



スイッチの起動設定の変更

この章では、Catalyst 6500 シリーズ スイッチ上で、BOOT 環境変数、CONFIG_FILE 環境変数、およびコンフィギュレーション レジスタを含むスイッチの起動設定を変更する方法について説明します。



(注)

この章で使用しているコマンドの完全な構文および使用方法の詳細については、『*Catalyst 6500 Series Switch Command Reference*』を参照してください。

この章で説明する内容は、次のとおりです。

- [スイッチの起動設定の機能 \(p.24-2\)](#)
- [スイッチのデフォルト起動設定 \(p.24-5\)](#)
- [コンフィギュレーション レジスタの設定 \(p.24-5\)](#)
- [BOOT 環境変数の設定 \(p.24-11\)](#)
- [CONFIG_FILE 環境変数の設定 \(p.24-12\)](#)
- [スイッチの起動設定の表示 \(p.24-13\)](#)

スイッチの起動設定の機能

ここでは、起動設定の機能について説明します。

- [ブートプロセスの概要 \(p.24-2\)](#)
- [ROM モニタの概要 \(p.24-2\)](#)
- [コンフィギュレーションレジスタの概要 \(p.24-3\)](#)
- [BOOT 環境変数の概要 \(p.24-3\)](#)
- [CONFIG_FILE 環境変数の概要 \(p.24-4\)](#)

ブート プロセスの概要

ブート プロセスには、ROM モニタとスーパーバイザ エンジン システム コードの 2 つのソフトウェア イメージが関与します。スイッチを起動またはリセットすると、ROM モニタ コードが実行されます。NVRAM (不揮発性 RAM) 内の設定に応じて、スイッチは ROM モニタ モードのままの場合と、スーパーバイザ エンジン システム コードをロードする場合とがあります。

スイッチを起動する方法は、コンフィギュレーション レジスタと BOOT 環境変数という、ユーザー側で設定できる 2 つのパラメータによって決まります。コンフィギュレーション レジスタについては、「[コンフィギュレーション レジスタの概要 \(p.24-3\)](#)」を参照してください。BOOT 環境変数については、「[BOOT 環境変数の概要 \(p.24-3\)](#)」を参照してください。

ROM モニタの概要

ROM モニタ コードは、スイッチの起動時、リセット時、または重大な例外が発生したときに実行されます。ROM モニタ モードが開始されるのは、スイッチが有効なシステム イメージを見つけることができなかった場合、NVRAM 内の設定が壊れていた場合、またはコンフィギュレーション レジスタが ROM モニタ モードを開始するように設定されていた場合です。ROM モニタ モードで、フラッシュ メモリ、ネットワーク サーバ ファイル、またはブートフラッシュからシステム イメージを手動でロードできます。

スイッチを再起動し、起動から 60 秒以内に **Break** キーを押すことにより、ROM モニタ モードを開始できます。端末サーバから接続している場合は、エスケープによって Telnet プロンプトを表示し、**send break** コマンドを入力すると、ROM モニタ モードが開始されます。



(注)

コンフィギュレーション レジスタの設定値で **Break** キーがオフに設定されているかどうかに関係なく、システムの再起動から 60 秒間は、**Break** キーが必ず有効です。

ROM モニタには、次の機能が組み込まれています。

- 電源投入時の信頼性テスト
- ハードウェアの初期化
- 起動 (手動起動および自動起動が可能)
- デバッグ ユーティリティおよびクラッシュ分析機能
- モニタ呼び出しインターフェイス (EMT コール — ROM モニタは EMT コールを使用して、実行システム イメージに情報およびある種の機能を提供します)
- ファイル システム (ROM モニタは、単純なファイル システムを認識し、動的にリンクされたファイル システム ライブラリ [MONLIB] によって新しく作成されたファイル システムをサポートします)
- 例外処理

