



システムインターフェイスとポート

この章では、コンテキストを作成し、コンテキスト内でシステムインターフェイスとポートを設定する方法について説明します。これらの手順を開始する前に、製品の設定情報について、製品固有のアドミニストレーションガイドを参照してください。



重要 システムインターフェイスとポートを設定する前に、少なくとも 1 枚のデータ処理カード (DPC/UDPC または DPC2/UDPC2) がアクティブであることを確認します。DPC のアクティブ化の詳細と手順については、このガイドの「システム設定」を参照してください。

- [コンテキスト \(1 ページ\)](#)
- [イーサネットインターフェイスとポート \(2 ページ\)](#)
- [VLANs \(6 ページ\)](#)

コンテキスト

特定の機能を実行するように複数のコンテキストを設定できる場合でも、それらはすべて同じ手順を使用して作成されます。

コンテキストの作成



重要 この項の設定例で使用されているコマンドは、最もよく使用されているコマンドまたはその可能性の高いコマンド、および/またはキーワードオプションです。多くの場合、他のコマンドやキーワードオプションを使用できます。すべてのコマンドの詳細については、『*Command Line Interface Reference*』を参照してください。

コンテキストを作成するには、次の設定例を適用します。

```
configure
context name
end
```

追加のコンテキストを設定するには、この手順を繰り返します。

コンテキストの表示と確認

手順

ステップ1 次のコマンドを入力して、コンテキストが正常に作成されたことを確認します。

```
[local]host_name# show context all
```

出力は、次の例のような2列のテーブルです。次の例は、2つのコンテキストが作成されたことを示しています。1つは「送信元」、もう1つは「接続先」という名前です。

Context Name	ContextID	State
local	1	Active
source	2	Active
destination	3	Active

左側の列には、現在設定されているコンテキストが一覧表示されます。中央の列には、設定されている各コンテキストに対応するコンテキストIDが一覧表示されます。3番目の列には、コンテキストの現在の状態が表示されます。

ステップ2 「設定の確認と保存」の章の説明に従って、設定を保存します。

ステップ3 これで、コンテキストが作成されて、コンテキスト内でインターフェイスと特定の機能を設定できるようになります。特定のサービスとオプションの設定手順については、他のセクションを参照してください。

イーサネットインターフェイスとポート

アプリケーションインターフェイスのタイプに関係なく、これを作成して設定する手順は次のとおりです。

手順

ステップ1 [インターフェイスの作成 \(3 ページ\)](#) の設定例を適用して、インターフェイスを作成し、IP アドレスとサブネットマスクを割り当てます。

ステップ2 [ポートの設定とインターフェイスへのバインド \(3 ページ\)](#) の設定例を適用して、インターフェイスで使用する物理ポートを割り当て、そのポートをインターフェイスにバインドします。

ステップ3 必要に応じて、[インターフェイスのスタティックルートの設定 \(4 ページ\)](#) の設定例を適用して、インターフェイスのスタティックルートを設定します。

ステップ4 設定するインターフェイスごとに上記の手順を繰り返します。

この項では、システムがネットワーク上で通信できるようにするためのインターフェイスとポートを設定するための最低限の手順について説明します。追加のインターフェイスまたはポートプロパティを設定するコマンドについては、『*Command Line Interface Reference*』の「*Ethernet Port Configuration Mode Commands*」および「*Ethernet Interface Configuration Mode Commands*」の章を参照してください。

システムラインカードとポートレベルの冗長性メカニズムが適切に機能するようにするには、任意のシステムポートに直接接続されているデバイスでスパニングツリープロトコルを無効にする必要があります。スパニングツリープロトコルをオフにしないと、冗長性メカニズムまたはサービスの停止に障害が発生する可能性があります。

インターフェイスの作成

コンテキストで新しいインターフェイスを作成するには、次の例を使用します。

```
configure
  context name
    interface name
      { ip | ipv6 } address address subnetmask [ secondary ]
    end
```

