



## CHAPTER 8

# RPR および SSO を使用したスーパーバイザエンジンの冗長設定

Catalyst 4500 シリーズ スイッチでは、アクティブ スーパーバイザ エンジンが故障した場合に、冗長スーパーバイザ エンジンが処理を引き継ぎます。ソフトウェアでは、冗長スーパーバイザ エンジンを Route Processor Redundancy (RPR; ルート プロセッサ冗長) またはステートフル スイッチオーバー (SSO) 動作モードで稼働することで、スーパーバイザ エンジン冗長構成をイネーブルにします。



(注) SSO を稼働させるための ROMmon の最小要件は、Cisco IOS Release 12.1(20r)EW1 または Cisco IOS Release 12.2(20r)EW1 となります。

この章では、Catalyst 4507R および Catalyst 4510R スイッチ上でスーパーバイザ エンジンの冗長構成を設定する方法について説明します。



(注) Cisco Nonstop Forwarding (NSF) with SSO (NSF/SSO) の詳細については、第 9 章「Cisco NSF/SSO スーパーバイザ エンジンの冗長構成の設定」を参照してください。

この章の主な内容は、次のとおりです。

- 「スーパーバイザ エンジンの冗長構成の概要」 (P.8-2)
- 「スーパーバイザ エンジンの冗長構成の同期化の概要」 (P.8-5)
- 「スーパーバイザ エンジンの冗長構成に関する注意事項および制約事項」 (P.8-6)
- 「スーパーバイザ エンジンの冗長構成の設定」 (P.8-8)
- 「手動による切り替え」 (P.8-13)
- 「ソフトウェア アップグレードの実行」 (P.8-14)
- 「冗長スーパーバイザ エンジンでの Bootflash 操作」 (P.8-16)



(注) この章のスイッチ コマンドの構文および使用方法の詳細については、『Catalyst 4500 Series Switch Cisco IOS Command Reference』および次の URL の関連マニュアルを参照してください。

<http://www.cisco.com/en/US/products/ps6350/index.html>

# スーパーバイザ エンジンの冗長構成の概要

ここでは、スーパーバイザ エンジンの冗長構成について説明します。

- 「概要」(P.8-2)
- 「RPR 動作」(P.8-3)
- 「SSO の動作」(P.8-3)
- 「スーパーバイザ エンジンの冗長構成の同期化の概要」(P.8-5)

## 概要

スーパーバイザ エンジンの冗長性が有効になっていれば、アクティブ スーパーバイザ エンジンが故障した場合や手動スイッチオーバーが実行された場合に、冗長スーパーバイザ エンジンがアクティブ スーパーバイザ エンジンとなります。冗長スーパーバイザ エンジンはアクティブ スーパーバイザ エンジンのスタートアップ コンフィギュレーション時に自動的に初期化されており、このことによって、スイッチオーバー時間が短縮されます（設定に応じて、RPR モードでは 30 秒以上、SSO モードでは 1 秒以下）。

スーパーバイザ エンジンの冗長性は、スイッチオーバー時間の削減以外にも次の内容をサポートしています。

- 冗長スーパーバイザ エンジンの活性挿抜（OIR）  
スーパーバイザ エンジンの冗長構成により、メンテナンス時に冗長スーパーバイザ エンジンの OIR が可能になります。冗長スーパーバイザ エンジンが搭載されている場合、アクティブ スーパーバイザ エンジンがその存在を検出し、冗長スーパーバイザ エンジンは RPR モードでは部分的初期化ステート、および SSO モードでは完全初期化ステートで起動します。
- ソフトウェアのアップグレード（「ソフトウェア アップグレードの実行」(P.8-14) を参照）。  
スーパーバイザ エンジンのソフトウェア変更中のダウン時間を最小限にするために、冗長スーパーバイザ エンジンに新しいイメージをロードしてスイッチオーバーを実施します。

スイッチを最初に起動して、最初に起動するスーパーバイザ エンジンがアクティブ スーパーバイザ エンジンとなって、スイッチオーバーが発生するまでアクティブのままとなります。

次のイベントが 1 つまたは複数発生するとスイッチオーバーが発生します。

- アクティブ スーパーバイザ エンジンの障害（ハードウェアまたはソフトウェア機能による）、または取り外し
- ユーザによる強制的なスイッチオーバー
- ユーザによるアクティブ スーパーバイザ エンジンのリロード

表 8-1 では、冗長用のシャーシおよびスーパーバイザ エンジンについて説明します。

表 8-1 シャーシおよびスーパーバイザのサポート

シャーシ (製品番号)	サポートされるスーパーバイザ エンジン
Catalyst 4507R (WS-C4507R)	冗長 Supervisor Engine II-Plus (WS-X4013+)、冗長 Supervisor Engine II-Plus (WS-X4013+GE)、Supervisor Engine IV (WS-X4515)、冗長 Supervisor Engine V (WS-X4516)、および冗長 Supervisor Engine V (WS-X4516-10GE)
Catalyst 4510R (WS-C4510R)	冗長 Supervisor Engine V (WS-X4516) および冗長 Supervisor Engine V (WS-X4516-10GE)

## RPR 動作

RPR は、Cisco IOS Release 12.2(12c)EW 以降のリリースでサポートされます。冗長スーパーバイザ エンジンは、RPR モードで稼働する場合、部分的に初期化された状態で起動し、アクティブ スーパーバイザ エンジンの固定コンフィギュレーションと同期化されます。