コンフィギュレーションレジスタの概要

コンフィギュレーションレジスタによって、スイッチが OS（オペレーティングシステム）イメージをロードするかどうか、また、どこにシステムイメージを保存するかが決まります。コンフィギュレーションレジスタのブートフィールドによって、起動時に ROM モニタがスーパーバイザエンジンのシステムイメージをロードするかどうか、また、ロードする場合にはその方法が決まります。ブートフィールドを変更することにより、起動時にデフォルトのシステムイメージを使用するのではなく、強制的に特定のシステムイメージを起動させることができます。

ブートフィールドは、16 ビットのコンフィギュレーションレジスタの下位 4 ビット（ビット 3、2、1、および 0）で形成されます。デフォルトのブートフィールド値は 0x10F です。次に挙げるコンフィギュレーションレジスタのブートフィールド設定を使用できます。

- ブートフィールドが 0000 の場合、スイッチはシステムイメージをロードしません。その代わりに、ROM モニタモードを開始します。ROM モニタモードでは、ROM モニタ コマンドを入力して、システムイメージを手動でロードできます。
- ブートフィールドが 0001 の場合、スイッチはオンボードフラッシュメモリ上で見つけた最初に有効なシステムイメージをロードします。
- ブートフィールドの値が 0010 ~ 1111 の場合、スイッチは NVRAM 設定の **boot system** コマンドで指定されたシステムイメージをロードします。**boot system** コマンドの入力順に、イメージの起動を試行します。BOOT 環境変数リストの中のどのイメージも起動できなかった場合は、ROM モニタモードのままになります。正確なブートシーケンスは、ROM モニタによって定義されます。

コンフィギュレーションレジスタの他のビットを設定する場合、次のような機能を持ちます。

- ビット 5 (0x0020) — CONFIG_FILE の反復がイネーブルになります。
- ビット 6 (0x0040) — システムソフトウェアによって NVRAM の内容が消去されます。
- ビット 7 (0x0080) — OEM ビットがイネーブルになります（未使用）。
- ビット 8 (0x0100) — ブレークがディセーブルになります。
- ビット 9 (0x0200) — セカンダリブートストラップが使用されます（ROM モニタでは未使用）。
- ビット 10 (0x0400) — すべてゼロで IP ブロードキャストを行います（未使用）。
- ビット 11/12 (0x0800/0x1000) — コンソールの回線速度を次のように指定します。0/0=9600、0/1=1200、1/0=4800、1/1=2400（デフォルトの設定は 9600）です。
- ビット 13 (0x2000) — ネットワークブートができなかった場合に、デフォルトのフラッシュソフトウェアを起動します（未使用）。
- ビット 14 (0x4000) — IP ブロードキャストにネットワーク番号がありません（未使用）。
- ビット 15 (0x8000) — 診断メッセージをイネーブルにし、NVRAM の内容を無視します（未使用）。

BOOT 環境変数の概要

BOOT 環境変数では、起動時にスイッチの起動元となる各種デバイス上のイメージファイルをリスト形式で指定します。

BOOT 環境変数に複数のイメージを追加すると、フェールセーフ起動設定が得られます。最初のファイルでスイッチを起動できなかった場合、BOOT 環境変数で次に指定されているイメージが試行され、スイッチが起動するか起動を試行するイメージがなくなるまで、この作業が順番に繰り返されます。最終的に、起動できる有効なイメージがなかった場合、システムは ROM モニタモードを開始して、ユーザが手動でブートイメージを指定できるようにします。

システムは、BOOT 環境変数に入力された順序で、イメージを保存および実行します。起動時のイメージ試行順序を変更する場合は、BOOT 環境変数のイメージを追加または消去して、適切な順序になるようにするか、BOOT 環境変数全体をいったん消去して、適切な順序でリストを再度定義します。

CONFIG_FILE 環境変数の概要

CONFIG_FILE 環境変数を使用することにより、起動時のスイッチ設定に使用する各種装置のコンフィギュレーション ファイル (auto-config ファイル) のリストを指定できます。次の動作を指定できます。

