



CHAPTER 5

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、スタンバイ 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 Stateful Switchover (SSO; ステートフル スイッチオーバー) 動作モードで稼動することで、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)

Supervisor Engine 7-E 上での SSO と RPR の実行に関する最小 ROMMON 要件は 15.0(1r)SG1 です。

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について」 (P.5-2)
- 「スーパーバイザ エンジンの冗長構成の同期化」 (P.5-5)
- 「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.5-6)
- 「スーパーバイザ エンジンの冗長設定」 (P.5-7)
- 「手動による切り替え」 (P.5-12)
- 「ソフトウェア アップグレードの実行」 (P.5-13)
- 「スタンバイ スーパーバイザ エンジン上でのブートフラッシュの操作」 (P.5-14)



(注)

この章で使用するスイッチ コマンドの構文および使用方法の詳細については、次の URL で『Cisco Catalyst 4500 Series Switch Command Reference』と関連資料を参照してください。

<http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html>

Catalyst 4500 のコマンド リファレンスに掲載されていないコマンドについては、より詳細な Cisco IOS ライブラリを参照してください。次の URL で『Catalyst 4500 Series Switch Cisco IOS Command Reference』と関連資料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について

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.5-2)
- 「RPR 動作」 (P.5-3)
- 「SSO 動作」 (P.5-3)

概要

スーパーバイザ エンジンの冗長構成がイネーブルになっていれば、アクティブ スーパーバイザ エンジンが故障した場合、または、手動スイッチオーバーが実行された場合に、スタンバイ 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。スタンバイ 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で自動的に初期化されているため、スイッチオーバー時間が短縮されます (コンフィギュレーションによっては RPR モードで 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。

- スーパーバイザ エンジンの Online Insertion and Removal (OIR; 活性挿抜)
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.5-13) を参照)
スーパーバイザ エンジン上で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、スタンバイ 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 5-1 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 5-1 シャーシおよびスーパーバイザのサポート

サポート対象のスイッチ	サポート対象のスーパーバイザ エンジン
WS-C4507R-E	Supervisor Engine 7-E (WS-X45-SUP7-E)
WS-C4510R-E	Supervisor Engine 7-E (WS-X45-SUP7-E)
WS-C4507R+E	Supervisor Engine 7-E (WS-X45-SUP7-E)
WS-C4510R+E	Supervisor Engine 7-E (WS-X45-SUP7-E)

RPR 動作

RPR は、Cisco IOS-XE Release 3.1.0 SG 以降のリリースでサポートされます。スタンバイ スーパーバイザ エンジンが RPR モードで動作している場合は、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 エンジンの永続的なコンフィギュレーションと同期されます。



(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`、Virtual LAN (VLAN; 仮想 LAN) データベースが含まれます。

スタンバイ スーパーバイザ エンジンは、基本的なシステム初期化後に起動シーケンスを一旦停止し、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に、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。スイッチオーバー時は、スタンバイ スーパーバイザ エンジンが、初期化を完了すると、直接モジュールからハードウェア情報を読み取って、アクティブ スーパーバイザ エンジンになります。

SSO 動作

SSO は、Cisco IOS-XE Release 3.1.0 SG 以降のリリースでサポートされます。スタンバイ スーパーバイザ エンジンが SSO モードで動作している場合は、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 エンジンの永続的なコンフィギュレーションと実行コンフィギュレーションに同期されます。冗長スーパーバイザ エンジンはそのあと、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。

スタンバイ スーパーバイザ エンジンは、すべてのリンクのハードウェア リンク ステータスを認識しているため、スイッチオーバー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、アクティブのままになります。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は、スタンバイ 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 エンジンは既存のレイヤ 2 スwitch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。



(注) IOS-XE ソフトウェアが LAN Base モードで動作している場合は、SSO がサポートされません。

次の機能のステートは、アクティブ スーパーバイザ エンジンとスタンバイ 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
- Voice 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、PACL、RACL)
- 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 データベースはスタンバイ スーパーバイザ エンジンとアクティブ スーパーバイザ エンジン間で同期されません。

- 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
- フラッドイング ブロック
- UDLD
- SPAN/RSPAN
- NetFlow

SSO 機能がイネーブルの場合は、次の機能がスタンバイ 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 (Switch Virtual Interface (SVI; スイッチ仮想インターフェイス))

スーパーバイザ エンジンの冗長構成の同期化

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) スタンバイ スーパーバイザ エンジン コンソールで Command-Line Interface (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.5-5)
- 「[SSO スーパーバイザ エンジンの設定の同期化](#)」 (P.5-6)