(注) 永続的なコンフィギュレーションには、`startup-config`、ブート変数、`config-register`、VLAN データベースが含まれます。

冗長スーパーバイザ エンジンは基本的なシステム初期化のあとで起動シーケンスを中止します。アクティブ スーパーバイザ エンジンに障害が発生した場合、冗長スーパーバイザ エンジンが新しいアクティブ スーパーバイザ エンジンになります。

スーパーバイザ エンジンのスイッチオーバーでは、モジュール タイプとステータスに関連するスーパーバイザ エンジンの間ではステートが維持されず、RPR モードの物理ポートすべてが再起動するので、トラフィックが中断します。冗長スーパーバイザ エンジンの初期化が完了すると、冗長スーパーバイザ エンジンはモジュールからハードウェア情報を読み込みます。

## SSO の動作



(注) ステートフル スwitchオーバーは、Supervisor Engine 6-E ではサポートされていません。

SSO は、Cisco IOS Release 12.2(20)EWA 以降のリリースでサポートされています。冗長スーパーバイザ エンジンは、SSO モードで稼働する場合、完全に初期化された状態で起動し、アクティブ スーパーバイザ エンジンの固定コンフィギュレーションおよび実行コンフィギュレーションと同期化します。そのあと、冗長スーパーバイザ エンジンは、次のプロトコルのステートを維持し、ステートフル スwitchオーバーをサポートする機能に関するハードウェアおよびソフトウェア ステートの変更すべてを同期化して維持します。そのため、冗長スーパーバイザ エンジン構成内のレイヤ 2 セッションへの割り込みはありません。

冗長スーパーバイザ エンジンでは、各リンクのハードウェア リンク ステータスを認識するため、スイッチオーバーになる前にアクティブだったポートはアップリンク ポートを含め、アクティブのままです。ただし、アップリンク ポートは物理的にスーパーバイザ エンジン上にあるので、スーパーバイザ エンジンが取り外されると切断されます。

アクティブ スーパーバイザ エンジンに障害が発生した場合、冗長スーパーバイザ エンジンがアクティブになります。この新しいアクティブ スーパーバイザ エンジンには既存のレイヤ 2 スイッチング情報を使用して、トラフィック転送を続けます。ルーティング テーブルが新しいアクティブ スーパーバイザ エンジンに追加されるまで、レイヤ 3 の転送は延期されます。

SSO は、次のレイヤ 2 機能のステートフル スイッチオーバーをサポートします。次の機能のステートは、アクティブおよび冗長スーパーバイザ エンジン間で保存されます。

- 802.3
- 802.3u
- 802.3x (フロー制御)
- 802.3ab (GE)
- 802.3z (CWDM を含めたギガビット イーサネット)
- 802.3ad (LACP)
- 802.1p (レイヤ 2 QoS)
- 802.1q
- 802.1X (認証)
- 802.1D (スパンニングツリー プロトコル)
- 802.3af (インライン パワー)
- PAgP
- VTP
- ダイナミック ARP インスペクション
- DHCP スヌーピング
- IP ソース ガード
- IGMP スヌーピング (バージョン 1 および 2)
- DTP (802.1Q および ISL)
- MST
- PVST+
- Rapid PVST
- PortFast/UplinkFast/BackboneFast
- BPDU ガードおよびフィルタリング
- 音声 VLAN
- ポートセキュリティ
- ユニキャスト MAC フィルタリング
- ACL (VACL、PAACL、RACLS)
- QoS (DBL)
- マルチキャスト ストーム制御/ブロードキャスト ストーム制御

SSO は、次の機能と互換性があります。ただし、次の機能のプロトコル データベースは冗長スーパーバイザ エンジンとアクティブ スーパーバイザ エンジン間では同期化されません。

- Layer 2 Protocol Tunneling (L2PT; レイヤ 2 プロトコル トンネリング) を備えた 802.1Q トンネリング

- ベビー ジャイアント
- ジャンボ フレーム サポート
- CDP
- フラッディング ブロック
- UDLD
- SPAN/RSPAN
- NetFlow

SSO 機能がイネーブルの場合、次の機能が冗長スーパーバイザ エンジンで学習されます。

- Catalyst 4500 シリーズ スイッチ上のレイヤ 3 プロトコルすべて (スイッチ仮想インターフェイス (SVI))

## スーパーバイザ エンジンの冗長構成の同期化の概要

通常の動作中、永続的なコンフィギュレーション (RPR および SSO) と実行コンフィギュレーション (SSO だけ) は、2 台のスーパーバイザ エンジンの間のデフォルトで同期化されます。スイッチオーバー時には、新しいアクティブ スーパーバイザ エンジンが現在の設定を使用します。



(注) 冗長スーパーバイザ エンジン コンソールに CLI (コマンドライン インターフェイス) コマンドを入力することはできません。

ここでは、スーパーバイザ エンジンの冗長構成の同期化について説明します。

- 「RPR スーパーバイザ エンジンの設定の同期化」 (P.8-5)
- 「SSO スーパーバイザ エンジンの設定の同期化」 (P.8-6)

## RPR スーパーバイザ エンジンの設定の同期化

RPR モードでは、冗長スーパーバイザ エンジンが部分的に初期化されるだけであるため、起動時に設定変更を受信する場合、および設定変更を保存する場合にかぎり、アクティブ スーパーバイザ エンジンと対話します。