- 反復不能 — CONFIG_FILE 環境変数にコンフィギュレーション ファイルのリストを追加すると、次のスイッチ再起動時に、システムによって NVRAM 内の設定が消去され、指定されたファイルを使用してスイッチが設定されます。CONFIG_FILE 環境変数は、スイッチの設定前に消去されます。反復不能がデフォルトの設定です。
- 反復可能 — CONFIG_FILE 環境変数にコンフィギュレーション ファイルのリストを追加すると、リストが NVRAM に無期限に保存されます。スイッチを再起動するたびに、システムによって NVRAM 内の設定が消去され、指定されたコンフィギュレーション ファイルを使用してスイッチが設定されます。CONFIG_FILE 環境変数は消去されません。

反復または反復不能の指定方法については、「[CONFIG_FILE 反復の設定](#)」(p.24-7) を参照してください。

- 上書き — CONFIG_FILE 環境変数にコンフィギュレーション ファイルのリストを追加すると、NVRAM 内の設定が消去され、そのあとでコンフィギュレーション ファイルが実行されます。上書きがデフォルトの設定です。
- 追加 — NVRAM を消去せずに、コンフィギュレーション ファイルが実行されます。
上書きまたは追加の指定方法については、「[CONFIG_FILE 上書きの設定](#)」(p.24-8) を参照してください。
- 同期イネーブル — コンフィギュレーション ファイルが自動的にスタンバイ スーパーバイザ エンジンと同期するよう同期化をイネーブルにします。ファイルはアクティブ スーパーバイザ エンジン上のファイルと同じ状態に保たれます。
- 同期ディセーブル — 同期化をディセーブルにします。
同期化を指定する方法については、「[CONFIG_FILE 同期の設定](#)」(p.24-8) を参照してください。



ヒント

CONFIG_FILE 環境変数は変更可能です。また、起動時にスイッチを設定するコンフィギュレーション ファイル内のコマンドによってプロパティが変更される可能性があります。

CONFIG_FILE 環境変数には、複数のコンフィギュレーション ファイルを追加できます。ローカル フラッシュ デバイス (bootflash: または slot0:) に保存されている任意の有効なコンフィギュレーション ファイルを指定できます。

スイッチの起動時に、CONFIG_FILE 環境変数で指定されたファイルのいずれかが有効なコンフィギュレーション ファイルであった場合、NVRAM 内の設定が消去され、指定のコンフィギュレーション ファイルを使用してスイッチが設定されます。有効なコンフィギュレーション ファイルが複数指定されていた場合、CONFIG_FILE 環境変数に指定された順に、1 つずつコンフィギュレーション ファイルが実行されます。

指定されたファイルが有効なコンフィギュレーション ファイルではなかった場合、そのエントリを無視して次のファイルが試行されます。指定されたイメージがほかになくなるまで、この動作が繰り返されます。有効なコンフィギュレーション ファイルが 1 つも指定されていなかった場合、NVRAM に最後に保存された設定が使用されます。

スイッチのデフォルト起動設定

表 24-1 に、スイッチのデフォルト起動設定を示します。

表 24-1 スイッチのデフォルト起動設定

機能	デフォルト設定
コンフィギュレーションレジスタの値	0x10f
起動方式	BOOT 環境変数で指定されたイメージからシステムを起動
ROM モニタ コンソール ポートのボーレート	9600 ボー
ignore-config パラメータ	ディセーブル
BOOT 環境変数	空
CONFIG_FILE 環境変数	slot0:switch.cfg
CONFIG_FILE 反復に関するコンフィギュレーションレジスタパラメータ	反復不能
CONFIG_FILE 上書きに関するコンフィギュレーションレジスタパラメータ	上書き
CONFIG_FILE 同期に関するコンフィギュレーションレジスタパラメータ	同期ディセーブル

コンフィギュレーションレジスタの設定



(注)

コンフィギュレーションレジスタの設定が、冗長スーパーバイザエンジンに自動的にコピーされることはありません。スイッチ内の各スーパーバイザエンジンに、個別にコンフィギュレーションレジスタを設定する必要があります。

