

スイッチ スタックの管理

この章では、Catalyst スタックの管理に関する概念と手順について説明します。コマンドの構文および使用方法については、コマンドリファレンスを参照してください。

- 「スタックの概要」(P.5-1)
- 「スイッチ スタックの設定」(P.5-16)
- 「特定のスタック メンバへの CLI アクセス」(P.5-21)
- 「スタック情報の表示」(P.5-22)
- 「スタックのトラブルシューティング」(P.5-22)

スタック ポートを使用したスイッチの配線方法や LED を使用したスイッチ スタック ステータスの表示方法など、スイッチ スタックに関するその他の情報については、ハードウェア インストレーション ガイドを参照してください。

スタックの概要

スイッチ スタックは、ポートを使用して接続された Catalyst スイッチから構成されています。スイッチのうち 1 台がスタックの動作を制御します。このスイッチをスタック マスターと呼びます。スタック マスターおよびスタック内のその他のスイッチはスタック メンバです。レイヤ 2 プロトコルは、ネットワークに対してスイッチ スタック全体を 1 つのエントリティとして提供します。



(注)

スイッチ スタックはスイッチ クラスタとは異なります。スイッチ クラスタは、10/100/1000 ポートなどの LAN ポートを介して接続されたスイッチのセットです。スイッチ スタックとスイッチ クラスタの違いの詳細については、Cisco.com の『*Getting Started with Cisco Network Assistant*』の「Planning and Creating Clusters」の章を参照してください。

スタック マスターはスタック全体の単一管理ポイントです。スタック マスターから、次の機能を設定します。

- すべてのスタック メンバに適用されるシステムレベル (グローバル) の機能
- スタック メンバごとのインターフェイス レベルの機能

各スタック メンバは、固有のスタック メンバ番号によって識別されます。

すべてのスタック メンバはスタック マスターになることができます。スタック マスターが使用できなくなると、残りのスタック メンバの中から新しいスタック マスターが選択されます。スタック マスターを決めるための要素の 1 つがスタック メンバプライオリティ値です。最高のスタック メンバプライオリティ値を持つスイッチが、新しいスタック マスターになります。

スタック マスターでサポートされているシステムレベルの機能は、スタック全体でサポートされます。スタック マスターには、スタックの保存済みの実行コンフィギュレーション ファイルが格納されています。コンフィギュレーション ファイルには、スタックのシステムレベルの設定と、スタック メンバごとのインターフェイス レベルの設定が含まれます。各スタック メンバは、バックアップ目的でこれらのファイルの最新のコピーを保持します。

スイッチ スタックは、単一の IP アドレスを使用して管理します。IP アドレスは、システムレベルの設定値で、スタック マスターや他のスタック メンバ固有の設定値ではありません。スタックからスタック マスターや他のスタック メンバを削除しても、同じ IP アドレスを使用してスタックを管理できます。

次の方法を使用して、スタックを管理できます。

- Network Assistant (Cisco.com から入手できます)
- スタック メンバのコンソール ポートへのシリアル接続上のコマンドライン インターフェイス (CLI)
- 簡易ネットワーク管理プロトコル (SNMP) を介したネットワーク管理アプリケーション



(注) SNMP を使用して、サポートされる MIB によって定義されるスタック全体のネットワーク機能を管理します。スイッチは、スタックのメンバーシップや選択などのスタック構成固有の機能を管理するための MIB をサポートしません。

- CiscoWorks ネットワーク管理ソフトウェア

スタックを管理するには、次のことを理解している必要があります。

- スタックの形成に関する次の概念
 - 「スタックのメンバーシップ」 (P.5-3)
 - 「スタック マスターの選択」 (P.5-3)
- スタックとスタック メンバの設定に関する次の概念
 - 「スタックの MAC アドレス」 (P.5-4)
 - 「スタック メンバ番号」 (P.5-5)
 - 「スタック メンバプライオリティ値」 (P.5-5)
 - 「スタックのオフライン設定」 (P.5-6)
 - 「スタックのソフトウェア互換性に関する推奨事項」 (P.5-9)
 - 「スタック プロトコル バージョンの互換性」 (P.5-9)
 - 「スイッチ間のメジャー バージョン番号の非互換性」 (P.5-9)
 - 「スイッチ間のマイナー バージョン番号の非互換性」 (P.5-9)
 - 「互換性のないソフトウェアおよびスタック メンバイメージのアップグレード」 (P.5-13)
 - 「スタックのコンフィギュレーション ファイル」 (P.5-13)
 - 「スイッチ スタックのシステム全体の設定に関するその他の考慮事項」 (P.5-13)
 - 「スタックの管理接続」 (P.5-14)
 - 「スタックの設定のシナリオ」 (P.5-15)

スタックのメンバーシップ

スタンドアロンスイッチは、スタック マスターでもあるスタック メンバを1つ持つスタックです。スタンドアロンスイッチを別のスイッチと接続して (図 5-1 (P.5-3))、2つのスタック メンバで構成され、一方がスタック マスターであるスタックを構築できます。スタンドアロンスイッチを既存のスタックに接続して (図 5-2 (P.5-3))、スタック メンバーシップを増やすことができます。

スタック メンバを同一のモデルと交換すると、新しいスイッチは交換されたスイッチと同じ設定で機能します (新しいスイッチが交換されたスイッチと同じスタック メンバ番号を使用する場合)。スイッチスタックをプロビジョニングする利点については、「スタックのオフライン設定」(P.5-6)を参照してください。障害が発生したスイッチの交換については、ハードウェア インストールガイドの「Troubleshooting」の章を参照してください。

スタック マスターを削除したり、電源が入っているスタンドアロンスイッチまたはスタックを追加したりしなければ、メンバーシップの変更中もスタックの動作は途切れることなく継続されます。



(注) スタックの動作が中断されないように、スタックに追加または削除するスイッチの電源が切れていることを確認します。

メンバを追加または削除した後で、スタック リングがすべての帯域幅 (Gbps) で動作していることを確認します。スタック モード LED が点灯するまで、スタック メンバの Mode ボタンを押します。スタック内のすべてのスイッチで、最後の2つのポート LED がグリーンに点灯します。最後の2つのポート LED の一方または両方がグリーンでない場合、スタックは全帯域幅で動作していません。

- 電源が入っているスイッチを追加すると (マージ)、マージされているスタックのスタック マスターは自分たちの中からスタック マスターを選択します。新しいスタック マスターはマスターの役割と設定を保持し、スタック メンバもメンバの役割と設定を保持します。以前のスタック マスターを含む残りのすべてのスイッチは、リロードされ、スタック メンバとしてスタックに参加します。これらのスイッチは、スタック メンバ番号を使用可能な最小の番号に変更し、新しいスタック マスターの設定を使用します。
- 電源が入っているスタック メンバを取り外すと、スタックはそれぞれが同じ設定を持つ複数のスイッチスタックに分割 (パーティション化) されます。これにより、ネットワーク内で IP アドレス設定が競合することがあります。スタックを分割したまま使用する場合は、新しく作成されたスタックの IP アドレスを変更します。

図 5-1 2台のスタンドアロンスイッチからのスイッチスタックの構築

図 5-2 スタンドアロンスイッチのスイッチスタックへの追加

スイッチスタックのケーブル接続および電源投入の詳細については、ハードウェア インストールガイドの「Switch Installation」の章を参照してください。