冗長スーパーバイザ エンジンが RPR モードで稼働している場合は、次のイベントがトリガーとなって設定情報の同期化が発生します。

- 冗長スーパーバイザ エンジンが起動したとき、**auto-sync** コマンドによって、固定コンフィギュレーションに同期化されます。このコマンドは、デフォルトでイネーブルになっています。詳細については、「スーパーバイザ エンジンの設定の同期化」 (P.8-12) を参照してください。
- アクティブ スーパーバイザ エンジンが冗長スーパーバイザ エンジンを検出すると、設定情報がアクティブ スーパーバイザ エンジンから冗長スーパーバイザ エンジンに同期化されます。この同期化により、冗長スーパーバイザ エンジン上にある既存のスタートアップ コンフィギュレーション ファイルが上書きされます。
- 設定を変更する場合は、**write** コマンドを使用して、冗長スーパーバイザ エンジンのスタートアップ コンフィギュレーションを保存および同期化する必要があります。

## SSO スーパーバイザ エンジンの設定の同期化



(注)

ステートフル スイッチオーバーは、Supervisor Engine 6-E ではサポート されていません。

冗長スーパーバイザ エンジンが SSO モードで稼働している場合は、次のイベントがトリガーとなって設定情報の同期化が発生します。

- アクティブ スーパーバイザ エンジンが冗長スーパーバイザ エンジンを検出すると、固定および実行コンフィギュレーションの同期化が発生し、これにより、冗長スーパーバイザ エンジンが完全初期化ステートに移行できるようになります。
- リアルタイムで変更が発生すると、アクティブ スーパーバイザ エンジンは、必要に応じて、実行コンフィギュレーションおよび（または）固定コンフィギュレーションと冗長スーパーバイザ エンジンを同期化します。
- 設定を変更する場合、**write** コマンドを使用して、アクティブ スーパーバイザ エンジンが冗長スーパーバイザ エンジンのスタートアップ コンフィギュレーションを保存および同期化できるようにする必要があります。

## スーパーバイザ エンジンの冗長構成に関する注意事項および制約事項

スーパーバイザ エンジンの冗長構成に関する注意事項および制約事項は、次のとおりです。

- スロット 10 で WS-X4302-GB なしでギガ ビット イーサネット および 10 ギガビット イーサネット アップリンクの両方を使用するように Supervisor Engine V-10GE を設定する場合は、モジュール 10 はディセーブルになり、ギガ ビット イーサネット ポートを使用して設定をロールバックできません。

回復するには、次のコマンドを入力します。

```
config t
hw-module uplink select tengigabitethernet // This sets the switch back to default
mode
!
ctrl z
wr me
redundancy reload shelf // The switch will reload with all 10 modules working ok

//The switch reloads
```

```
config t
hw-module uplink select gigabitethernet // This sets the switch to the desired link
!
ctrl z
wr me
redundancy reload shelf // The switch reloads with module 10 active with the gigabit
ethernet port(s) ON and the ten gigabit ethernet port(s) Off
```

- RPR では Cisco IOS Release 12.1(12c)EW、リリース 12.1(19)E 以降のリリースが必要です。SSO では Cisco IOS Release 12.2(20)EWA 以降のリリースが必要です。
- Catalyst 4507R スイッチおよび 4510R スイッチは、スーパーバイザ エンジンの冗長構成に対応する唯一の Catalyst 4500 シリーズ スイッチです。