ここでは、コンフィギュレーションレジスタを変更する方法について説明します。

- [コンフィギュレーションレジスタのブートフィールドの設定 \(p.24-6\)](#)
- [ROM モニタ コンソール ポートのボーレートの設定 \(p.24-6\)](#)
- [CONFIG_FILE 反復の設定 \(p.24-7\)](#)
- [CONFIG_FILE 上書きの設定 \(p.24-8\)](#)
- [CONFIG_FILE 同期の設定 \(p.24-8\)](#)
- [スイッチに NVRAM 内の設定情報を無視させる設定 \(p.24-9\)](#)
- [コンフィギュレーションレジスタ値の設定 \(p.24-10\)](#)

コンフィギュレーションレジスタのブートフィールドの設定

コンフィギュレーションレジスタのブートフィールドを設定することにより、次回の起動時にスイッチが使用するブート方式を指定できます。このコマンドが作用するのは、ブートフィールドを制御するコンフィギュレーションレジスタビットだけです。他のビットの設定は変わりません。使用できる起動方式は次のとおりです。

- ROM モニタ — **rommon** キーワードを使用して、起動時にスイッチが ROM モニタ モードにとどまるようにします。
- ブートフラッシュ — **bootflash** キーワードを使用して、オンボードフラッシュメモリ上の最初のイメージからスイッチを起動させます。
- システム — **system** キーワードを使用して、BOOT 環境変数で指定されたイメージから起動させます（デフォルトの設定）。



(注)

set boot config-register boot コマンドで使用するキーワードは、**rommon** および **system** に限定することを推奨します。

コンフィギュレーションレジスタのブートフィールドを設定するには、イネーブルモードで次の作業を行います。

作業	コマンド
コンフィギュレーションレジスタのブートフィールドを設定します。	set boot config-register boot {rommon bootflash system} [mod]

次に、コンフィギュレーションレジスタのブートフィールドを設定する例を示します。

```
Console> (enable) set boot config-register boot rommon
Configuration register is 0x0
ignore-config: disabled
auto-config: non-recurring
console baud: 9600
boot: the ROM monitor
Console> (enable)
```

ROM モニタ コンソール ポートのボーレートの設定

ROM モニタが使用するコンソールポートのボーレートを設定できます。新しいボーレートが使用されるのは、次にスイッチを再起動したときです。このコマンドの作用を受けるのは、ボーレートを制御するコンフィギュレーションレジスタビットだけです。他のビットの設定は変わりません。



(注)

コンフィギュレーションレジスタで指定したボーレートを使用するのは、ROM モニタだけです。**set system baud** コマンドで指定するボーレートとは異なります。

コンフィギュレーションレジスタでROM モニタ コンソールポートのボーレートを設定するには、イネーブルモードで次のコマンドを入力します。

作業	コマンド
コンフィギュレーションレジスタでROM モニタ コンソールポートのボーレートを設定します。	set boot config-register baud {1200 2400 4800 9600} [mod]

次に、コンフィギュレーションレジスタで ROM モニタ コンソール ポートのボーレートを 2400 に設定する例を示します。

```
Console> (enable) set boot config-register baud 2400
Configuration register is 0x1800
ignore-config: disabled
auto-config: non-recurring
console baud: 2400
boot: the ROM monitor
Console> (enable)
```

CONFIG_FILE 反復の設定

デフォルトでは、CONFIG_FILE 環境変数を設定した場合に、起動時に使用するコンフィギュレーションファイルのリストが維持されるのは、次にスイッチを再起動するまでの間だけです。

システム ソフトウェアに CONFIG_FILE 環境変数の設定値を無期限に維持させ、スイッチを再起動するたびに、指定のコンフィギュレーションファイルを使用してスイッチが設定されるようにすることができます。

このコマンドが作用するのは、CONFIG_FILE 環境変数の設定が反復か反復不能かを制御するコンフィギュレーションレジスタビットだけです。他のコンフィギュレーションレジスタビットは変更されません。