注：

- オプション：インターフェイスのタイプを「loopback」に設定して常にUPにし、物理ポートにバインドしないようにするには、**interface name** コマンドに **loopback** キーワードを追加します。
- オプション：複数のIPアドレスをインターフェイスに割り当てるには、**{ ip | ipv6 } address** コマンドに **secondary** キーワードを追加します。IPアドレスは、IPv4のドット付き10進表記またはIPv6のコロンで区切られた16進表記を使用して入力できます。
- オプション：指定したIPアドレスへの接続が失われた場合に、冗長ラインカード上のポートへのスイッチオーバー用のインターフェイスを設定するには、インターフェイス構成モードで、**port-switch-on-L3-fail address** コマンドを追加します。このIPアドレスは、IPv4のドット付き10進表記またはIPv6のコロンで区切られた16進表記を使用して入力できます。

ポートの設定とインターフェイスへのバインド

インターフェイスにポートを設定して割り当てるには、次の設定例を使用します。

```
configure
  port ethernet slot#/port#
    description description
    no shutdown
    bind interface interface_name context_name
  end
```

注：

- ポートイーサネット *slot#* の場合、MIO/UMIO カードが取り付けられている実際のシャーシスロットを使用します。スロット 5 または 6 のいずれかになります。
- ポートイーサネット *port#* の場合、ポート 10 ~ 19 (DC1) または 20 ~ 29 (DC2) を使用します。
- オプション：ポート設定を指定する場合は、イーサネットポートの構成モードで、優先スロット *slot#* コマンドを追加します。
- バインドにより、ポートとそのすべての設定が指名されたインターフェイスに関連付けます。
- UPF のポートがシャットダウンされ、その後起動した場合、ポートインターフェイスは Ubuntu バージョン 18.04 および NIC ドライバ i40e バージョン 2.12.6 で表示されます。これらのインターフェイスの BGP は自動的に回復しません。

UPF を完全に復元するには、UPF をリロードする必要があります。Ubuntu バージョン 20.04 および NIC ドライバ i40e バージョン 2.17.15 では、ポートインターフェイスと BGP の両方が自動的に回復します。

インターフェイスのスタティックルートの設定

インターフェイスのスタティックルートを設定するには、次の例を使用します。

```
configure
  context name
    { ip | ipv6 } route ip_address netmask next-hop gw_address interface_name
  end
```

注：

- *ip_address* と *netmask* は、ターゲットネットワークの IP アドレスとサブネットマスクです。この IP アドレスは、IPv4 のドット付き 10 進表記または IPv6 のコロンで区切られた 16 進表記を使用して入力できます。
- *gw_address* は、デフォルトゲートウェイまたはネクストホップルートの IP アドレスです。この IP アドレスは、IPv4 のドット付き 10 進表記または IPv6 のコロンで区切られた 16 進表記を使用して入力できます。
- ゲートウェイルータへのルートを設定するには、ネットワークおよびマスク変数に 209.165.200.225 を使用します。
- 必要に応じて繰り返します。優先ルートに障害が発生した場合に、代替の通信手段を提供するために、複数のスタティックルートを同じ接続先に設定できます。

ポート設定の表示と確認

手順

ステップ1 次のコマンドを入力して、インターフェイスの設定が正しいことを確認します。

```
[local]host_name# context context_name  
[context_name]host_name# show { ip | ipv6 } interface
```

context_name は、インターフェイスが作成されたコンテキストの名前を表します。これらのコマンドの出力は、以下の例のようになります。

この例では、*mgmt1* という名前のインターフェイスがローカルコンテキストで設定されています。

例：

この例では、*mgmt1* という名前のインターフェイスがローカルコンテキストで設定されています。

```
Intf Name:      mgmt1  
Intf Type:      Broadcast  
IP State:       UP (Bound to 10/11 untagged, ifIndex 285278209)  
IP Address:     192.168.100.3      Subnet Mask:    255.255.255.0  
Bcast Address:  192.168.100.255      MTU:           1500  
Resoln Type:    ARP              ARP timeout:    3600 secs  
Number of Secondary Addresses:  0  
Total interface count:          1
```