- Catalyst 4510R シリーズ スイッチでは、WS-X4516 および WS-X4516-10GE スーパーバイザ エンジンだけがサポートされています。Catalyst 4507R シリーズ スイッチでは、WS-X4013+、WS-X4013+10GE、WS-X4515、WS-X4516、および WS-X4516-10GE スーパーバイザ エンジンがサポートされています。
- Catalyst 4507R シリーズ スイッチの Cisco IOS Release 12.2(25)SG 以降のリリースでは、10-GigabitEthernet および GigabitEthernet アップリンクは、Supervisor Engine V-10GE (WS-X4516-10GE) と Supervisor Engine II+10GE (WS-4013+10GE) で同時に使用できます。Cisco IOS リリース 12.2(25)SG よりも前のリリースでは、10-GigabitEthernet アップリンクまたは GigabitEthernet アップリンクを選択するには **hw-module uplink select** コンフィギュレーション コマンドを使用する必要があります。
- Cisco IOS Release 12.2(25)SG 以降のリリースでは、Supervisor Engine V-10GE (WS-X4516-10GE) を Catalyst 4510R シリーズ スイッチ上で使用する場合、10-GigabitEthernet アップリンクと GigabitEthernet アップリンクの両方を同時に使用できます。ただし、この場合は、スロット 10 に WS-X4302-GB が搭載されている必要があります。10-GigabitEthernet アップリンクまたは GigabitEthernet アップリンクのいずれかを選択する場合は、スロット 10 に任意のラインカードを搭載できます。アップリンクを選択する場合は、**hw-module uplink select** コンフィギュレーション コマンドを使用します。Cisco IOS リリース 12.2(25)SG よりも前のリリースでは、10-GigabitEthernet アップリンクと GigabitEthernet アップリンクの両方を同時に使用することはできません。
- RPR または SSO モードで、WS-X4516-10GE および WS-X4013+10GE Supervisor Engines で 10-GigabitEthernet アップリンクを選択する場合、インターフェイス 10-GigabitEthernet 1/1 および 2/1 だけを使用できます。同様に、GigabitEthernet アップリンクを選択する場合は、インターフェイス GigabitEthernet 1/3、1/4、2/3、および 2/4 だけを使用できます。両方のアップリンクを同時に選択すると、インターフェイス TenGigabitEthernet 1/1 および 2/1 と、インターフェイス GigabitEthernet 1/3、1/4、2/3、および 2/4 を使用できます。
- 冗長構成では、同じスーパーバイザ エンジン モデル (同じモデル、メモリ、NFL ドータカードなど) で同じ Cisco IOS ソフトウェア イメージを使用する、シャーシに搭載されているスーパーバイザ エンジンが必要です。
- WS-X4013+ および WS-X4515 スーパーバイザ エンジンで RPR または SSO モードを使用する場合に使用できるのは、GigabitEthernet インターフェイスの Gig1/1 および Gig2/1 だけです。Gig1/2 および Gig2/2 アップリンク ポートは使用できません。
- WS-X4516 アクティブおよび冗長スーパーバイザ エンジンが同じシャーシに搭載されている場合、アップリンク ポート (Gig1/1、Gig2/1、Gig1/2、Gig2/2) を使用できます。
- シャーシのアクティブ スーパーバイザ エンジンおよび冗長スーパーバイザ エンジンは、スロット 1 およびスロット 2 に搭載する必要があります。
- シャーシの各スーパーバイザ エンジンには、スイッチを独自に動作させるために、独自のフラッシュ デバイスとコンソール ポート接続が備えられている必要があります。
- 各スーパーバイザ エンジンには、個別のコンソール ポート接続を行う必要があります。コンソール ポートに Y 字ケーブルを接続しないでください。
- スーパーバイザ エンジンの冗長構成にはスーパーバイザ エンジンのロード バランシング機能がありません。
- スイッチオーバー時に Cisco Express Forwarding (CEF; シスコ エクスプレス フォワーディング) テーブルがクリアされます。その結果、ルート テーブルの再コンバージェンスが行われるまで、ルーティング対象トラフィックは中断されます。SSO 機能がスーパーバイザ エンジンの冗長スイッチオーバー時間を 30 秒以上から 1 秒以下に削減するので、この再コンバージェンス時間は最小となります。また、スイッチが SSO 用に設定された場合のレイヤ 3 のフェールオーバー時間も早くなります。

- スタティック IP ルートはコンフィギュレーション ファイル内のエントリから設定されるため、スイッチオーバー中も維持されます。
- アクティブ スーパーバイザ エンジン上で維持されているレイヤ 3 ダイナミック ステート情報は、冗長スーパーバイザ エンジンへの同期化が行われず、スイッチオーバー時に失われます。
- Cisco IOS Release 12.2 以降、サポートされていない状態が検出された場合（アクティブ スーパーバイザ エンジンで Cisco IOS Release 12.2(20)EW が稼働している場合など）、冗長スーパーバイザ エンジンで Cisco IOS Release 12.1(20)EW が稼働している場合など）、冗長スーパーバイザ エンジンが複数回リセットされ、ROMmon モードになります。したがって、「ソフトウェア アップグレードの実行」(P.8-14) に示す正しい手順に従ってください。
- Cisco IOS Release 12.2(20)EWA または Cisco IOS Release 12.2(25)EW を実行（またはアップグレード）して、冗長シャーシ（Catalyst 4507R または Catalyst 4510R シリーズ スイッチ）で単一のスーパーバイザ エンジンを使用し、ルーテッド ポートを使用する場合は、次のいずれかを実行します。
  - ルーテッド ポートの代わりに、SVI を使用します。
  - 冗長モードを SSO から RPR に変更します。
- SSO モードでの SNMP 同期および SNMP 設定操作による冗長スーパーバイザ エンジンの設定変更は、冗長スーパーバイザ エンジンと同期化されません。SSO モードで SNMP 設定操作を実行することはできますが、予期しない動作が発生する場合があります。  
SSO モードで SNMP によってスイッチを設定したあとで、アクティブ スーパーバイザ エンジン上の `running-config` ファイルを `startup-config` ファイルにコピーすると、冗長スーパーバイザ エンジン上の `startup-config` ファイルの同期化が発生します。新しい設定が冗長スーパーバイザ エンジンに適用されるように冗長スーパーバイザ エンジンをリロードします。
- スタートアップ（一括）同期中は、設定を変更できません。このプロセス中に設定を変更しようとすると、次のメッセージが生成されます。  
`Config mode locked out till standby initializes`
- スーパーバイザ エンジンのスイッチオーバー時に設定を変更した場合、その変更内容は失われます。

## スーパーバイザ エンジンの冗長構成の設定

ここでは、スーパーバイザ エンジンの冗長構成を設定する手順について説明します。

- 「冗長構成の設定」(P.8-9)
- 「スタンバイ スーパーバイザ エンジンの仮想コンソール」(P.8-10)
- 「スーパーバイザ エンジンの設定の同期化」(P.8-12)