スタック マスターの選択

スタック マスターは、次に示されている順序で次のいずれかの要素に基づいて選択されます。

1. 現在スタック マスターであるスイッチ
2. 最高のスタック メンバプライオリティ値を持つスイッチ



(注) スタック マスターにするスイッチに最高のプライオリティ値を割り当てることを推奨します。それによって、再選択時にはそのスイッチがスタック マスターとして選択されます。

スタック マスターは、次のイベントのいずれかが発生しない限り、その役割を維持します。

- スタックがリセットされた。*
- スタック マスターがスタックから取り外された。
- スタック マスターがリセットされたか、電源が切れた。
- スタック マスターに障害が発生した。
- 電源が入っているスタンドアロン スイッチまたはスイッチ スタックが追加されて、スタック メンバシップが増えた。*

アスタリスク (*) が付いているイベントでは、示されている要素に基づいて現在のスタック マスターが再選択される場合があります。

スタック全体に電源を入れるかリセットすると、一部のスタック メンバがスタック マスター選択に参加しない場合があります。

- 再選択には、すべてのスタック メンバが参加します。
- 同じ 20 秒の間に電源が投入されたスタック メンバは、スタック マスターの選択に参加し、スタック マスターとして選択される可能性があります。
- この 20 秒間後に電源が投入されたスタック メンバは、この初回の選択には参加せずにスタック メンバになります。

新しいスタック マスターは数秒後に使用可能になります。その間、スイッチ スタックはメモリ内の転送テーブルを使用してネットワークの中断を最小限に抑えます。新しいスタック マスターが選択され、リセットされている間、その他の使用可能なスタック メンバの物理インターフェイスは影響を受けません。

新しいスタック マスターが選択され、以前のスタック マスターが使用可能になっても、以前のスタック マスターはマスターとしての役割を再開しません。

スタック マスターの選択に影響を与える電源投入に関する考慮事項については、ハードウェア インストールガイドの「Switch Installation」の章を参照してください。

スタックの MAC アドレス

スタック マスターの MAC アドレスによってスタックの MAC アドレスが決定します。

スタックが初期化すると、スタック マスターの MAC アドレスによってネットワーク内のスタックを識別するブリッジ ID が決定されます。

スタック マスターが変わると、新しいスタック マスターの MAC アドレスによって新しいブリッジ ID が決定します。ただし、永続的 MAC アドレス機能がイネーブルの場合、スタックの MAC アドレスが変更されるまで約 4 分の遅延があります。この間、前のスタック マスターがスタックに再加入すると、そのスイッチが現在はスタック メンバであってスタック マスターではない場合でも、スタックはその MAC アドレスをスタックの MAC アドレスとして使用し続けます。以前のスタック マスターがこの間にスタックに再加入しない場合は、新しいスタック マスターの MAC アドレスがスタックの MAC アドレスになります。詳細については、「永続的 MAC アドレスのイネーブル化」(P.5-17) を参照してください。

スタック メンバ番号

メンバ番号 (1 ~) は、スタック内の各メンバを識別します。また、スタック メンバ番号によってスタック メンバが使用するインターフェイス レベルの設定が決定します。

新しいアウトオブボックス スイッチ (スタックに参加していないか、スタック メンバ番号が手動で割り当てられていないスイッチ) は、デフォルトのスタック メンバ番号 1 が設定された状態で出荷されています。スタックに参加すると、デフォルトのスタック メンバ番号はスタック内で使用可能な最小のメンバ番号に変更されます。

同じスタック内のメンバは、同じスタック メンバ番号を持つことはできません。

- **switch current-stack-member-number renumber new-stack-member-number** グローバル コンフィギュレーション コマンドを使用して手動でスタック メンバ番号を変更した場合、新しい番号が有効になるのはそのスタック メンバのリセット後 (または **reload slot stack-member-number** 特権 EXEC コマンドの使用後) で、その番号がまだ変更されていない場合だけです。

SWITCH_NUMBER 環境変数を使用してスタック メンバ番号を変更することもできます。

番号がスタック内の別のメンバによって使用されている場合、スイッチはスタック内で使用可能な最小の番号を選択します。

手動でスタック メンバ番号を変更し、その番号にインターフェイス レベルの設定が関連付けられていない場合は、そのスタック メンバはデフォルト設定にリセットされます。

プロビジョニングされたスイッチでは、**switch current-stack-member-number renumber new-stack-member-number** グローバル コンフィギュレーション コマンドを使用できません。使用すると、コマンドは拒否されます。

- スタック メンバを別のスイッチスタックへ移動した場合、スタック メンバは、番号がスタック内の別のメンバによって使用されていない場合にだけ自分の番号を保持します。番号がスタック内の別のメンバによって使用されている場合、スイッチはスタック内で使用可能な最小の番号を選択します。

スタック メンバの設定の詳細については、次の項を参照してください。

- スタック メンバ番号を変更する手順については、「メンバ番号の割り当て」(P.5-19) を参照してください。
- SWITCH_NUMBER 環境変数については、「環境変数の制御」(P.3-22) を参照してください。
- スタック メンバ番号および設定については、「スタックのコンフィギュレーション ファイル」(P.5-13) を参照してください。
- スタックのマージについては、「スタックのメンバーシップ」(P.5-3) を参照してください。

スタック メンバ プライオリティ値

スタック メンバのプライオリティ値が高いほど、スタック マスターとして選択され、そのメンバ番号を保持する可能性が高くなります。プライオリティ値は 1 ~ 15 の範囲で指定できます。デフォルトのプライオリティ値は 1 です。



(注)

スタック マスターにするスイッチに最高のプライオリティ値を割り当てることを推奨します。それによって、再選択時にはそのスイッチがスタック マスターとして選択されます。

新しいプライオリティ値はすぐに有効となりますが、現在のスタック マスターまたはスタックがリセットされるまで現在のスタック マスターには影響しません。

スタックのオフライン設定

オフライン設定機能を使用すると、新しいスイッチがスタックに参加する前に新しいスイッチの（設定の）プロビジョニングを実行できます。現在スタックに属していないスイッチに関連するスタックメンバー番号、スイッチタイプ、およびインターフェイスを設定できます。その設定をプロビジョニングされた設定といいます。スタックに追加され、この設定を使用するスイッチをプロビジョニングされたスイッチといいます。

プロビジョニングされた設定は、スイッチがスタックに追加されたときにプロビジョニングされた設定が存在しない場合、自動的に作成されます。**switch stack-member-number provision type** グローバルコンフィギュレーション コマンドを使用して、プロビジョニングされた設定を手動で作成できます。

プロビジョニングされたスイッチのインターフェイスを設定すると（たとえば、VLAN の一部として）、プロビジョニングされたスイッチがスタックに属しているかどうかに関係なく、その情報がスタックの実行コンフィギュレーションに表示されます。プロビジョニングされたスイッチのインターフェイスはアクティブではなく、特定の機能のディスプレイに表示されません（たとえば、**show vlan** ユーザ EXEC コマンドの出力）。**no shutdown** インターフェイス コンフィギュレーション コマンドを入力しても効果はありません。