注意

CONFIG_FILE 環境変数を **recurring** に設定すると、スイッチを再起動するたびに NVRAM の現在の設定が消去され、指定のコンフィギュレーションファイルを使用してスイッチが設定されます。CONFIG_FILE 環境変数を **non-recurring** に設定すると、次回スイッチを再起動したときに NVRAM の現在の設定が消去され、指定のコンフィギュレーションファイルを使用してスイッチが設定されます。NVRAM 設定は、(再び CONFIG_FILE 変数を設定しないかぎり) その後の再起動後も維持されます。

スイッチが現在の CONFIG_FILE 環境変数を無期限に維持するように設定するには、イネーブルモードで次の作業を行います。

作業	コマンド
現在の CONFIG_FILE 環境変数が無期限に維持されるようにスイッチを設定します。	set boot config-register auto-config {recurring non-recurring}

次に、現在の CONFIG_FILE 環境変数が無期限に維持されるようにスイッチを設定する例を示します。

```
Console> (enable) set boot config-register auto-config recurring
Configuration register is 0x1820
ignore-config: disabled
auto-config: recurring, overwrite, sync disabled
console baud: 2400
boot: the ROM monitor
Console> (enable)
```

CONFIG_FILE 上書きの設定

このコマンドでは、auto-config ファイルを使用して NVRAM 内の設定を上書きするか、NVRAM の現在の内容にファイルの設定を追加するかを指定できます。上書きを指定すると、NVRAM 内の設定が消去されてから、auto-config ファイルが実行されます。追加を指定すると、NVRAM を消去せずに auto-config ファイルが実行されます。デフォルトの設定は **overwrite** です。

auto-config ファイルで NVRAM 内の設定を上書きするか、それとも NVRAM の現在の内容にファイルの設定を追加するかを指定するには、イネーブルモードで次の作業を行います。

作業	コマンド
auto-config ファイルで NVRAM 内の設定を上書きするか、それとも NVRAM の現在の内容にファイルの設定を追加するかを指定します。	set boot config-register auto-config {overwrite append}

次に、auto-config ファイルを使用して NVRAM 内の設定を上書きするように指定する例を示します。

```
Console> (enable) set boot config-register auto-config overwrite
Configuration register is 0x12F
ignore-config: disabled
auto-config: recurring, overwrite, sync disabled
console baud: 9600
boot: image specified by the boot system commands
Console> (enable)
```

次に、NVRAM の現在の内容に auto-config ファイルを追加するように指定する例を示します。

```
Console> (enable) set boot config-register auto-config append
Configuration register is 0x12F
ignore-config: disabled
auto-config: recurring, append, sync disabled
console baud: 9600
boot: image specified by the boot system commands
Console> (enable)
```

CONFIG_FILE 同期の設定

set boot config-register auto-config sync コマンドは、auto-config ファイルを強制的にスタンバイ スーパーバイザ エンジンと自動的に同期化するために同期化をイネーブルにします。ファイルはアクティブ スーパーバイザ エンジン上のファイルと同じ状態に保たれます。デフォルトの設定は **disabled** です。同期化チェックおよび（必要な場合）同期化をトリガするイベントは、次のとおりです。

- どちらかのスーパーバイザ エンジン上での auto-config ファイルの変更（アクティブ スーパーバイザ エンジン上でファイルが削除された場合は、スタンバイ スーパーバイザ エンジン上でも削除されます）
- ブートストリングの CONFIG_FILE 変数の設定変更
- 新しいスーパーバイザ エンジンの搭載
- システムの起動

アクティブ スーパーバイザ エンジンの CONFIG_FILE 変数が、スタンバイ スーパーバイザ エンジン上でも同一に保持されます。アクティブ スーパーバイザ エンジンの各 auto-config ファイルが、スタンバイ スーパーバイザ エンジンの対応する auto-config ファイルと比較されます。2つのファイルの長さおよび Cyclic Redundancy Check (CRC; 巡回冗長検査) が同じであれば、それらのファイルは同一とみなされます。スタンバイ スーパーバイザ エンジン上のファイルがアクティブ スーパーバイザ エン