## 冗長構成の設定

冗長構成を設定する手順は、次のとおりです。

	コマンド	目的
ステップ1	Switch(config)# <b>redundancy</b>	冗長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(config-red)# <b>mode {sso   rpr}</b>	SSO または RPR を設定します。このコマンドを入力すると、冗長スーパーバイザ エンジンがリロードされ、SSO または RPR モードで動作を開始します。
ステップ3	Switch# <b>show running-config</b>	SSO または RPR がイネーブルであることを確認します。
ステップ4	Switch# <b>show redundancy</b> [clients   counters   history   states]	アクティブおよび冗長スーパーバイザ エンジン用に冗長構成情報（カウンタ、ステートなど）を表示します。

冗長構成を設定する場合は、次のことに注意してください。

- **sso** キーワードは、Cisco IOS Release 12.2(20)EWA 以降のリリースでサポートされています。
- **rpr** キーワードは、Cisco IOS Release 12.1(12c)EW 以降のリリースでサポートされます。

次に、SSO のシステムを設定し、冗長ファシリティ情報を表示する例を示します。

```
Switch> enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# redundancy
Switch(config-red)# mode sso
Switch(config-red)# end
Switch# show redundancy
Redundant System Information :
-----
      Available system uptime = 2 days, 2 hours, 39 minutes
Switchovers system experienced = 0
      Standby failures = 0
      Last switchover reason = none

      Hardware Mode = Duplex
Configured Redundancy Mode = Stateful Switchover
Operating Redundancy Mode = Stateful Switchover
Maintenance Mode = Disabled
Communications = Up

Current Processor Information :
-----
      Active Location = slot 1
      Current Software state = ACTIVE
      Uptime in current state = 2 days, 2 hours, 39 minutes
      Image Version = Cisco Internetwork Operating System Software
IOS (tm) Catalyst 4000 L3 Switch Software (cat4000-I5S-M), Version 12.2(20)EWA(3
.92), CISCO INTERNAL USE ONLY ENHANCED PRODUCTION VERSION
Copyright (c) 1986-2004 by cisco Systems, Inc.
Compiled Wed 14-Jul-04 04:42 by esi
      BOOT = bootflash:cat4000-i5s-mz.122_20_EWA_392,1
      Configuration register = 0x2002

Peer Processor Information :
-----
      Standby Location = slot 2
      Current Software state = STANDBY HOT
      Uptime in current state = 2 days, 2 hours, 39 minutes
```

```

Image Version = Cisco Internetwork Operating System Software
IOS (tm) Catalyst 4000 L3 Switch Software (cat4000-I5S-M), Version 12.2(20)EWA(3
.92), CISCO INTERNAL USE ONLY ENHANCED PRODUCTION VERSION
Copyright (c) 1986-2004 by cisco Systems, Inc.
Compiled Wed 14-Jul-04 0
          BOOT = bootflash:cat4000-i5s-mz.122_20_EWA_392,1
Configuration register = 0x2002

```

Switch#

次に、冗長ファシリティ ステート情報を表示する例を示します。

```

Switch# show redundancy states
my state = 13 -ACTIVE
    peer state = 8 -STANDBY HOT
        Mode = Duplex
        Unit = Primary
        Unit ID = 2

Redundancy Mode (Operational) = Stateful Switchover
Redundancy Mode (Configured) = Stateful Switchover
    Split Mode = Disabled
    Manual Swact = Enabled
    Communications = Up

    client count = 21
    client_notification_TMR = 240000 milliseconds
        keep_alive TMR = 9000 milliseconds
        keep_alive count = 0
        keep_alive threshold = 18
        RF debug mask = 0x0

```

Switch#

次に、システム設定を RPR モードから SSO モードに変更する例を示します。

```

Switch(config)# redundancy
Switch(config-red)# mode
Switch(config-red)# mode sso
Changing to sso mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
Switch#
*Aug  1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug  1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost

```

次に、システム設定を SSO モードから RPR モードに変更する例を示します。

```

Switch(config)# redundancy
Switch(config-red)# mode rpr
Changing to rpr mode will reset the standby. Do you want to continue?[confirm]
Switch(config-red)# end
*Aug  1 13:11:16: %C4K_REDUNDANCY-3-COMMUNICATION: Communication with the peer Supervisor
has been lost
*Aug  1 13:11:16: %C4K_REDUNDANCY-3-SIMPLEX_MODE: The peer Supervisor has been lost

```

## スタンバイ スーパーバイザ エンジンの仮想コンソール

Catalyst 4500 シリーズ スイッチには、冗長性を持たせるため、2 つのスーパーバイザ エンジンを搭載できます。スイッチに電源が入ると、スーパーバイザ エンジンの 1 つがアクティブになり、スイッチ オーバーが発生するまでアクティブのままになります。もう 1 つのスーパーバイザ エンジンはスタンバイ モードのままです。

スーパーバイザ エンジンのそれぞれには、自身のコンソール ポートがあります。スタンバイ スーパーバイザ エンジンのコンソール ポート経由でだけ、スタンバイ スーパーバイザ エンジンにアクセスできます。したがって、スタンバイ スーパーバイザ に対するアクセス、モニタリング、またはデバッグを行うには、スタンバイ コンソールに接続する必要があります。

スタンバイ スーパーバイザ エンジンの仮想コンソールを使用すると、スタンバイ コンソールへの物理的な接続がなくてもアクティブ スーパーバイザ エンジンからスタンバイ コンソールにアクセスできます。EOBC で IPC を使用してスタンバイ スーパーバイザ エンジンと通信し、アクティブ スーパーバイザ エンジン上でスタンバイ コンソールをエミュレートします。一度にアクティブにできるアクティブ スタンバイ コンソール セッションは 1 つだけです。