スタートアップ コンフィギュレーション ファイルでは、プロビジョニングされたスイッチがスタックに属しているかどうかに関係なく、スタックは保存された情報をリロードして使用できます。

プロビジョニングされたスイッチのスタックへの追加による影響

プロビジョニングされたスイッチをスイッチスタックに追加すると、スタックはプロビジョニングされた設定またはデフォルト設定のいずれかを適用します。表 5-1 に、スイッチスタックがプロビジョニングされた設定とプロビジョニングされたスイッチを比較するときが発生するイベントを示します。

表 5-1 プロビジョニングされた設定とプロビジョニングされたスイッチの比較結果

シナリオ		結果
スタックメンバ番号およびスイッチタイプが一致する	<ol style="list-style-type: none"> 1. プロビジョニングされたスイッチのスタックメンバ番号と、スタックのプロビジョニングされた設定のスタックメンバ番号が一致する場合、かつ 2. プロビジョニングされたスイッチのスイッチタイプと、スタックのプロビジョニングされた設定のスイッチタイプが一致する場合 	スイッチスタックは、プロビジョニングされた設定をプロビジョニングされたスイッチに適用し、スタックに追加します。
スタックメンバ番号は一致するが、スイッチタイプが一致しない	<ol style="list-style-type: none"> 1. プロビジョニングされたスイッチのスタックメンバ番号と、スタックのプロビジョニングされた設定のスタックメンバ番号が一致する場合、ただし 2. プロビジョニングされたスイッチのスイッチタイプと、スタックのプロビジョニングされた設定のスイッチタイプが一致しない場合 	<p>スイッチスタックは、デフォルト設定をプロビジョニングされたスイッチに適用し、スタックに追加します。</p> <p>プロビジョニングされた設定は、新しい情報を反映するために変更されます。</p>
プロビジョニングされた設定でスタックメンバ番号が検出されない		<p>スイッチスタックは、デフォルト設定をプロビジョニングされたスイッチに適用し、スタックに追加します。</p> <p>プロビジョニングされた設定は、新しい情報を反映するために変更されます。</p>

表 5-1 プロビジョニングされた設定とプロビジョニングされたスイッチの比較結果 (続き)

シナリオ	結果
プロビジョニングされたスイッチのスタック メンバ番号が既存のスタック メンバと競合する	スタック マスターは、新しいスタック メンバ番号をプロビジョニングされたスイッチに割り当てます。 スタック メンバ番号およびスイッチ タイプが次のように一致します。 <ol style="list-style-type: none"> 1. プロビジョニングされたスイッチの新しいスタック メンバ番号と、スタックのプロビジョニングされた設定のスタック メンバ番号が一致する場合、かつ 2. プロビジョニングされたスイッチのスイッチ タイプと、スタックのプロビジョニングされた設定のスイッチ タイプが一致する場合
プロビジョニングされたスイッチのスタック メンバ番号が、プロビジョニングされた設定で検出されない	スタック メンバ番号は一致しますが、スイッチ タイプが一致しません。 <ol style="list-style-type: none"> 1. プロビジョニングされたスイッチのスタック メンバ番号と、スタックのプロビジョニングされた設定のスタック メンバ番号が一致する場合、ただし 2. プロビジョニングされたスイッチのスイッチ タイプと、スタックのプロビジョニングされた設定のスイッチ タイプが一致しない場合
プロビジョニングされたスイッチのスタック メンバ番号が、プロビジョニングされた設定で検出されない	スイッチ スタックは、デフォルト設定をプロビジョニングされたスイッチに適用し、スタックに追加します。

プロビジョニングされた設定で指定されているタイプとは異なるプロビジョニングされたスイッチを、電源が切られたスイッチ スタックに追加して電力を供給すると、スイッチ スタックはスタートアップ コンフィギュレーション ファイルの (現在は不正な) **switch stack-member-number provision type** グローバル コンフィギュレーション コマンドを拒否します。ただし、スタックの初期化中は、(間違っただけの可能性はある) プロビジョニングされたインターフェイスに対してスタートアップ コンフィギュレーション ファイル内のデフォルトでないインターフェイス設定情報が実行されます。実際のスイッチ タイプとプロビジョニング済みのスイッチ タイプの違いによって、拒否されるコマンドと受け入れられるコマンドがあります。



(注)

スイッチ スタックに新しいスイッチのプロビジョニングされた設定が含まれていない場合、スイッチはデフォルトのインターフェイス設定でスタックに参加します。スイッチ スタックは、新しいスイッチと一致する **switch stack-member-number provision type** グローバル コンフィギュレーション コマンドを実行コンフィギュレーションに追加します。

設定については、「[スタックの新しいスタック メンバのプロビジョニング](#)」(P.5-20) を参照してください。

スタックのプロビジョニングされたスイッチの交換による影響

スイッチ スタック内のプロビジョニングされたスイッチに障害が発生し、スタックから取り外して別のスイッチと交換する場合、スタックはプロビジョニングされた設定またはデフォルト設定をこのスイッチに適用します。スイッチ スタックがプロビジョニングされた設定とプロビジョニングされたスイッチを比較するときに発生するイベントは、「[プロビジョニングされたスイッチのスタックへの追加による影響](#)」(P.5-7) で説明されているイベントと同じです。

プロビジョニングされたスイッチのスタックからの取り外しによる影響

スイッチ スタックからプロビジョニングされたスイッチを取り外すと、取り外されたスタック メンバに関連付けられた設定は、プロビジョニングされた情報として実行コンフィギュレーション内に残ります。設定を完全に削除するには、**no switch stack-member-number provision** グローバル コンフィギュレーション コマンドを使用します。

スタックのソフトウェア互換性に関する推奨事項

スタック メンバ間でスタック プロトコルバージョンの互換性を確保するために、すべてのスタック メンバが同じ Cisco IOS ソフトウェア バージョンを実行している必要があります。

スタック プロトコルバージョンの互換性

スタック プロトコルバージョンには、メジャーバージョン番号とマイナーバージョン番号があります(たとえば、1.4 の場合、1 がメジャーバージョン番号、4 がマイナーバージョン番号になります)。

Cisco IOS ソフトウェアバージョンが同じスイッチは、スタック プロトコルバージョンも同じです。すべての機能がスタック全体で適切に動作します。スタック マスターと Cisco IOS ソフトウェアバージョンが同じスイッチは、すぐにスイッチ スタックに参加します。

非互換性が存在する場合、特定のスタック メンバの非互換性の原因を示すシステム メッセージが生成されます。スタック マスターは、このメッセージをすべてのスタック メンバに送信します。

詳細については、「[スイッチ間のメジャーバージョン番号の非互換性](#)」(P.5-9) の手順および「[スイッチ間のマイナーバージョン番号の非互換性](#)」(P.5-9) の手順を参照してください。

スイッチ間のメジャーバージョン番号の非互換性

Cisco IOS ソフトウェアバージョンが異なるスイッチは、スタック プロトコルバージョンも異なっている可能性があります。メジャーバージョン番号が異なるスイッチは非互換で、同じスタック内に存在できません。

スイッチ間のマイナーバージョン番号の非互換性

メジャーバージョン番号が同じでマイナーバージョン番号が異なるスイッチは、部分的に互換性があると見なされます。スタックに接続されている場合、部分的に互換性があるスイッチはバージョンミスマッチモードになり、完全に機能するメンバとしてスタックに参加できません。ソフトウェアは不一致ソフトウェアを検出すると、スタック イメージまたはスタック フラッシュ メモリの tar ファイル