ジン上のファイルと同じでなければ、アクティブ スーパーバイザ エンジン上のファイル名を使用して、スタンバイ スーパーバイザ エンジン上に新しいファイルが作成されます。スタンバイ スーパーバイザ エンジン上に同名のファイルが存在する場合には、上書きされます。

同期化をイネーブルまたはディセーブルにするには、イネーブル モードで次の作業を行います。

作業	コマンド
同期化をイネーブルにするか、ディセーブルにするかを指定します。	set boot config-register auto-config sync {enable disable}

次に、同期化をイネーブルにする例を示します。

```
Console> (enable) set boot config-register auto-config sync enable
Configuration register is 0x12F
ignore-config: disabled
auto-config: recurring, append, sync enabled
console baud: 9600
boot: image specified by the boot system commands
Console> (enable)
```

次に、同期化をディセーブルにする例を示します。

```
Console> (enable) set boot config-register auto-config sync disable
Configuration register is 0x12F
ignore-config: disabled
auto-config: recurring, append, sync disabled
console baud: 9600
boot: image specified by the boot system commands
Console> (enable)
```

スイッチに NVRAM 内の設定情報を無視させる設定

次にスイッチを再起動するときに、NVRAM に保存されている設定情報をシステム ソフトウェアに無視させることができます。**set boot config-register ignore-config enable** コマンドが作用するのは、スイッチに NVRAM 内の設定を無視させるかどうかを制御するコンフィギュレーション レジスタ ビットだけです。他のビットの設定は変わりません。このコマンドは、次にスイッチを再起動したときにだけ作用します。



注意

ignore-config パラメータをイネーブルにすることは、**clear config all** コマンドを入力するのと同じことです。次回のスイッチ再起動時に、NVRAM に保存されているすべての設定が消去されます。

次回の再起動時に NVRAM 内の設定をスイッチに無視させるには、イネーブル モードで次のコマンドを入力します。

作業	コマンド
起動時に NVRAM の内容を無視するようにスイッチを設定します。	set boot config-register ignore-config enable

■ コンフィギュレーションレジスタの設定

次に、次回の再起動時に NVRAM 内の設定を無視するようにスイッチを設定する例を示します。

```
Console> (enable) set boot config-register ignore-config enable
Configuration register is 0x1860
ignore-config: enabled
auto-config: recurring
console baud: 2400
boot: the ROM monitor
Console> (enable)
```

コンフィギュレーションレジスタ値の設定

コンフィギュレーションレジスタ値を設定するには、イネーブルモードで次の作業を行います。

作業	コマンド
コンフィギュレーションレジスタを設定します。	set boot config-register 0xvalue [mod]

次に、コンフィギュレーションレジスタ値を 0x90f に設定する例を示します。

```
Console> (enable) set boot config-register 0x90f
Configuration register is 0x90f
ignore-config: disabled
auto-config: non-recurring
console baud: 4800
boot: image specified by the boot system commands
Console> (enable)
```

BOOT 環境変数の設定



(注)

BOOT 環境変数の設定値が冗長スーパーバイザ エンジン（存在する場合）に自動的にコピーされることはありません。スイッチ内の各スーパーバイザ エンジンに、個別に BOOT 変数を設定する必要があります。

ここでは、BOOT 環境変数を変更する方法について説明します。

- [BOOT 環境変数の設定 \(p.24-11\)](#)
- [BOOT 環境変数の設定値の消去 \(p.24-11\)](#)

BOOT 環境変数の設定

BOOT 環境変数を設定するには、イネーブル モードで次の作業を行います。

作業	コマンド
BOOT 環境変数を設定します。	set boot system flash device:[filename] [prepend] [mod]

次に、BOOT 環境変数を設定する例を示します。

```
Console> (enable) set boot system flash bootflash:cat6000-sup.5-5-1.bin
BOOT variable = bootflash:cat6000-sup.5-5-1.bin,1;
Console> (enable) set boot system flash bootflash:cat6000-sup.4-5-2.bin
BOOT variable = bootflash:cat6000-sup.5-1-1.bin,1;bootflash:cat6000-sup.4-5-2.
bin,1;
Console> (enable) set boot system flash bootflash:cat6000-sup.5-2-1.bin prepend
BOOT variable = bootflash:cat6000-sup.5-2-1.bin,1;bootflash:cat6000-sup.5-5-1.
bin,1;bootflash:cat6000-sup.4-5-2.bin,1;
Console> (enable)
```