スタンバイ スーパーバイザ エンジンの仮想コンソールにより、アクティブ スーパーバイザ エンジンにログインしているユーザは、スタンバイ スーパーバイザ エンジン上で `show` コマンドをリモートで実行し、アクティブ スーパーバイザ エンジンでその結果を表示できます。仮想コンソールは、アクティブ スーパーバイザ エンジンからだけ利用できます。

アクティブ スーパーバイザ エンジンからアクティブ スーパーバイザ エンジンの `attach module` コマンド、`session module` コマンド、または `remote login` コマンドを使用してスタンバイ仮想コンソールにアクセスできます。これらのコマンドを実行してスタンバイ コンソールにアクセスするには、特権 EXEC モード (レベル 15) を開始している必要があります。

スタンバイ仮想コンソールにアクセスすると、端末プロンプトは自動的に [`<hostname>-standby-console#`] に変わります。`hostname` はスイッチに設定した名前です。仮想コンソールを終了すると、このプロンプトは元のプロンプトに戻ります。

`exit` コマンドまたは `quit` コマンドを入力すると、仮想コンソールは終了します。ログインしたアクティブ スーパーバイザ エンジンの端末の無活動時間が設定されたアイドル時間を超えると、アクティブ スーパーバイザ エンジンの端末から自動的にログアウトします。この場合、仮想コンソールセッションも終了します。また、スタンバイが再起動すると、仮想コンソールセッションも自動的に終了します。スタンバイが起動したあとは、別の仮想コンソールセッションを作成する必要があります。

仮想コンソールを使用してスタンバイ スーパーバイザ エンジンにログインするには、次のようにします。

```
Switch# session module 2
Connecting to standby virtual console
Type "exit" or "quit" to end this session

Switch-standby-console# exit
Switch#
```

スタンバイ コンソールがイネーブルにされていない場合は、次のメッセージが表示されます。

```
Switch-standby-console#
Standby console disabled.
Valid commands are: exit, logout
```



(注)

スタンバイ仮想コンソールには、コマンド履歴、コマンド補完、コマンドヘルプ、部分コマンドキーワードなど、スーパーバイザ コンソールから利用できる標準的な機能が備わっています。

次の制限事項がスタンバイ仮想コンソールに適用されます。

- 仮想コンソールで実行されたコマンドは、すべて最後まで実行されます。`auto-more` 機能はありません。したがって、`terminal length 0` コマンドの実行時と同じように機能します。また、対話形式ではありません。したがって、アクティブ スーパーバイザ エンジン上でキー シーケンスを入力しても、コマンドの実行を中断できません。このため、コマンドの出力量が多い場合、仮想コンソールはこの出力をスーパーバイザの画面上に表示します。

## ■ スーパーバイザ エンジンの冗長構成の設定

- 仮想コンソールは対話形式ではありません。仮想コンソールはコマンドのインタラクティブ性を検出しないため、ユーザとの対話を必要とするコマンドが入力されると、RPC タイマーがコマンドを中断するまで仮想コンソールは待機します。

仮想コンソール タイマーは 60 秒に設定されています。60 秒後に仮想コンソールはプロンプトに戻ります。この間、キーボードからコマンドを中断できません。操作を続ける前に、タイマーが期限切れになるのを待つ必要があります。

- 仮想コンソールを使用して、スタンバイ スーパーバイザ エンジン上で表示されているデバッグおよび Syslog メッセージを表示することはできません。仮想コンソールは、仮想コンソールから実行されたコマンドの出力だけを表示します。実際のスタンバイ コンソールで表示される別の情報は、仮想コンソールでは表示できません。

## スーパーバイザ エンジンの設定の同期化

2 台のスーパーバイザ エンジンが使用する設定を手動で同期化するには、アクティブ スーパーバイザ エンジン上で次の作業を行います。

	コマンド	目的
ステップ 1	Switch(config)# <b>redundancy</b>	冗長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(config-red)# <b>main-cpu</b>	main-cpu コンフィギュレーション サブモードを開始します。
ステップ 3	Switch(config-r-mc)# <b>auto-sync</b> { <b>startup-config</b>   <b>config-register</b>   <b>bootvar</b>   <b>standard</b> }	設定要素を同期化します。
ステップ 4	Switch(config-r-mc)# <b>end</b>	特権 EXEC モードに戻ります。
ステップ 5	Switch# <b>copy running-config startup-config</b>	Dynamic Random-Access Memory (DRAM; ダイナミック ランダム メモリ) の実行コンフィギュレーション ファイルを NVRAM (不揮発性 RAM) のスタートアップ コンフィギュレーション ファイルに同期化します。  (注) DRAM 上の実行コンフィギュレーション ファイルを同期化する場合、このステップは不要です。



(注) SNMP によるアクティブ スーパーバイザ エンジンの設定変更は、冗長スーパーバイザ エンジンに同期化されません。この場合の詳細な取り扱いについては、「[スーパーバイザ エンジンの冗長構成に関する注意事項および制約事項](#)」(P.8-6) を参照してください。



(注) **auto-sync** コマンドは、**config-reg**、**bootvar**、および **startup/private** コンフィギュレーション ファイルの同期化だけを制御します。カレンダーおよび VLAN データベース ファイルは、変更するたびに常に同期化されます。SSO モードでは、**running-config** は常に同期化されます。