イメージを使用して、バージョン不一致モードのスイッチをアップグレード（またはダウングレード）しようとしています。ソフトウェアでは、自動的なアップグレード（自動アップグレード）および自動的なアドバイス（自動アドバイス）機能を使用します。

バージョンミスマッチモードのスイッチ上のポート LED はオフのままです。Mode ボタンを押しても、LED モードは変更されません。



(注)

自動アドバイスおよび自動コピーでは、`info` ファイルの調査およびスイッチスタック上のディレクトリ構造の検索により、実行中のイメージを識別します。`archive download-sw` 特権 EXEC コマンドではなく、`copy tftp:` コマンドを使用してイメージをダウンロードすると、ディレクトリ構造が正しく作成されません。`info` ファイルの詳細については、「サーバまたは Cisco.com 上のイメージの tar ファイル形式」(P.A-25) を参照してください。

自動アップグレードおよび自動アドバイスの概要

ソフトウェアが一致しないソフトウェアを検出し、バージョンミスマッチモードのスイッチをアップグレードしようとする場合、自動的なアップグレードと自動的なアドバイスの2つのソフトウェアプロセスが実行されます。

- 自動的なアップグレード（自動アップグレード）プロセスには、自動コピープロセスと自動抽出プロセスがあります。デフォルトでは、自動アップグレードはイネーブルです（`boot auto-copy-sw` グローバル コンフィギュレーション コマンドがイネーブルです）。自動アップグレードをディセーブルにするには、スタック マスター上で `no boot auto-copy-sw` グローバル コンフィギュレーション コマンドを使用します。`show boot` 特権 EXEC コマンドを使用し、表示された *Auto upgrade* 行を確認することで、自動アップグレードのステータスを確認できます。
 - 自動コピーでは、スタック メンバ上で稼働しているソフトウェア イメージをバージョンミスマッチモードのスイッチに自動的にコピーしてそのスイッチをアップグレード（自動アップグレード）します。自動コピーが実行されるのは、自動アップグレードがイネーブルの場合、バージョンミスマッチモードのスイッチに十分なフラッシュ メモリがある場合、およびスタックで稼働しているソフトウェア イメージがバージョンミスマッチモードのスイッチに適している場合です。



(注)

バージョンミスマッチモードのスイッチでは、すべてのリリース済みソフトウェアが稼働するとは限りません。たとえば、新しいスイッチ ハードウェアは以前のバージョンのソフトウェアでは認識されません。

- 自動的な抽出（自動抽出）は、自動アップグレードプロセスでバージョンミスマッチモードのスイッチにコピーする適切なソフトウェアがスタック内で検出されない場合に実行されます。その場合、自動抽出プロセスは、バージョンミスマッチモードかどうかに関係なくスタック内のすべてのスイッチで、スイッチスタックまたはバージョンミスマッチモードのスイッチのアップグレードに必要な `tar` ファイルを検索します。`tar` ファイルは、スタック内のどのフラッシュ ファイル システムにあってもかまいません（バージョンミスマッチモードのスイッチを含む）。バージョンミスマッチモードのスイッチに適した `tar` ファイルが検出されると、このプロセスではそのファイルを抽出し、スイッチを自動的にアップグレードします。

自動アップグレード（自動コピーおよび自動抽出）プロセスは、一致しないソフトウェアが検出されて数分後に開始されます。

自動アップグレードプロセスが完了すると、バージョンミスマッチモードであったスイッチはリロードされ、完全に機能するメンバとしてスタックに参加します。リロード中に両方のケーブルが接続されている場合、スタックは2つのリング上で稼働するため、ネットワーク ダウンタイムが発生しません。

- 自動的なアドバイス（自動アドバイス）：自動アップグレードプロセスがバージョンミスマッチモードのスイッチにコピーする適切なバージョンミスマッチメンバソフトウェアを検出できない場合、自動アドバイスプロセスはスイッチスタックまたはバージョンミスマッチモードのスイッチを手動でアップグレードするために必要なコマンド（**archive copy-sw** または **archive download-sw** 特権 EXEC コマンド）およびイメージ名（tar ファイル名）を指示します。推奨されるイメージは、実行中のスタックイメージまたはスタック（バージョンミスマッチモードのスイッチを含む）内のいずれかのフラッシュファイルシステムの tar ファイルです。スタックのフラッシュファイルシステムで適切なイメージが検出されない場合、自動アドバイスプロセスによってスタックに新しいソフトウェアをインストールするように指示されます。自動アドバイスはディセーブルにできません。また、そのステータスを確認するコマンドはありません。

スタックソフトウェアおよびバージョンミスマッチモードのスイッチのソフトウェアに同じフィーチャセットが含まれていない場合は、自動アドバイスソフトウェアからの指示はありません。暗号化イメージおよび非暗号化イメージが稼働している場合も同様です。

異なるフィーチャセットを持つイメージをインストールするには、**archive-download-sw /allow-feature-upgrade** 特権 EXEC コマンドを使用します。

自動アップグレードおよび自動アドバイスのメッセージ例

マイナーバージョン番号が異なるスイッチをスタックに追加すると、メッセージが連続して表示されます（スイッチによってその他のシステムメッセージが生成されない場合）。

次に、スタックがスタックと異なるマイナーバージョン番号を実行している新しいスイッチを検出した例を示します。自動コピーが起動し、スタックメンバからバージョンミスマッチモードのスイッチにコピーするのに適したソフトウェアを検出し、バージョンミスマッチモードのスイッチをアップグレードして、リロードします。

```
*Mar 11 20:31:19.247:%STACKMGR-6-STACK_LINK_CHANGE:Stack Port 2 Switch 2 has changed to
state UP
*Mar 11 20:31:23.232:%STACKMGR-6-SWITCH_ADDED_VM:Switch 1 has been ADDED to the stack
(VERSION_MISMATCH)
*Mar 11 20:31:23.291:%STACKMGR-6-SWITCH_ADDED_VM:Switch 1 has been ADDED to the stack
(VERSION_MISMATCH) (Stack_1-3)
*Mar 11 20:33:23.248:%IMAGEMGR-6-AUTO_COPY_SW_INITIATED:Auto-copy-software process
initiated for switch number(s) 1
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Searching for stack member to act
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:as software donor...
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Found donor (system #2) for
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:member(s) 1
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:System software to be uploaded:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:System Type:                0x00000000
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:archiving (directory)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:archiving /.bin (4945851 bytes)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:archiving /info (450 bytes)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:archiving info (104 bytes)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:examining image...
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:extracting info (104 bytes)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:extracting /info (450 bytes)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:extracting info (104 bytes)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Stacking Version Number:1.4
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:System Type:                0x00000000
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: Ios Image File Size:  0x004BA200
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: Total Image File Size:0x00818A00
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: Minimum Dram required:0x08000000
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: Image Suffix:universalk9-122-53.SE
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: Image Directory:
```

```

*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: Image Name:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: Image
Feature:IP|LAYER_3|PLUS|MIN_DRAM_MEG=128
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Old image for switch 1:flash1:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: Old image will be deleted after download.
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Extracting images from archive into flash on
switch 1...
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: (directory)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:extracting / (4945851 bytes)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:extracting /info (450 bytes)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:extracting info (104 bytes)
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Installing (renaming):`flash1:update/' ->
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW: `flash1:'
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:New software image installed in flash1:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Removing old image:flash1:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:All software images installed.
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Requested system reload in progress...
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Software successfully copied to
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:system(s) 1
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Done copying software
*Mar 11 20:36:15.038:%IMAGEMGR-6-AUTO_COPY_SW:Reloading system(s) 1

