



Cisco IOS XE Fuji 16.9.x (Catalyst 9300 スイッチ) システム管理 コンフィギュレーションガイド

初版：2018年7月18日

最終更新：2019年2月19日

シスコシステムズ合同会社

〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー

<http://www.cisco.com/jp>

お問い合わせ先：シスコ コンタクトセンター

0120-092-255 (フリーコール、携帯・PHS含む)

電話受付時間：平日 10:00～12:00、13:00～17:00

<http://www.cisco.com/jp/go/contactcenter/>



目次

第 1 章

デバイスの管理 1

デバイスの管理に関する情報 1

システム日時の管理 1

システムクロック 1

ネットワークタイムプロトコル 2

NTPストラタム 3

NTPアソシエーション 4

NTPセキュリティ 5

特定のインターフェイス上のNTPサービス 7

NTPパケットの送信元IPアドレス 7

NTPの実装 7

システム名およびシステムプロンプト 8

スタックのシステム名およびシステムプロンプト 9

デフォルトのシステム名とプロンプトの設定 9

DNS 9

DNSのデフォルト設定値 9

ログインバナー 10

バナーのデフォルト設定 10

MACアドレステーブル 10

MACアドレステーブルの作成 10

MACアドレスおよびVLAN 11

MACアドレスおよびデバイスのスタック 11

MACアドレステーブルのデフォルト設定 11

ARPテーブルの管理 12

デバイスを管理する方法	12
手動による日付と時刻の設定	12
システムクロックの設定	12
タイムゾーンの設定	13
夏時間の設定	14
NTPの設定	18
NTPのデフォルト設定	18
NTP認証の設定	18
ポーリングベースのNTPアソシエーションの設定	20
ブロードキャストベースのNTPアソシエーションの設定	22
NTPアクセス制限の設定	23
システム名の設定	26
DNSの設定	27
Message-of-the-Day ログインバナーの設定	29
ログインバナーの設定	31
MACアドレステーブルの管理	32
アドレスエイジングタイムの変更	32
MACアドレス変更通知トラップの設定	33
MACアドレス移動通知トラップの設定	36
MACしきい値通知トラップの設定	38
スタティックアドレスエントリの追加および削除	40
ユニキャストMACアドレスフィルタリングの設定	42
デバイスのモニターリングおよび保守の管理	43
デバイス管理の設定例	44
例：システムクロックの設定	44
例：サマータイムの設定	44
例：MOTDバナーの設定	45
例：ログインバナーの設定	45
例：MACアドレス変更通知トラップの設定	46
例：MACしきい値通知トラップの設定	46
例：MACアドレステーブルへのスタティックアドレスの追加	46

例：ユニキャスト MAC アドレス フィルタリングの設定	47
デバイス管理に関する追加情報	47
デバイス管理の機能履歴	48

第 2 章

ブート整合性の可視性	49
ブート整合性の可視性について	49
ソフトウェアイメージとハードウェアの確認	49
プラットフォーム ID とソフトウェア整合性の確認	50
ブート整合性の可視性に関する追加情報	52
ブート整合性の可視性の機能履歴	53

第 3 章

デバイスのセットアップ設定の実行	55
ソフトウェア インストールの制約事項	55
デバイスセットアップ設定の実行に関する情報	55
デバイスブート プロセス	55
ソフトウェア インストールの概要	56
ソフトウェアのブート モード	57
ソフトウェア パッケージのインストール	58
ソフトウェアインストールの終了	58
Devices 情報の割り当て	59
デフォルトのスイッチ情報	59
DHCP ベースの自動設定の概要	60
DHCP クライアントの要求プロセス	61
DHCP ベースの自動設定およびイメージアップデート	62
DHCP ベースの自動設定の制約事項	62
DHCP 自動設定	62
DHCP 自動イメージアップデート	63
DHCP サーバ設定時の注意事項	63
TFTP サーバの目的	64
DNS サーバの目的	65
コンフィギュレーション ファイルの入手方法	65