RPR スーパーバイザ エンジンの設定の同期化

スタンバイ スーパーバイザ エンジンには RPR モードでは部分的にしか初期化されないため、起動時にコンフィギュレーション変更を受信し、コンフィギュレーション変更を保存する場合にだけ、アクティブ スーパーバイザ エンジンと対話します。

スタンバイ スーパーバイザ エンジンが RPR モードで動作している場合は、次のイベントによってコンフィギュレーション情報の同期化がトリガーされます。

- スタンバイ スーパーバイザ エンジンの起動時に、**auto-sync** コマンドが永続的なコンフィギュレーションを同期させます。このコマンドは、デフォルトでイネーブルに設定されています。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.5-11) を参照してください。
- アクティブ スーパーバイザ エンジンがスタンバイ スーパーバイザ エンジンを検出すると、コンフィギュレーション情報がアクティブ スーパーバイザ エンジンからスタンバイ スーパーバイザ エンジンに同期されます。この同期化によって、スタンバイ スーパーバイザ エンジン上のすべてのスタートアップ コンフィギュレーション ファイルが上書きされます。
- コンフィギュレーションを変更した場合は、**write** コマンドを使用して、スタンバイ スーパーバイザ エンジンにスタートアップ コンフィギュレーションを保存して同期させる必要があります。

SSO スーパーバイザ エンジンの設定の同期化

スタンバイ スーパーバイザ エンジンが SSO モードで動作している場合は、次のイベントによってコンフィギュレーション情報の同期化がトリガーされます。

- アクティブ スーパーバイザ エンジンがスタンバイ スーパーバイザ エンジンを検出すると、永続的なコンフィギュレーションと実行コンフィギュレーションの同期化が実施されるため、スタンバイ スーパーバイザ エンジンが完全初期化ステートに移行することができます。
- リアルタイムで変更が発生した場合は、アクティブ スーパーバイザ エンジンが、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）永続的なコンフィギュレーションとスタンバイ スーパーバイザ エンジンを同期させます。
- コンフィギュレーションを変更した場合は、**write** コマンドを使用して、アクティブ スーパーバイザ エンジンを通してスタンバイ スーパーバイザ エンジンにスタートアップ コンフィギュレーションを保存して同期させる必要があります。

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。

- RPR と SSO には Cisco IOS-XE Release 3.1.0 SG 以降のリリースが必要です。
- IOS-XE ソフトウェアが LAN Base モードで動作している場合は、SSO がサポートされません。
- WS-C4507R-E、WS-C4510R-E、WS-C4507R+E、および WS-C4510R+E は、Supervisor Engine 7-E 冗長性をサポートする唯一の Catalyst 4500 シリーズ スイッチです。
- SSO には、シャーシ内の両方のスーパーバイザ エンジンが、同じコンポーネント（モデルとメモリ）を備え、同じ Cisco IOS XE ソフトウェア イメージを使用する必要があります。
- WS-X45-SUP7-E を RPR または SSO モードで使用している場合は、それぞれのスーパーバイザ エンジン上で最初のアップリンクだけが使用できます。2 つ目のアップリンクは使用できません。
- アクティブ スーパーバイザ エンジンとスタンバイ スーパーバイザ エンジンは、7 スロット シャーシの場合はスロット 3 と 4 に、10 スロット シャーシの場合はスロット 5 と 6 に実装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、スイッチを稼働させるための独自のフラッシュ デバイスとコンソール ポート接続を備え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 字型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。したがって、ルート テーブルの再コンバージェンスまでの間、トラフィックのルーティングが中断されます。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 ファイルのエントリに基づいて設定されるので、スイッチオーバー後も維持されます。