```

次に、スタックがスタックと異なるマイナーバージョン番号を実行している新しいスイッチを検出した例を示します。自動コピーは起動しますが、スタックと互換性を持たせるための、バージョン mismatch モードのスイッチにコピーするソフトウェアをスタック内で検出できません。自動アドバース プロセスが起動し、ネットワークからバージョン mismatch モードのスイッチに tar ファイルをダウンロードするように推奨されます。

```

*Mar 1 00:01:11.319:%STACKMGR-6-STACK_LINK_CHANGE:Stack Port 2 Switch 2 has changed to
state UP
*Mar 1 00:01:15.547:%STACKMGR-6-SWITCH_ADDED_VM:Switch 1 has been ADDED to the stack
(VERSION_MISMATCH)
stack_2#
*Mar 1 00:03:15.554:%IMAGEMGR-6-AUTO_COPY_SW_INITIATED:Auto-copy-software process
initiated for switch number(s) 1
*Mar 1 00:03:15.554:%IMAGEMGR-6-AUTO_COPY_SW:
*Mar 1 00:03:15.554:%IMAGEMGR-6-AUTO_COPY_SW:Searching for stack member to act
*Mar 1 00:03:15.554:%IMAGEMGR-6-AUTO_COPY_SW:as software donor...
*Mar 1 00:03:15.554:%IMAGEMGR-6-AUTO_COPY_SW:Software was not copied
*Mar 1 00:03:15.562:%IMAGEMGR-6-AUTO_ADVISE_SW_INITIATED:Auto-advise-software process
initiated for switch number(s) 1
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:Systems with incompatible software
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:have been added to the stack. The
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:storage devices on all of the stack
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:members have been scanned, and it has
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:been determined that the stack can be
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:repaired by issuing the following
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:command(s):
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW: archive download-sw /force-reload
/overwrite /dest 1 flash1:.tar
*Mar 1 00:04:22.537:%IMAGEMGR-6-AUTO_ADVISE_SW:

```

archive download-sw 特権 EXEC コマンドの使用の詳細については、「ソフトウェアイメージの操作 (P.A-24)」を参照してください。

互換性のないソフトウェアおよびスタック メンバ イメージのアップグレード

archive copy-sw 特権 EXEC コマンドを使用すると、互換性のないソフトウェア イメージを持つスイッチをアップグレードして、既存のスタック メンバからソフトウェア イメージをコピーできます。このスイッチは新しいイメージで自動的にリロードされ、完全に機能するメンバとしてスタックに参加します。

スタックのコンフィギュレーション ファイル

スタック マスターは、スタックの保存済みの実行コンフィギュレーション ファイルを保持します。すべてのスタック メンバは、スタック マスターから定期的にコンフィギュレーション ファイルの同期化されたコピーを受け取ります。スタック マスターが使用できなくなると、スタック マスターの役割を引き受けたスタック メンバが最新のコンフィギュレーション ファイルを保持します。

- すべてのスタック メンバに適用されるシステムレベル（グローバル）のコンフィギュレーション 設定（IP、STP、VLAN、SNMP 設定など）
- 各スタック メンバに固有の、スタック メンバのインターフェイス固有のコンフィギュレーション 設定

スタックに参加する新しいアウトオブボックス スイッチは、そのスタックのシステムレベルの設定を使用します。スイッチは、別のスタックに移動されると保存済みのコンフィギュレーション ファイルを失い、新しいスタックのシステムレベルの設定を使用します。

各スタック メンバのインターフェイス固有の設定には、スタック メンバ番号が関連付けられます。スタック メンバは、番号が手動で変更されているか、同じスタック内の別のメンバによってすでに使用されている場合を除き、その番号を保持します。

- そのスタック メンバ番号のインターフェイス固有の設定が存在しない場合は、スタック メンバはデフォルトのインターフェイス固有の設定を使用します。
- そのスタック メンバ番号のインターフェイス固有の設定が存在する場合は、スタック メンバはそのメンバ番号に関連付けられたインターフェイス固有の設定を使用します。

障害が発生したスタック メンバを同一のモデルと交換した場合、交換後のスタック メンバは自動的に同じインターフェイス固有の設定を使用します。インターフェイス設定を再設定する必要はありません。交換後のスイッチは、障害が発生したスイッチと同じスタック メンバ番号を持つ必要があります。

スタンドアロン スイッチの設定の同じ方法で、スタック設定をバックアップして復元します。

詳細については、それぞれ次を参照してください。

- スイッチ スタックをプロビジョニングする利点については、「[スタックのオフライン設定](#)」(P.5-6)を参照してください。
- ファイル システムおよびコンフィギュレーション ファイルについては、[付録 A 「Cisco IOS ファイル システム、コンフィギュレーション ファイル、およびソフトウェア イメージの操作」](#)を参照してください。

スイッチ スタックのシステム全体の設定に関するその他の考慮事項

- Cisco.com から入手できる『*Getting Started with Cisco Network Assistant*』の「Planning and Creating Clusters」の章
- 「[QoS の設定](#)」(P.36-1)

スタックの管理接続

スタック マスターを使用して、スタックおよびスタック メンバのインターフェイスを管理します。CLI、SNMP、Network Assistant、および CiscoWorks ネットワーク管理アプリケーションを使用できます。スタック メンバを個々のスイッチとして管理することはできません。

- 「IP アドレスを使用したスタック」(P.5-14)
- 「SSH セッションを使用したスタック」(P.5-14)
- 「コンソール ポートを使用したスタック」(P.5-14)
- 「特定のスタック メンバ」(P.5-14)

IP アドレスを使用したスタック

スタックはシステムレベルの IP アドレスを使用して管理されます。スタックからスタック マスターまたは他のスタック メンバを取り外しても IP 接続があれば、そのまま同じ IP アドレスを使用してスタックを管理できます。



(注)

スタックからスタック メンバを取り外した場合、スタック メンバは自分の IP アドレスを保持します。そのため、ネットワーク内で 2 つのデバイスが同じ IP アドレスを持たないようにするために、スタックから取り外したスイッチの IP アドレスを変更します。

スイッチ スタックの設定に関連する情報については、「スタックのコンフィギュレーション ファイル」(P.5-13) を参照してください。

SSH セッションを使用したスタック

暗号化バージョンが稼働しているスタック マスターに障害が発生し、非暗号化バージョンが稼働しているスイッチと交換すると、スタックへの Secure Shell (SSH; セキュア シェル) 接続が失われることがあります。暗号化バージョンのソフトウェアが稼働しているスイッチをスタック マスターにすることを推奨します。