環境変数の制御方法	66
TFTP の環境変数	67
ソフトウェア イメージのリロードのスケジューリング	67
デバイスセットアップ設定の実行方法	68
DHCP 自動設定 (コンフィギュレーション ファイルだけ) の設定	68
DHCP 自動イメージアップデート (コンフィギュレーション ファイルおよびイメージ) の設定	70
DHCP サーバからファイルをダウンロードするクライアントの設定	74
複数の SVI への IP 情報の手動割り当て	75
デバイスのスタートアップ コンフィギュレーションの変更	77
システム コンフィギュレーションを読み書きするためのファイル名の指定	77
スイッチの手動による起動	78
インストール モードでのデバイスのブート	80
デバイスをバンドルモードで起動する場合	82
ソフトウェア イメージのリロードのスケジューリング設定	83
デバイスのセットアップ設定のモニターリング	84
例: インストール モードでのソフトウェアブートアップ ディスプレイ	84
例: 緊急インストール	87
デバイスのセットアップを実行する場合の設定例	89
例: 更新プログラム パッケージの管理	89
ソフトウェア インストールの確認	99
例: DHCP サーバとしてのデバイスの設定	102
例: DHCP 自動イメージアップデートの設定	103
例: DHCP サーバから設定をダウンロードするための デバイス の設定	103
例: ソフトウェア イメージのリロードのスケジューリング	104
デバイスセットアップの実行に関する追加情報	104
デバイスセットアップ設定の実行に関する機能履歴	104
第 4 章	スマート ライセンスの設定 107
	スマートライセンシングの設定の前提条件 107
	スマートライセンシングの概要 107

CSSM の概要	108
CSSM への接続	109
CSSM への既存のライセンスのリンク	111
CSSM への接続の設定とライセンスレベルの設定	111
CSSM への接続の設定	112
ダイレクトクラウドアクセス用の Call Home サービスの設定	114
HTTPS プロキシサーバを介したダイレクトクラウドアクセス用の Call Home サービスの設定	116
Cisco Smart Software Manager オンプレミス用の Call Home サービスの設定	119
ライセンスレベルの設定	122
CSSM でのデバイスの登録	124
CSSM からの新しいトークンの生成	124
新しいトークンを使用するデバイスの登録	126
登録後のライセンスステータスの確認	127
CSSM でのデバイスの登録キャンセル	128
スマートライセンスの設定のモニターリング	130
スマートライセンシングの設定例	131
例：Call Home プロファイルの表示	131
例：登録前のライセンス情報の表示	132
例：デバイスの登録	134
例：登録後のライセンスステータスの表示	135
その他の参考資料	138
スマートライセンシングに関する機能情報	138

 第 5 章

有線ネットワークでの Application Visibility and Control の設定	141
機能情報の確認	141
有線ネットワークでの Application Visibility and Control について	142
サポートされる AVC クラス マップおよびポリシー マップのフォーマット	142
有線 Application Visibility and Control の制限	144
Application Visibility and Control の設定方法	145
有線ネットワークでの Application Visibility and Control の設定	145

インターフェイスでのアプリケーション認識の有効化	146
AVC QoS ポリシーの作成	147
スイッチポートへの QoS ポリシーの適用	150
有線 AVC Flexible Netflow の設定	150
NBAR2 カスタム アプリケーション	167
NBAR2 ダイナミック ヒットレス プロトコル パックのアップグレード	170
Application Visibility and Control のモニターリング	172
例：Application Visibility and Control の設定	173
基本的なトラブルシューティング：質問と回答	183
Application Visibility and Control に関する追加情報	184
有線ネットワークでの Application Visibility and Control の機能履歴	185