次に、**auto-sync standard** コマンドを使用して、デフォルトの自動同期化機能を再びイネーブルにし、アクティブ スーパーバイザ エンジンの **startup-config** および **config-register** 設定を冗長スーパーバイザ エンジンと同期化する例を示します。ブート変数のアップデートは自動的に行われるため、ディセーブルにできません。

```
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-sync standard
```

```
Switch(config-r-mc)# end
Switch# copy running-config startup-config
```



(注)

標準の自動同期対象の設定要素を個別に手動で同期化するには、デフォルトの自動同期化機能をディセーブルにします。



(注)

`auto-sync standard` を設定すると、個別の同期化オプション (`no auto-sync startup-config` など) は無視されます。

次に、デフォルトの自動同期化をディセーブルにして、アクティブ スーパーバイザ エンジンの `config-register` だけを冗長スーパーバイザ エンジンに自動的に同期化し、スタートアップ コンフィギュレーションの同期化を許可しない例を示します。

```
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# no auto-sync standard
Switch(config-r-mc)# auto-sync config-register
Switch(config-r-mc)# end
```

## 手動による切り替え

ここでは、テストのため手動による切り替え (アクティブ スーパーバイザ エンジンから冗長スーパーバイザ エンジンへ) を行う方法を説明します。ご使用の稼働環境に SSO を展開する前に、手動で切り替えることを推奨します。



(注)

これは、SSO が冗長モードとして設定されていることを前提としています。

手動で切り替えるには、アクティブ スーパーバイザ エンジンで次の作業を行います。

	コマンド	目的
ステップ1	Switch# <code>show redundancy</code>	ピア ステートが STANDBY HOT ステートであることを確認します。 P.8-10 の <code>show redundancy states</code> コマンドの例を参照してください。
ステップ2	Switch# <code>redundancy force-switchover</code>	アクティブ スーパーバイザ エンジンから冗長スーパーバイザ エンジンへのスイッチオーバーが開始されます。 冗長スーパーバイザ エンジンのステートが STANDBY HOT でない場合、このコマンドは実行されません。

次の使用上の注意事項に留意してください。

- 強制的にスイッチオーバーを実施するには、冗長スーパーバイザ エンジンを STANDBY HOT ステートにする必要があります。 `show redundancy` コマンドを使用すると、ステートを確認できます。ステートが STANDBY HOT でない場合、 `redundancy force-switchover` コマンドは実行されません。

## ■ ソフトウェア アップグレードの実行

- スイッチオーバーを開始するには、**reload** コマンドではなく、**redundancy force-switchover** コマンドを使用します。**redundancy force-switchover** コマンドが、冗長スーパーバイザ エンジンが正しいステートであるかどうかを最初に確認します。**reload** コマンドを使用してステータスが **STANDBY HOT** でない場合、**reload** コマンドは現在のスーパーバイザ エンジンだけをリセットします。

最初のスイッチオーバーのあと、シャーシのスロット 1 のスーパーバイザ エンジンをアクティブ スーパーバイザ エンジンにする必要が生じる場合があります。スーパーバイザ エンジン 1 上のイメージが、両方のスーパーバイザ エンジン上で実行するイメージである場合は、冗長構成にするためにスロット 1 のスーパーバイザ エンジン上のイメージを再起動する必要はありません。代わりに、別のスイッチオーバーを強制的に実行できます。ただし、両方のスーパーバイザ エンジン上で実行するイメージのバージョンを新しくする場合、「ソフトウェア アップグレードの実行」(P.14) の手順に従ってください。どのスロットにアクティブ スーパーバイザ エンジンが含まれているかを判断し、必要に応じて別のスイッチオーバーを強制的に実行するには、**show module** コマンドを使用します。

## ソフトウェア アップグレードの実行

スーパーバイザ エンジンの冗長構成がサポートするソフトウェアのアップグレード手順によって、冗長スーパーバイザ エンジン上の Cisco IOS ソフトウェア イメージをリロードし、そのあとでもう一度、アクティブ スーパーバイザ エンジンにリロードできます。

スーパーバイザ エンジンの冗長構成によってサポートされているソフトウェアのアップグレード手順を使用すると、冗長スーパーバイザ エンジン上の Cisco IOS ソフトウェア イメージをリロードし、そのあとでもう一度、アクティブ スーパーバイザ エンジンにリロードできます。

次のシナリオはサポートされません。アクティブ スーパーバイザ エンジンで Cisco IOS Release 12.1(x)E が稼働しており、スタンバイ スーパーバイザ で Cisco IOS Release 12.2(x)S が稼働している場合。スタンバイ スーパーバイザ が繰り返しリセットされます。

冗長スーパーバイザ を Cisco IOS Release 12.1(x)E から 12.2(x)S にアップグレードする場合は、システムを完全に再起動することが必要となります。

ソフトウェアのアップグレードを実行するには、次の作業を行います。

	コマンド	目的
ステップ 1	Switch# <b>copy source_device:source_filename slot0:target_filename</b>  または:  Switch# <b>copy source_device:source_filename bootflash:target_filename</b>	スーパーバイザ エンジンのブートフラッシュに、新しい Cisco IOS ソフトウェア イメージをコピーします。
ステップ 2	Switch# <b>copy source_device:source_filename slaveslot0:target_filename</b>  または:  Switch# <b>copy source_device:source_filename slavebootflash:target_filename</b>	スレーブ デバイス (slavebootflash、slaveslot0 など) に、新しいイメージをコピーします。