コンソール ポートを使用したスタック

1 台または複数のスタック メンバのコンソール ポートを使用してスタック マスターに接続できます。スタック マスターに複数の CLI セッションを使用する場合は注意が必要です。1 つのセッションで入力したコマンドは、別のセッションには表示されません。そのため、コマンドを入力したセッションを識別できなくなることがあります。

スタックを管理する場合は、CLI セッションを 1 つだけ使用することを推奨します。

特定のスタック メンバ

特定のスタック メンバ ポートを設定する場合は、CLI 表記にスタック メンバ番号を含める必要があります。

特定のメンバにアクセスするには、「特定のスタック メンバへの CLI アクセス」(P.5-21) を参照してください。

スタックの設定のシナリオ

表 5-2 に示すほとんどのシナリオでは、少なくとも 2 台のスイッチがポートを使用して接続されていることを前提にしています。

表 5-2 スイッチスタックの設定のシナリオ

シナリオ		結果
既存のスタック マスターによって明確に決定されるスタック マスター選択	ポートを使用して 2 つの電源の入ったスタックを接続します。	2 つのスタック マスターの一方だけが新しいスタック マスターになります。
スタック メンバプライオリティ値によって明確に決定されるスタック マスター選択	<ol style="list-style-type: none"> ポートを使用して、2 台のスイッチを接続します。 switch stack-member-number priority new-priority-number グローバル コンフィギュレーション コマンドを使用して、一方のスタック メンバにより高いスタック メンバプライオリティ値を設定します。 両方のスタック メンバを同時に再起動します。 	高い方のプライオリティ値を持つスタック メンバがスタック マスターに選択されます。
コンフィギュレーション ファイルによって明確に決定されるスタック マスター選択	<p>両方のスタック メンバが同じプライオリティ値を持つと仮定します。</p> <ol style="list-style-type: none"> 一方のスタック メンバがデフォルト設定を持ち、他方のスタック メンバが保存済み（デフォルトでない）のコンフィギュレーション ファイルを持つことを確認します。 両方のスタック メンバを同時に再起動します。 	保存済みのコンフィギュレーション ファイルを持つスタック メンバがスタック マスターに選択されます。
MAC アドレスによって明確に決定されるスタック マスター選択	両方のスタック メンバが同じプライオリティ値、コンフィギュレーション ファイル、およびソフトウェア イメージを持つと仮定して、両方のスタック メンバを同時に再起動します。	小さい方の MAC アドレスを持つスタック メンバがスタック マスターに選択されます。
スタック メンバ番号の競合	<p>一方のスタック メンバが他方のスタック メンバより高いプライオリティ値を持つと仮定します。</p> <ol style="list-style-type: none"> 両方のスタック メンバが同じメンバ番号を持っていることを確認します。必要に応じて、switch current-stack-member-number renumber new-stack-member-number グローバル コンフィギュレーション コマンドを使用します。 両方のスタック メンバを同時に再起動します。 	高い方のプライオリティ値を持つスタック メンバが、自分のメンバ番号を保持します。他のスタック メンバは新しいメンバ番号を持ちます。
スタック メンバの追加	<ol style="list-style-type: none"> 新しいスイッチの電源を切ります。 ポートを使用して、新しいスイッチを電源が入っているスタックに接続します。 新しいスイッチの電源を入れます。 	スタック マスターはそのままです。新しいスイッチがスタックに追加されます。

表 5-2 スイッチ スタックの設定のシナリオ (続き)

シナリオ		結果
スタック マスターの障害	スタック マスターを取り外します (または電源を切ります)。	残りのスタック メンバのいずれかが新しいスタック マスターになります。スタック内の他のすべてのスタック メンバはメンバのままで、再起動されません。
を超えるメンバの追加	<ol style="list-style-type: none"> 1. ポートを使用して、10 台のスイッチを接続します。 2. すべてのスイッチの電源を入れます。 	<p>2 台のスイッチがスタック マスターになります。一方のマスターにスタック メンバが属します。もう一方のスタック マスターはスタンドアロン スイッチとして維持されます。</p> <p>スイッチの Mode ボタンとポート LED を使用して、どのスイッチがスタック マスターで、各スタック マスターにどのスイッチが属しているかを識別できます。Mode ボタンと LED の詳細については、ハードウェア インストールガイドを参照してください。</p>

スイッチ スタックの設定

- ・「デフォルトのスイッチ スタック設定」(P.5-16)
- ・「永続的 MAC アドレスのイネーブル化」(P.5-17)
- ・「スタック メンバ情報の割り当て」(P.5-19)
- ・「スタック メンバーシップの変更」(P.5-21)

デフォルトのスイッチ スタック設定

表 5-3 に、デフォルトのスイッチ スタック設定を示します。

表 5-3 デフォルトのスイッチ スタック設定

機能	デフォルト設定
スタック MAC アドレス タイマー	ディセーブル。
スタック メンバ番号	1
スタック メンバ プライオリティ値	1
オフライン設定	スイッチ スタックはプロビジョニングされていません。
永続的 MAC アドレス	ディセーブル。

永続的 MAC アドレスのイネーブル化

スタック マスターの MAC アドレスによってスタックの MAC アドレスが決定します。スタック マスターがスタックから取り外されて新しいスタック マスターに引き継がれた場合、新しいスタック マスターの MAC アドレスが新しいスタック MAC アドレスになります。ただし、スタック MAC アドレスが変更されるまでの遅延時間を設定できる永続的 MAC アドレス機能を設定できます。この間に以前のスタック マスターがスタックに再加入すると、そのスタック マスターが現在はスタック メンバであってスタック マスターではない場合でも、スタックはその MAC アドレスをスタック MAC アドレスとして引き続き使用します。また、スタック MAC アドレスが新しいスタック マスターの MAC アドレスに変更されないように、スタックの MAC 永続性を設定することもできます。

**注意**

この機能を設定すると、警告メッセージに設定の結果が表示されます。この機能は慎重に使用してください。古いスタック マスターの MAC アドレスをドメイン内で使用すると、トラフィックが失われることがあります。

時間は 0 ～ 60 分の範囲で指定できます。

- このコマンドに値を入力しない場合、デフォルトの遅延は 4 分です。必ず値を入力することを推奨します。コンフィギュレーション ファイルには、遅延時間が明示タイマー値 4 分として表示されます。
- 0 を入力すると、スタック MAC アドレスを現在のスタック マスターの MAC アドレスに変更する **no stack-mac persistent timer** グローバル コンフィギュレーション コマンドを入力するまで、以前のスタック マスターのスタック MAC アドレスが使用されます。このコマンドを入力しないと、スタック MAC アドレスは変更されません。
- 1 ～ 60 分の遅延時間を入力した場合は、設定した時間が経過するか、**no stack-mac persistent timer** コマンドを入力するまで、以前のスタック マスターのスタック MAC アドレスが使用されません。


この間に以前のスタック マスターがスタックに再加入しない場合は、スタックは新しいスタック マスターの MAC アドレスをスタック MAC アドレスとして使用します。