- 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、スタンバイ スーパーバイザ エンジンに同期されないため、スイッチオーバー時に失われます。
- SNMP 設定操作を通して冗長スイッチ上のコンフィギュレーションが変更された場合は、SSO モードであっても、その変更がスタンバイ スーパーバイザ エンジンに同期されません。予期せぬ動作が発生する可能性があります。
- SSO モードで SNMP を介してスイッチを設定したら、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` ファイルにコピーして、スタンバイ スーパーバイザ エンジンへの `startup-config` ファイルの同期化をトリガーします。その後で、新しいコンフィギュレーションがスタンバイ スーパーバイザ エンジンに適用されるように、スタンバイ 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
```

- 設定変更がスーパーバイザ エンジンのスイッチオーバーと同時に発生した場合、それらの設定変更は失われます。
- ラインカードを冗長スイッチから取り外して SSO スイッチオーバーを開始し、ラインカードを再び挿入すると、すべてのインターフェイスがシャットダウンされます。ラインカードの元の設定の残りの部分は維持されます。

この状況が発生するのは、ラインカードを取り外す前にスイッチが SSO に到達した場合だけです。

スーパーバイザ エンジンの冗長設定

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.5-7)
- 「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.5-9)
- 「スーパーバイザ エンジンの設定の同期化」(P.5-11)

冗長構成の設定



(注)

IOS XE ソフトウェアは、スーパーバイザ エンジン上で使用可能なライセンスに基づく 3 つの異なるレベル（Enterprise Services、IP Base、および LAN Base）で起動することができます。LAN Base モードでイメージを起動している場合は、RPR 冗長モードのみが使用できます。

冗長構成を設定するには、次の作業を行います。

	コマンド	目的
ステップ1	Switch(config)# redundancy	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)# mode {sso rpr}	SSO または RPR を設定します。このコマンドを入力すると、スタンバイ スーパーバイザ エンジンがリロードされ、SSO または RPR モードで動作が開始されます。

	コマンド	目的
ステップ3	Switch# show running-config	SSO または RPR が設定されていることを確認します。
ステップ4	Switch# show redundancy [clients counters history states]	アクティブ スーパーバイザ エンジンとスタンバイ スーパーバイザ エンジンに関する冗長情報 (カウンタやステートなど) を表示します。

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 = 10 minutes Switchovers system experienced = 0
Standby failures = 1
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 3
Current Software state = ACTIVE
Uptime in current state = 9 minutes
Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500e-UNIVERSALK9-M), Version 15.0(100)XO(1.42), INTERIM SOFTWARE Copyright (c)
1986-2010 by Cisco Systems, Inc.
Compiled Sun 01-Aug-10 04:12 by gsbuprod
Configuration register = 0x920

Peer Processor Information :
-----
Standby Location = slot 4
Current Software state = STANDBY HOT
Uptime in current state = 0 minute
Image Version = Cisco IOS Software, Catalyst 4500 L3 Switch Software
(cat4500e-UNIVERSALK9-M), Version 15.0(100)XO(1.42), INTERIM SOFTWARE Copyright (c)
1986-2010 by Cisco Systems, Inc.
Compiled Sun 01-Aug-10 04:12 by gsbuprod
Configuration register = 0x920
```

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 = 3

Redundancy Mode (Operational) = Stateful Switchover
```



```
Redundancy Mode (Configured) = Stateful Switchover
Redundancy State = Stateful Switchover
Manual Swact = enabled
```

```
Communications = Up
```

```
client count = 64
client_notification_TMR = 240000 milliseconds
keep_alive TMR = 9000 milliseconds
keep_alive count = 1
keep_alive threshold = 18
RF debug mask = 0
```

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 エンジンにログオ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を開始すると、端末プロンプトが *hostname-standby-console#* に自動的に変更されます（ここで、*hostname* はスイッチに設定された名前です）。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、次の操作を実行します。

```
Switch# session module 4
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でない場合、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。コマンドによって大量の出力が発生した場合、仮想コンソールはスーパーバイザ エンジンの画面に出力を表示します。