第 6 章

SDM テンプレートの設定	187
SDM テンプレート	187
SDM テンプレートとスイッチ スタック	187
SDM テンプレートの設定方法	188
SDM テンプレートの設定	188
SDM テンプレートのモニターリングおよびメンテナンス	189
SDM テンプレートの設定例	189
例：SDM テンプレートの表示	189
例：SDM テンプレートの設定	190
SDM テンプレートに関する追加情報	190
SDM テンプレートの機能履歴	191

第 7 章

システム メッセージ ログの設定	193
機能情報の確認	193
システム メッセージ ログの設定に関する情報	193
システム メッセージ ロギング	193
システム ログ メッセージのフォーマット	194
デフォルトのシステム メッセージ ロギングの設定	195
syslog メッセージの制限	196

システム メッセージ ログの設定方法	196
メッセージ表示宛先デバイスの設定	196
ログ メッセージの同期化	198
メッセージ ロギングのディセーブル化	200
ログ メッセージのタイム スタンプのイネーブル化およびディセーブル化	201
ログ メッセージのシーケンス番号のイネーブル化およびディセーブル化	202
メッセージ重大度の定義	203
履歴テーブルおよび SNMP に送信される syslog メッセージの制限	204
UNIX Syslog デーモンへのメッセージのロギング	205
システム メッセージ ログのモニタリングおよびメンテナンス	206
コンフィギュレーション アーカイブ ログのモニタリング	206
システム メッセージ ログの設定例	206
例：システム メッセージのスタック構成	206
例：スイッチ システム メッセージ	207
システム メッセージ ログの機能履歴と情報	207

第 8 章

オンライン診断の設定 209

オンライン診断の設定に関する情報	209
Generic Online Diagnostics (GOLD)	210
TestPortTxMonitoring	210
TestUnusedPortLoopback	211
オンライン診断の設定方法	211
オンライン診断テストの開始	211
オンライン診断の設定	212
オンライン診断のスケジューリング	213
ヘルス モニタリング診断の設定	214
オンライン診断のモニタリングおよびメンテナンス	217
オンライン診断テストの設定例	217
例：診断テストの開始	217
例：ヘルス モニタリング テストの設定	217
例：診断テストのスケジューリング	218

例：オンライン診断の表示	218
オンライン診断に関する追加情報	219
オンライン診断設定の機能情報	220

第 9 章

コンフィギュレーション ファイルの管理 221

コンフィギュレーション ファイルの管理の前提条件	221
コンフィギュレーション ファイルの管理の制約事項	221
コンフィギュレーション ファイルの管理について	222
コンフィギュレーション ファイルのタイプ	222
コンフィギュレーション モードおよびコンフィギュレーション ソースの選択	222
CLI を使用したコンフィギュレーション ファイルの変更	223
コンフィギュレーション ファイルの場所	223
ネットワーク サーバからデバイスへのコンフィギュレーション ファイルのコピー	224
デバイスから TFTP サーバへのコンフィギュレーション ファイルのコピー	224
デバイスから RCP サーバへのコンフィギュレーション ファイルのコピー	225
デバイスから FTP サーバへのコンフィギュレーション ファイルのコピー	227
VRF によるファイルのコピー	228
スイッチから別のスイッチへのコンフィギュレーション ファイルのコピー	228
NVRAM より大きいコンフィギュレーション ファイル	228
コンフィギュレーション ファイルをダウンロードするデバイスの設定	230
コンフィギュレーション ファイル情報の管理方法	230
コンフィギュレーション ファイル情報の表示	230
コンフィギュレーション ファイルの変更	231
デバイスから TFTP サーバへのコンフィギュレーション ファイルのコピー	233
次の作業	234
デバイスから RCP サーバへのコンフィギュレーション ファイルのコピー	235
例	236
次の作業	236
デバイスから FTP サーバへのコンフィギュレーション ファイルのコピー	237
例	238
次の作業	239

