の例では、正確なインターフェイスに正規表現を使用して、ISIS ルーティングパラメータでギガビットイーサネットインターフェイスを設定するための設定グループの定義を示しています。

```

RP/0/RP0/cpu 0: router(config)# group g-isis-gige
RP/0/RP0/cpu 0: router(config-GRP)# router isis '.*'
RP/0/RP0/cpu 0: router(config-GRP-isis)# interface 'GigabitEthernet.*'
RP/0/RP0/cpu 0: router(config-GRP-isis-if)# lsp-interval 20
RP/0/RP0/cpu 0: router(config-GRP-isis-if)# hello-interval 40
RP/0/RP0/cpu 0: router(config-GRP-isis-if)# address-family ipv4 unicast
RP/0/RP0/cpu 0: router(config-GRP-isis-if-af)# metric 10
RP/0/RP0/cpu 0: router(config-GRP-isis-if-af)# end-group
RP/0/RP0/cpu 0: router(config)#
```

この設定グループの使用について説明するために、ISIS ルーティング パラメータを使用してこれらのギガビットイーサネットインターフェイスを設定すると仮定します。

```
router isis green
interface TenGigE0/11/0/0
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
  metric 10
  !
!
interface TenGigE0/11/0/1
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
  metric 10
  !
!
interface TenGigE0/11/0/2
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
  metric 10
  !
!
interface TenGigE0/11/0/3
  lsp-interval 20
  hello-interval 40
  address-family ipv4 unicast
  metric 10
  !
!
!
```

設定グループを使用してこれらのインターフェイスを設定するには、3つの方法が考えられます。1つ目は、次に示すように、インターフェイス設定内でグループを適用することです。

```
router isis green
interface TenGigE0/11/0/0
  apply-group g-isis-gige
  !
!
interface TenGigE0/11/0/1
  apply-group g-isis-gige
  !
!
interface TenGigE0/11/0/2
  apply-group g-isis-gige
  !
!
interface TenGigE0/11/0/3
  apply-group g-isis-gige
  !
!
```

この状況では、設定グループを適用するインターフェイスのみが設定を継承します。

正規表現を使用した設定グループの継承：例

設定グループを使用してこれらのインターフェイスを設定する2つ目の方法は、次に示すように、**router isis** 設定内に設定グループを適用することです。

```
router isis green
  apply-group g-isis-gige
interface TenGigE0/11/0/0
!
interface TenGigE0/11/0/1
!
interface TenGigE0/11/0/2
!
interface TenGigE0/11/0/3
!
!
```

このようにすると、**ISIS** グリーン設定で設定する他のギガビットイーサネットインターフェイスもこれらの設定を継承します。

設定グループを使用してこれらのインターフェイスを設定する3つ目の方法は、次に示すように、グローバルレベルでグループを適用することです。

```
  apply-group g-isis-gige
router isis green
interface TenGigE0/11/0/0
!
interface TenGigE0/11/0/1
!
interface TenGigE0/11/0/2
!
interface TenGigE0/11/0/3
!
!
```

この例では、**ISIS** 用に設定されたすべてのギガビットイーサネットインターフェイスにグループの設定が適用されます。

正規表現を使用した設定グループの継承：例

設定グループより優先されるローカル設定

明示的な設定は、設定グループから適用された設定よりも優先されます。たとえば、次の設定がルータ上で実行されているとします。

```
router ospf 100
  packet-size 1000
!
```

次の設定グループを構成して適用し、設定にコミットします。

```
RP/0/RP0/cpu 0: router(config)# group g-ospf
RP/0/RP0/cpu 0: router(config-GRP)# router ospf '.*'
RP/0/RP0/cpu 0: router(config-GRP-ospf)# nsf cisco
```

```
RP/0/RP0/cpu 0: router(config-GRP-ospf)# packet-size 3000
RP/0/RP0/cpu 0: router(config-GRP-ospf)# end-group

RP/0/RP0/cpu 0: router(config)# apply-group g-ospf
```

結果は事実上次の構成となります。

```
router ospf 100
  packet-size 1000
  nsf cisco
```

明示的なローカル設定が優先されるため、packet-size 3000 は設定グループから継承されないことに注意してください。

互換性のある構成は継承される

設定グループの構成は、継承されるルータ上の構成と一致する必要があります。構成が一致しない場合は、継承されません。たとえば、次の設定がルータ上で実行されているとします。

```
router ospf 100
  auto-cost disable
!
```