**(注)**

スイッチ スタック全体をリロードする場合、スタックはスタック マスターの MAC アドレスをスタック MAC アドレスとして取得します。

■ スイッチ スタックの設定

永続的 MAC アドレスをイネーブルにするには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

	コマンド	目的
ステップ1	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2	stack-mac persistent timer [0 <i>time-value</i>]	<p>スタック マスターが変更された後、スタック MAC アドレスが新しいスタック マスターの MAC アドレスに変更されるまでの遅延時間をイネーブルにします。この間に以前のスタック マスターがスタックに再加入した場合、スタックはその MAC アドレスをスタック MAC アドレスとして使用します。</p> <ul style="list-style-type: none"> • 値を指定しないでコマンドを入力すると、デフォルトの遅延 4 分が設定されます。必ず値を指定することを推奨します。 • 現在のスタック マスターの MAC アドレスを無期限に使用するには、0 を入力します。 • スタック MAC アドレスが新しいスタック マスターの MAC アドレスに変更されるまでの時間 (分) を設定するには、<i>time-value</i> に 1 ~ 60 の範囲内の値を入力します。 <p> 注意 このコマンドを入力すると、古いスタック マスターの MAC アドレスがネットワーク ドメイン内にあるとトラフィックが失われる可能性があることを示す警告が表示されます。</p> <p>新しいスタック マスターが引き継いでから有効期間が切れる前に no stack-mac persistent timer コマンドを入力すると、スタックは現在のスタック マスターの MAC アドレスを使用します。</p>
ステップ3	end	特権 EXEC モードに戻ります。
ステップ4	show running-config または show switch	<p>スタック MAC アドレス タイマーがイネーブルであることを確認します。</p> <p>出力には、stack-mac persistent timer と時間が分単位で表示されます。</p> <p>出力には、Mac persistency wait time、設定されている分数、およびスタック MAC アドレスが表示されます。</p>
ステップ5	copy running-config startup-config	(任意) コンフィギュレーション ファイルに設定を保存します。

永続的 MAC アドレス機能をディセーブルにするには、**no stack-mac persistent timer** グローバル コンフィギュレーション コマンドを使用します。

次に、永続的 MAC アドレス機能に 7 分の遅延時間を設定し、設定を確認する例を示します。

```
Switch(config)# stack-mac persistent timer 7
WARNING: The stack continues to use the base MAC of the old Master
WARNING: as the stack MAC after a master switchover until the MAC
WARNING: persistency timer expires. During this time the Network
WARNING: Administrators must make sure that the old stack-mac does
WARNING: not appear elsewhere in this network domain. If it does,
WARNING: user traffic may be blackholed.
Switch(config)# end
Switch# show switch
Switch/Stack Mac Address : 0016.4727.a900
Mac persistency wait time: 7 mins

                H/W   Current
Switch#  Role   Mac Address      Priority Version  State
-----
*1      Master 0016.4727.a900    1         0         Ready
```

スタック メンバ情報の割り当て

- 「メンバ番号の割り当て」(P.5-19) (任意)
- 「スタック メンバプライオリティ値の設定」(P.5-20) (任意)
- 「スタックの新しいスタック メンバのプロビジョニング」(P.5-20) (任意)

メンバ番号の割り当て



(注)

この作業を実行できるのはスタック マスターからだけです。

スタック メンバ番号をスタック メンバに割り当てるには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

	コマンド	目的
ステップ1	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2	switch current-stack-member-number renumber new-stack-member-number	スタック メンバの現在のメンバ番号と新しいメンバ番号を指定します。指定できる範囲は 1 ~ です。 show switch ユーザ EXEC コマンドを使用すると、現在のスタック メンバ番号を表示できます。
ステップ3	end	特権 EXEC モードに戻ります。
ステップ4	reload slot stack-member-number	スタック メンバをリセットします。
ステップ5	show switch	スタック メンバ番号を確認します。
ステップ6	copy running-config startup-config	コンフィギュレーション ファイルに設定を保存します。

スタック メンバ プライオリティ値の設定



(注) この作業を実行できるのはスタック マスターからだけです。

プライオリティ値をスタック メンバに割り当てるには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

	コマンド	目的
ステップ 1	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 2	switch stack-member-number priority new-priority-number	スタック メンバのメンバ番号と新しいプライオリティ値を指定します。メンバ番号の範囲は 1 ～ です。プライオリティ値の範囲は 1 ～ 15 です。 show switch ユーザ EXEC コマンドを使用すると、現在のプライオリティ値を表示できます。 新しいプライオリティ値はすぐに有効となりますが、現在のスタック マスターまたはスタック がリセットされるまで現在のスタック マスターには影響しません。
ステップ 3	end	特権 EXEC モードに戻ります。
ステップ 4	reload slot stack-member-number	スタック メンバをリセットし、この設定を適用します。
ステップ 5	show switch stack-member-number	スタック メンバ プライオリティ値を確認します。
ステップ 6	copy running-config startup-config	(任意) コンフィギュレーション ファイルに設定を保存します。

SWITCH_PRIORITY 環境変数を設定することもできます。詳細については、「[環境変数の制御 \(P.3-22\)](#)」を参照してください。

スタックの新しいスタック メンバのプロビジョニング



(注) この作業を実行できるのはスタック マスターからだけです。

スタックに新しいスタック メンバをプロビジョニングするには、特権 EXEC モードで次の手順を実行します。この手順は任意です。

	コマンド	目的
ステップ 1	show switch	スタックに関するサマリー情報を表示します。
ステップ 2	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	switch stack-member-number provision type	プロビジョニングされたスイッチのスタック メンバ番号を指定します。デフォルトでは、スイッチはプロビジョニングされません。 <i>stack-member-number</i> の範囲は 1 ～ です。スタックで使用されていないスタック メンバ番号を入力します。ステップ 1 を参照してください。 <i>type</i> には、スタック メンバのモデル番号を入力します。
ステップ 4	end	特権 EXEC モードに戻ります。
ステップ 5	show running-config	設定内のインターフェイスの番号付けが正しいことを確認します。

	コマンド	目的
ステップ 6	<code>show switch stack-member-number</code>	プロビジョニングされたスイッチのステータスを確認します。 <code>stack-member-number</code> には、ステップ 2 と同じ番号を入力します。
ステップ 7	<code>copy running-config startup-config</code>	(任意) コンフィギュレーション ファイルに設定を保存します。

プロビジョニングされた情報を削除し、エラー メッセージを受信しないようにするには、このコマンドの **no** 形式を使用する前に、指定されたスイッチをスタックから取り外します。

次に、スタックにスタック メンバ番号が 2 のスイッチをプロビジョニングする例を示します。**show running-config** コマンドの出力は、プロビジョニングされたスイッチに関連付けられたインターフェイスを示します。

```
Switch(config)# switch 2 provision switch PID
Switch(config)# end
Switch# show running-config | include switch 2
!
interface GigabitEthernet2/0/1
!
interface GigabitEthernet2/0/2
!
interface GigabitEthernet2/0/3
<output truncated>
```

スタック メンバシップの変更

スタックを分割しないで、電源が入ったスタック メンバを取り外す場合、次の手順を実行します。

- ステップ 1 新しく構築したスタックの電源を切ります。
- ステップ 2 ポートを使用して、元のスタックに再接続します。
- ステップ 3 スイッチの電源を入れます。