コマンド	目的
<b>ステップ 3</b> Switch# <b>config terminal</b> Switch(config)# <b>config-register 0x2</b> Switch(config)# <b>boot system flash</b> device:file_name	新しいイメージを起動するように、スーパーバイザ エンジンを設定します。  システムが古いイメージを自動的に起動するよう設定されている場合、次のコマンドストリングを発行して代わりに新しいイメージを起動します。 <b>no boot system flash device:old_file_name</b>
<b>ステップ 4</b> Switch(config)# <b>redundancy</b>	冗長コンフィギュレーション モードを開始します。
<b>ステップ 5</b> Switch(config-red)# <b>main-cpu</b>	main-cpu コンフィギュレーション サブモードを開始します。
<b>ステップ 6</b> Switch(config-r-mc)# <b>auto-syn standard</b>	設定要素を同期化します。
<b>ステップ 7</b> Switch(config-r-mc)# <b>end</b>	特権 EXEC モードに戻ります。
<b>ステップ 8</b> Switch# <b>copy running-config start-config</b>	設定を保存します。
<b>ステップ 9</b> Switch# <b>redundancy reload peer</b>	冗長スーパーバイザ エンジンを読み込み、オンラインに戻します (新しいリリースの Cisco IOS ソフトウェアを使用します)。  <b>(注)</b> ステップ 10 に進む前に、スイッチが RPR モードで動作していることを確認します。
<b>ステップ 10</b> Switch# <b>redundancy force-switchover</b>	冗長スーパーバイザ エンジンへのスイッチオーバーを手動で実行します。冗長スーパーバイザ エンジンが、新しい Cisco IOS ソフトウェア イメージを使用するアクティブ スーパーバイザ エンジンになります。  それまでアクティブだったスーパーバイザ エンジンが新しいイメージで再起動され、冗長スーパーバイザ エンジンになります。

次に、ソフトウェア アップグレードの実行例を示します。

```
Switch# config terminal
Switch(config)# config-register 0x2
Switch(config)# boot system flash slot0:cat4000-i5s-mz.122-20.EWA
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-syn standard
Switch(config-r-mc)# end
Switch# copy running-config start-config
Switch# redundancy reload peer
Switch# redundancy force-switchover
Switch#
```

次に、アクティブ スーパーバイザ エンジン上の実行コンフィギュレーションと冗長スーパーバイザ エンジンとの同期化が成功したことを確認する例を示します。

```
Switch# config terminal
Switch(config)# redundancy
Switch(config-red)# main-cpu
Switch(config-r-mc)# auto-sync standard
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The bootvar has been successfully synchronized to the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The config-reg has been successfully synchronized to the standby supervisor
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The startup-config has been successfully synchronized to the standby supervisor
```

## ■ 冗長スーパーバイザ エンジンでの Bootflash 操作

```
4d01h: %C4K_REDUNDANCY-5-CONFIGSYNC: The private-config has been successfully synchronized
to the standby supervisor
```

上記の例では、アクティブ スーパーバイザ エンジンからのブート変数、config-register、スタートアップ コンフィギュレーションが冗長スーパーバイザ エンジンに同期化したことを示します。

## 冗長スーパーバイザ エンジンでの Bootflash 操作



(注) 冗長スーパーバイザ エンジン上のコンソール ポートは使用できません。

冗長スーパーバイザ エンジン bootflash を操作するには、次のうち 1 つまたは複数の作業を行います。

コマンド	目的
Switch# <b>dir slaveslot0:target_filename</b>  or: Switch# <b>dir slavebootflash:target_filename</b>	冗長スーパーバイザ エンジンの <b>slot0</b> : デバイスの内容を表示します。  冗長スーパーバイザ エンジンの <b>bootflash</b> : デバイスの内容を表示します。
Switch# <b>delete slaveslot0:target_filename</b>  or: Switch# <b>delete slave bootflash:target_filename</b>	冗長スーパーバイザ エンジンの <b>slot0</b> : デバイスから、特定のファイルを削除します。  冗長スーパーバイザ エンジンの <b>bootflash</b> : デバイスから、特定のファイルを削除します。
Switch# <b>squeeze slaveslot0:target_filename</b>  or: Switch# <b>squeeze slavebootflash:target_filename</b>	冗長スーパーバイザ エンジンの <b>slot0</b> : デバイスをスクイーズします。  冗長スーパーバイザ エンジンの <b>bootflash</b> : デバイスをスクイーズします。
Switch# <b>format slaveslot0:target_filename</b>  or: Switch# <b>format slavebootflash:target_filename</b>	冗長スーパーバイザ エンジンの <b>slot0</b> : デバイスをフォーマットします。  冗長スーパーバイザ エンジンの <b>bootflash</b> : デバイスをフォーマットします。
Switch# <b>copy source_device:source_filename slaveslot0:target_filename</b>  or: Switch# <b>copy source_device:source_filename slavebootflash:target_filename</b>	アクティブ スーパーバイザ エンジンから冗長スーパーバイザ エンジンの <b>slot0</b> : デバイスへファイルをコピーします。  冗長スーパーバイザ エンジンの <b>bootflash</b> : デバイスにファイルをコピーします。  (注) 送信元はアクティブ スーパーバイザ エンジン、または Trivial File Transfer Protocol (TFTP; 簡易ファイル転送プロトコル) サーバです。