ステップ2 次のコマンドを入力して、ポートの設定が正しいことを確認します。

```
[context_name]host_name# show configuration port slot#/port#
```

slot# は、物理ポートがある MIO/UMIO のシャーシのスロット番号です。*slot#* は 5 または 6 のいずれかです。

このコマンドは、次の例に示すような出力を生成します。これは、シャーシスロット 5 に取り付けられた MIO/UMIO のポート 11 の設定を示しています。

例：

この例では、ポートは、*source* という名前のコンテキストで設定された *rp1* という名前のインターフェイスにバインドされています。

```
config  
  port ethernet 5/11  
    description MIO5/11_RP1  
    no shutdown  
    bind interface rp1 source  
  #end
```

例：

このコマンドは、次の例に示すような出力を生成します。これは、シャーシスロット 1 のポート 11 の設定を示しています。この例では、ポートは、*source* という名前のコンテキストで設定された *rp1* という名前のインターフェイスにバインドされています。

```
config  
  port ethernet 1/11
```

```

description 11_RP1
no shutdown
bind interface rp1 source
end

```

ステップ3 次のコマンドを入力して、スタティックルートが正しく設定されていることを確認します。

```
[context_name]host_name# show ip static-route
```

例：

このコマンドでは、次の例に示すような出力が生成されます。これは、209.165.200.226 の IP アドレスを持つゲートウェイへのスタティックルートを示しています。

Destination	Nexthop	Protocol	Prec	Cost	Interface
0.0.0.0/0	209.165.200.226	Static	0	0	MI01
0.0.0.0/0	209.165.200.226	Static	0	0	rp1 source

例：

このコマンドでは、次の例に示すような出力が生成されます。これは、209.165.200.226 の IP アドレスを持つゲートウェイへのスタティックルートを示しています。

Destination	Nexthop	Protocol	Prec	Cost	Interface
0.0.0.0/0	209.165.200.226	Static	0	0	vNIC1
0.0.0.0/0	209.165.200.226	Static	0	0	rp1 source

ステップ4 「設定の確認と保存」の章の説明に従って、設定を保存します。

VLANs

仮想 LAN (VLAN) を使用すると、2つの論理的に分離されたネットワークで同じ物理メディアを使用できます。また、VLAN セグメンテーション (802.1q タグ付けとも呼ばれる) は、VLAN ID を識別するタグを各イーサネットフレームに追加することによって機能します。

特定の packets タイプを処理するための VLAN の作成方法については、「VLAN」の章を参照してください。

ハイパーバイザ

KVM での VLAN の使用は、ブリッジインターフェイス共有の拡張機能です。違いは、ブリッジセットに参加するインターフェイスです。物理インターフェイス (eth0、eth1 など) は、各ゲストが使用するブリッジにバインドされます。これらのインターフェイスは、VLAN ID タグの有無にかかわらず、外部から着信するか、または内部で生成された未変更の packets を伝送します。

VMware は、1 台の vSphere ホスト上の仮想マシンが物理スイッチと同じプロトコルを使用して相互に通信できるようにする仮想スイッチの使用をサポートしています。vSwitch は、データリンク層でフレームを転送することによって、従来の物理イーサネットネットワークスイッチをエミュレートします。vSphere ホストは、仮想マシン用に 1,000 を超える内部仮想ポート

をそれぞれに備えた多数の仮想スイッチを搭載できます。vSphere プラットフォームは、ホストレベルの vSphere 標準スイッチの仮想スイッチ設定と、vSphere 分散型スイッチ（関連付けられた複数のホストを対象とする 1 つの仮想スイッチ）をサポートしています。

VLAN および管理ポート

管理インターフェイスは、VLAN 設定をサポートしています。このサポートはローカルコンテキストに拡張されます。

バルク統計情報は、通常の管理インターフェイス以外のインターフェイスから送信できます。このインターフェイスは、VLAN もサポートしています。

また、別の VLAN で他の OA&M サービスを設定することもできます。

OA&M サービスに個別の送信元 IP アドレスを割り当てることができます。OA&M サービスは、サービス VLAN と同じ VLAN にバインドすることはできません。サービスには、SGi、Gi、Pi、eGTP またはその他のパケットコア固有のインターフェイスとサービスが含まれます。

この機能は、**vlan** コマンドのサポートをローカルコンテキストの管理ポートに追加することによって実装されます。次のコマンドシーケンスの例を参照してください。

```
configure
  port ethernet 1/1
    vlan 184
    no shutdown
    bind interface 19/3-UHA foo
```


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。