次の設定を構成し、設定にコミットします。

```
RP/0/RP0/cpu 0: router(config)# group g-ospf
RP/0/RP0/cpu 0: router(config-GRP)# router ospf '.*'
RP/0/RP0/cpu 0: router(config-GRP-ospf)# area '.*'
RP/0/RP0/cpu 0: router(config-GRP-ospf-ar)# packet-size 2000
RP/0/RP0/cpu 0: router(config-GRP-ospf)# end-group

RP/0/RP0/cpu 0: router(config)# apply-group g-ospf

RP/0/RP0/cpu 0: router(config)# router ospf 200
RP/0/RP0/cpu 0: router(config-ospf)# area 1
```

結果は事実上次の構成となります。

```
router ospf 100
  auto-cost disable

router ospf 200
  area 1
  packet-size 2000
```

パケットサイズは、ospf 200 構成によって継承されますが、ospf 100 構成では領域が設定されていないため継承されません。

レイヤ2トランスポート設定グループ：例

次に、レイヤ2トランスポートサブインターフェイスを使用して設定グループを構成および適用する例を示します。

```
RP/0/RP0/cpu 0: router(config)# group g-l2trans-if
RP/0/RP0/cpu 0: router(config-GRP)# interface 'TenGigE.*\.*' l2transport
RP/0/RP0/cpu 0: router(config-GRP)# mtu 1514
RP/0/RP0/cpu 0: router(config-GRP)# end-group

RP/0/RP0/cpu 0: router(config)# interface TenGigE0/0/0/0.1 l2transport
RP/0/RP0/cpu 0: router(config-if)# apply-group g-l2trans-if
```

この設定がコミットされると、10ギガビットイーサネットインターフェイス0/11/0/0.1は1514 MTU 値を継承します。これは、10ギガビットイーサネットインターフェイスの **show running-config inheritance** コマンドで表示される出力です。

```
interface TenGigE0/11/0/0.1 l2transport
## Inherited from group g-l2trans-if
mtu 1514
!
```

設定グループの優先順位：例

同様の構成ステートメントが複数の設定グループに含まれている場合、内部の設定モードで適用されるグループが外部モードで適用されるグループよりも優先されます。次の例では、OSPFにさまざまなコスト値を設定している2つの設定グループを示しています。

```
RP/0/RP0/cpu 0: router(config)# group g-ospf2
RP/0/RP0/cpu 0: router(config-GRP)# router ospf '.*'
RP/0/RP0/cpu 0: router(config-GRP-ospf)# area '.*'
RP/0/RP0/cpu 0: router(config-GRP-ospf-ar)# cost 2
RP/0/RP0/cpu 0: router(config-GRP-ospf-ar)# end-group

RP/0/RP0/cpu 0: router(config)# group g-ospf100
RP/0/RP0/cpu 0: router(config-GRP)# router ospf '.*'
RP/0/RP0/cpu 0: router(config-GRP-ospf)# area '.*'
RP/0/RP0/cpu 0: router(config-GRP-ospf-ar)# cost 100
RP/0/RP0/cpu 0: router(config-GRP-ospf-ar)# end-group
```

これらの設定グループを次のように適用する場合、g-ospf2で指定されたコスト2は、グループがより内部の設定モードで適用されるため、OSPFエリア0によって継承されます。この場合、グループ g-ospf100 の設定は無視されます。

```
RP/0/RP0/cpu 0: router(config)# router ospf 0
RP/0/RP0/cpu 0: router(config-ospf)# apply-group g-ospf100
RP/0/RP0/cpu 0: router(config-ospf)# area 0
RP/0/RP0/cpu 0: router(config-ospf-ar)# apply-group g-ospf2
```

設定グループへの変更は自動的に継承される：例

ルータ構成にコミットされて適用されている設定グループに変更を加えると、変更はルータ構成によって自動的に継承されます。たとえば、次の構成がコミットされているとします。

```
group g-interface-mtu
  interface 'GigabitEthernet.*'
    mtu 1500
  !
end-group

interface POS0/0/0/0
  apply-group g-interface-mtu
!
```

ここで、次の例のように設定グループを変更します。

```
RP/0/RP0/cpu 0: router(config)# group g-interface-mtu
RP/0/RP0/cpu 0: router(config-GRP)# interface 'GigabitEthernet.*'
RP/0/RP0/cpu 0: router(config-GRP-if)# mtu 2000
RP/0/RP0/cpu 0: router(config-GRP-if)# end-group
```

この設定グループがコミットされると、インターフェイス GigabitEthernet0/0/0/0 の MTU 設定が自動的に 2000 に更新されます。

■ 設定グループへの変更は自動的に継承される：例