- 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)# redundancy	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)# main-cpu	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)# auto-sync { startup-config config-register bootvar standard }	設定要素を同期化します。
ステップ 4	Switch(config-r-mc)# end	特権 EXEC モードに戻ります。
ステップ 5	Switch# copy running-config startup-config	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.5-6) 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** コンフィギュレーション をスタンバイ スーパーバイザ エンジンと同期させる例を示します。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` コンフィギュレーションのスタンバイ スーパーバイザ エンジンへの自動同期化を許可し、スタートアップ コンフィギュレーションの同期化を禁止する例を示します。

```
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、テスト目的の手動スイッチオーバー（アクティブ スーパーバイザ エンジンからスタンバイ スーパーバイザ エンジンへ）を実行する方法について説明します。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.6 ~ 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 エンジンからスタンバイ スーパーバイザ エンジンへのスイッチオーバーを開始します。

次の使用上の注意事項に留意してください。

- スイッチオーバーを強制します。冗長スーパーバイザ エンジンは、STANDBY HOT (SSO) ステートまたは STANDBY COLD (RPR) ステートにする必要があります。`show redundancy` コマンドを使用して、ステートを確認することができます。ステートが STANDBY HOT でも STANDBY COLD でもない場合は、`redundancy force-switchover` コマンドが実行されません。
- 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` コマンドを発行して、そのステータスが STANDBY HOT でも STANDBY COLD でもなかった場合は、`reload` コマンドによって現在のスーパーバイザ エンジンがリセットされ、ピア スーパーバイザは `terminals` ステート (STANDBY HOT または COLD) でなかったことによって処理を引き継ぐことができません。

通常のスイッチオーバー後に、シャーシ内のよりスロット番号の低いスーパーバイザ エンジンをアクティブ スーパーバイザ エンジンにすることができます。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

これは、IOS -XE ソフトウェアが LAN Base モードで動作している場合にのみ有効です。Enterprise Services モードまたは IP Base モードでは、ISSU を使用して RPR 冗長モードと SSO 冗長モードの両方のソフトウェアをアップグレードします。

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。

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# copy source_device:source_filename slot0:target_filename または Switch# copy source_device:source_filename bootflash:target_filename	スーパーバイザ エンジン上のブートフラッシュに、新しい Cisco IOS-XE ソフトウェア イメージをコピーします。
ステップ2	Switch# copy source_device:source_filename slaveslot0:target_filename または Switch# copy source_device:source_filename slavebootflash:target_filename	スタンバイ スーパーバイザ エンジン上のデバイス (slavebootflash や slaveslot0 など) に新しいイメージをコピーします。
ステップ3	Switch# config terminal Switch(config)# config-register 0x2 Switch(config)# boot system flash device:file_name	新しいイメージが起動されるように、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tring を発行して代わりに新しいイメージを起動します。 no boot system flash device:old_file_name
ステップ4	Switch(config)# redundancy	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5	Switch(config-red)# main-cpu	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6	Switch(config-r-mc)# auto-syn standard	設定要素を同期化します。
ステップ7	Switch(config-r-mc)# end	特権 EXEC モードに戻ります。
ステップ8	Switch# copy running-config start-config	設定を保存します。
ステップ9	Switch# redundancy reload peer	スタンバイ スーパーバイザ エンジンをリロードして、オンラインに戻します (Cisco IOS-XE ソフトウェアの新しいリリースを使用します)。
ステップ10	Switch# redundancy force-switchover	スタンバイ スーパーバイザ エンジンへの手動スイッチオーバーを実施します。スタンバイ スーパーバイザ エンジンが、新しい Cisco IOS-XE ソフトウェア イメージを使用する新しいアクティブ スーパーバイザ エンジンになります。 以前のアクティブ スーパーバイザ エンジンが新しいイメージで再起動し、スタンバイ スーパーバイザ エンジンになります。

■ スタンバイ スーパーバイザ エンジン上でのブートフラッシュの操作

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
bootflash0:cat4500e-universalk9.SSA.03.01.00.150.1.XO.bin
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
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。

スタンバイ スーパーバイザ エンジン上でのブートフラッシュの操作



(注) スタンバイ 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。

スタンバイ スーパーバイザ エンジン ブートフラッシュを操作するには、次の 1 つまたは複数の作業を行います。

コマンド	目的
Switch# <code>dir slaveslot0:target_filename</code> または Switch# <code>dir slavebootflash:target_filename</code>	スタンバイ スーパーバイザ エンジンの slot0 : デバイスの内容を表示します。 スタンバイ スーパーバイザ エンジンの bootflash : デバイスの内容を表示します。
Switch# <code>delete slaveslot0:target_filename</code> または Switch# <code>delete slave bootflash:target_filename</code>	スタンバイ スーパーバイザ エンジンの slot0 : デバイスから特定のファイルを削除します。 スタンバイ スーパーバイザ エンジンの bootflash : デバイスから特定のファイルを削除します。

コマンド	目的
Switch# squeeze slaveslot0:target_filename または Switch# squeeze slavebootflash:target_filename	スタンバイ スーパーバイザ エンジンの slot0: デバイスをスクイーズします。 スタンバイ スーパーバイザ エンジンの bootflash: デバイスをスクイーズします。
Switch# format slaveslot0:target_filename または Switch# format slavebootflash:target_filename	スタンバイ スーパーバイザ エンジンの slot0: デバイスをフォーマットします。 スタンバイ スーパーバイザ エンジンの bootflash: デバイスをフォーマットします。
Switch# copy source_device:source_filename slaveslot0:target_filename または Switch# copy source_device:source_filename slavebootflash:target_filename	アクティブ スーパーバイザ エンジンからスタンバイ スーパーバイザ エンジンの slot0: デバイスにファイルをコピーします。 スタンバイ スーパーバイザ エンジンの bootflash: 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。

■ スタンバイ スーパーバイザ エンジン上でのブートフラッシュの操作