特定のスタック メンバへの CLI アクセス



(注) この作業はデバッグだけを目的とし、実行できるのはスタック マスターからだけです。

remote command {all | stack-member-number} 特権 EXEC コマンドを使用して、すべてまたは特定のスタック メンバにアクセスできます。スタック メンバ番号の範囲は 1 ~ です。

session stack-member-number 特権 EXEC コマンドを使用して、特定のスタック メンバにアクセスできます。スタック メンバ番号は、システム プロンプトに追加されます。たとえば、スタック メンバ 2 のプロンプトは Switch-2#、スタック マスターのプロンプトは Switch# です。スタック マスターの CLI セッションに戻るには、**exit** と入力します。特定のスタック メンバ上では、**show** コマンドと **debug** コマンドだけが使用できます。

詳細については、「[インターフェイス コンフィギュレーション モードの使用法](#)」(P.13-19) を参照してください。

スタック情報の表示

特定のスタックメンバまたはスタックをリセットした後で保存済みの設定変更を表示するには、次の特権 EXEC コマンドを使用します。

表 5-4 スタック情報を表示するコマンド

コマンド	説明
<code>show platform stack passive-links all</code>	スタックプロトコルバージョンなど、すべてのスイッチスタック情報を表示します。
<code>show switch</code>	プロビジョニングされたスイッチおよびバージョンミスマッチモードのスイッチのステータスなど、スタックに関するサマリー情報を表示します。
<code>show switch stack-member-number</code>	特定のスタックメンバに関する情報を表示します。
<code>show switch detail</code>	スタックリングに関する詳細情報を表示します。
<code>show switch neighbors</code>	スタックのネイバーを表示します。
<code>show switch stack-ports</code>	スタックのポート情報を表示します。

スタックのトラブルシューティング

- ・「[手動でのポートのディセーブル化](#)」(P.5-22)
- ・「[別のメンバがスタート中のポートの再イネーブル化](#)」(P.5-23)
- ・「[show switch stack-ports summary コマンドの出力の概要](#)」(P.5-23)

手動でのポートのディセーブル化

ポートがフラッピングしていることが原因で、スタックリングが不安定になるためにポートをディセーブルにするには、`switch stack-member-number stack port port-number disable` 特権 EXEC コマンドを入力します。ポートを再びイネーブルにするには、`switch stack-member-number stack port port-number enable` コマンドを入力します。



(注)

`switch stack-member-number stack port port-number disable` コマンドの使用には注意が必要です。ポートをディセーブルにすると、スタックは半分の帯域幅で稼働します。

- ・すべてのメンバがポートを通じて接続されていて、ready ステートの場合は、スタックは *full-ring* ステートになります。
- ・スタックが *partial-ring* ステートになるのは次のような場合です。
 - すべてのメンバがスタックポートを通じて接続されているが、一部またはすべてのメンバが準備完了状態ではない。
 - ポートを通じて接続されていないメンバがある。

`switch stack-member-number stack port port-number disable` 特権 EXEC コマンドを入力した場合、次のようになります。

- ・スタックが *full-ring* ステートのときは、1つのポートだけをディセーブルにできます。次のメッセージが表示されます。

```
Enabling/disabling a stack port may cause undesired stack changes. Continue?[confirm]
```


- スタックが **partial-ring** ステートのときは、ポートをディセーブルにできません。次のメッセージが表示されます。

Disabling stack port not allowed with current stack configuration.

別のメンバがスタート中のポートの再イネーブル化

スイッチ 1 のポート 1 がスイッチ 4 のポート 2 に接続されています。ポート 1 でフラッピングが発生した場合、**switch 1 stack port 1 disable** 特権 EXEC コマンドを使用してポート 1 をディセーブルにします。

スイッチ 1 のポート 1 がディセーブルで、スイッチ 1 の電源が入ったままのときに、次の手順を実行します。

- スイッチ 1 のポート 1 とスイッチ 4 のポート 2 の間のスタック ケーブルを取り外します。
- スタックからスイッチ 4 を取り外します。
- スイッチを追加してスイッチ 4 を交換し、スイッチ番号 4 を割り当てます。
- スイッチ 1 のポート 1 とスイッチ 4 (交換後のスイッチ) のポート 2 の間のケーブルを再接続します。
- スイッチ間のリンクを再びイネーブルにします。**switch 1 stack port 1 enable** 特権 EXEC コマンドを入力して、スイッチ 1 のポート 1 をイネーブルにします。
- スイッチ 4 の電源を入れます。



注意

スイッチ 1 のポート 1 をイネーブルにする前にスイッチ 4 の電源を入れると、スイッチのいずれかがリロードされる場合があります。

最初にスイッチ 4 の電源を入れると、リンクを起動するために **switch 1 stack port 1 enable** および **switch 4 stack port 2 enable** 特権 EXEC コマンドを入力する必要がある場合があります。

show switch stack-ports summary コマンドの出力の概要

スタック メンバ 2 のポート 1 だけがディセーブルです。

```
Switch# show switch stack-ports summary
Switch#/  Stack  Neighbor  Cable  Link  Link  Sync  #  In
Port#/   Port   Port      Length OK   Active OK  Changes  Loopback
          Status
-----
1/1      OK      3         50 cm  Yes  Yes  Yes   1      No
1/2      Down   None      3 m    Yes  No   Yes   1      No
2/1      Down   None      3 m    Yes  No   Yes   1      No
2/2      OK      3         50 cm  Yes  Yes  Yes   1      No
3/1      OK      2         50 cm  Yes  Yes  Yes   1      No
3/2      OK      1         50 cm  Yes  Yes  Yes   1      No
```

表 5-5 show switch stack-ports summary コマンドの出力

フィールド	説明
Switch#/Port#	メンバ番号とそのポート番号。
Stack Port Status	<ul style="list-style-type: none"> • Absent : ポートでケーブルが検出されません。 • Down : ケーブルは検出されていますが、接続されたネイバーが動作していないか、またはポートがディセーブルになっています。 • OK : ケーブルが検出され、接続済みのネイバーが起動しています。
Neighbor	ケーブルのもう一方の終端のアクティブなメンバのスイッチ番号
Cable Length	有効な長さは 50 cm、1 m、または 3 m です。 スイッチがケーブルの長さを検出できない場合は、値は <i>no cable</i> になります。ケーブルが接続されていないか、リンクが信頼できない可能性があります。
Link OK	これは、リンクが安定しているかどうかを示します。 リンク パートナーとは、ネイバー スイッチ上のポートを意味していません。 <ul style="list-style-type: none"> • No : リンクの相手側は、ポートから無効なプロトコル メッセージを受信します。 • Yes : リンクの相手側は、ポートから有効なプロトコル メッセージを受信します。
Link Active	このポートがリンク パートナーと同じ状態にある場合に表示されます。 <ul style="list-style-type: none"> • No : ポートはリンクの相手側にトラフィックを送信できません。 • Yes : ポートはリンクの相手側にトラフィックを送信できます。
Sync OK	<ul style="list-style-type: none"> • No : リンク パートナーからこのポートに有効なプロトコル メッセージが送信されませんでした。 • Yes : リンクの相手側は、ポートに有効なプロトコル メッセージを送信します。
# Changes to LinkOK	これは、リンクの相対的安定性を示します。 短時間で多数の変更が行われた場合は、リンクのフラップが発生することがあります。
In Loopback	<ul style="list-style-type: none"> • No : メンバの少なくとも 1 つのポートにケーブルが接続されています。 • Yes : メンバのどのポートにも、ケーブルが接続されていません。