BOOT 環境変数の設定値の消去

BOOT 環境変数からエントリを消去するには、イネーブル モードで次のいずれかの作業を行います。

作業	コマンド
BOOT 環境変数から特定のイメージを消去します。	clear boot system flash device:[filename] [mod]
BOOT 環境変数全体を消去します。	clear boot system all [mod]

次に、BOOT 環境変数から特定のエントリを消去する例を示します。

```
Console> (enable) clear boot system flash bootflash:cat6000-sup.5-1-1.bin
BOOT variable = bootflash:cat6000-sup.5-2-1.bin,1;bootflash:cat6000-sup.4-5-2.
bin,1;
Console> (enable)
```

次に、BOOT 環境変数全体を消去する例を示します。

```
Console> (enable) clear boot system all
BOOT variable =
Console> (enable)
```

CONFIG_FILE 環境変数の設定

ここでは、CONFIG_FILE 環境変数を変更する方法について説明します。

- [CONFIG_FILE 環境変数の設定 \(p.24-12\)](#)
- [CONFIG_FILE 環境変数の設定値の消去 \(p.24-12\)](#)

CONFIG_FILE 環境変数の設定

`set boot auto-config` コマンドを使用し、セミコロン (;) で区切ることにより、複数のコンフィギュレーション ファイルを指定できます。各コンフィギュレーション ファイルに、装置名とファイル名の両方を指定する必要があります。



(注)

CONFIG_FILE 環境変数の前後にコンフィギュレーション ファイルを付加することはできません。`set boot auto-config` コマンドを入力すると、それまで `set boot auto-config` コマンドによって指定されていたコンフィギュレーション ファイル リストが消去されます。

CONFIG_FILE 環境変数を設定するには、イネーブル モードで次の作業を行います。

作業	コマンド
CONFIG_FILE 環境変数を設定します。	<code>set boot auto-config device:filename[;device:filename...]</code>

次に、CONFIG_FILE 環境変数を設定する例を示します。

```
Console> (enable) set boot auto-config bootflash:generic.cfg;bootflash:6509_1_noc.cfg
CONFIG_FILE variable = bootflash:generic.cfg;bootflash:6509_1_noc.cfg
WARNING: nvram configuration may be lost during next bootup,
        and re-configured using the file(s) specified.
Console> (enable)
```

CONFIG_FILE 環境変数の設定値の消去

CONFIG_FILE 環境変数からエントリを消去するには、イネーブル モードで次の作業を行います。

作業	コマンド
CONFIG_FILE 環境変数のエントリを消去します。	<code>clear boot auto-config</code>

次に、CONFIG_FILE 環境変数のエントリを消去する例を示します。

```
Console> (enable) clear boot auto-config
CONFIG_FILE variable =
Console> (enable)
```

スイッチの起動設定の表示

現在のコンフィギュレーションレジスタ、BOOT 環境変数、および CONFIG_FILE 環境変数の設定値を表示するには、次の作業を行います。

作業	コマンド
現在のコンフィギュレーションレジスタ、BOOT 環境変数、および CONFIG_FILE 環境変数の設定値を表示します。	<code>show boot [mod]</code>

次に、現在のコンフィギュレーションレジスタ、BOOT 環境変数、および CONFIG_FILE 環境変数の設定値を表示する例を示します。

```
Console> (enable) show boot
BOOT variable = bootflash:cat6000-sup.5-2-1.bin,1;
CONFIG_FILE variable = bootflash:generic.cfg;bootflash:6509_1_noc.cfg

Configuration register is 0x12f
ignore-config: disabled
auto-config: recurring
console baud: 9600
boot: image specified by the boot system commands

Console> (enable)
```