TFTP サーバからデバイスへのコンフィギュレーション ファイルのコピー	239
次の作業	240
rcp サーバからデバイスへのコンフィギュレーション ファイルのコピー	240
例	241
次の作業	242
FTP サーバからデバイスへのコンフィギュレーション ファイルのコピー	242
例	243
次の作業	244
NVRAM より大きいコンフィギュレーション ファイルの保守	244
コンフィギュレーション ファイルの圧縮	244
コンフィギュレーションのクラス A フラッシュ ファイル システム上のフラッシュ メモリへの格納	246
ネットワークからのコンフィギュレーション コマンドのロード	248
フラッシュ メモリからスタートアップまたは実行コンフィギュレーションへのコンフィギュレーション ファイルのコピー	249
フラッシュ メモリ ファイル システム間でのコンフィギュレーション ファイルのコピー	250
FTP サーバからフラッシュ メモリ デバイスへのコンフィギュレーション ファイルのコピー	252
次の作業	253
RCP サーバからフラッシュ メモリ デバイスへのコンフィギュレーション ファイルのコピー	253
TFTP サーバからフラッシュ メモリ デバイスへのコンフィギュレーション ファイルのコピー	254
スタートアップ コンフィギュレーション ファイルでのコンフィギュレーション コマンドの再実行	255
スタートアップ コンフィギュレーションのクリア	255
指定されたコンフィギュレーション ファイルの削除	256
クラス A フラッシュ ファイル システムでの CONFIG_FILE 環境変数の指定	257
次の作業	259
コンフィギュレーション ファイルをダウンロードするデバイスの設定	260
ネットワーク コンフィギュレーション ファイルをダウンロードするデバイスの設定	260

ホスト コンフィギュレーション ファイルをダウンロードするデバイスの設定	262
コンフィギュレーション ファイルの管理の機能履歴	263

第 10 章**コンフィギュレーションの置換とロールバック 265**

コンフィギュレーションの置換とロールバックの前提条件	265
コンフィギュレーションの置換とロールバックの制約事項	266
コンフィギュレーションの置換とロールバックについて	266
コンフィギュレーション アーカイブ	266
コンフィギュレーションの置換	267
コンフィギュレーション ロールバック	268
コンフィギュレーション ロールバック変更確認	269
コンフィギュレーションの置換とロールバックの利点	269
コンフィギュレーションの置換とロールバックの使用方法	269
コンフィギュレーション アーカイブの作成	269
コンフィギュレーションの置換またはロールバックの実行	271
機能のモニタリングおよびトラブルシューティング	274
コンフィギュレーションの置換とロールバックの設定例	277
コンフィギュレーション アーカイブの作成	277
現在の実行コンフィギュレーションを保存された Cisco IOS コンフィギュレーション ファイルで置換	277
スタートアップ コンフィギュレーション ファイルへの復帰	278
configure confirm コマンドを使用したコンフィギュレーション置換操作の実行	278
コンフィギュレーション ロールバック操作の実行	278
コンフィギュレーションの置換およびコンフィギュレーションのロールバックの機能履歴	279

第 11 章**ソフトウェア メンテナンス アップグレード 281**

ソフトウェア メンテナンス アップグレードの制約事項	281
ソフトウェア メンテナンス アップグレードについて	281
SMU の概要	281
SMU のワークフロー	282

SMU パッケージ	282
SMU のリロード	282
ソフトウェア メンテナンスの更新の管理方法	283
SMU パッケージのインストール	283
SMU パッケージの管理	284
ソフトウェア メンテナンス アップグレードの設定例	285
ソフトウェア メンテナンス アップグレードのその他の参考資料	289
ソフトウェア メンテナンス アップグレードの機能の履歴	289

 第 12 章

フラッシュ ファイル システムの操作	291
機能情報の確認	291
フラッシュ ファイル システムについて	291
使用可能なファイル システムの表示	292
デフォルト ファイル システムの設定	295
ファイル システムのファイルに関する情報の表示	295
ディレクトリの変更および作業ディレクトリの表示	296
ディレクトリの作成	297
ディレクトリの削除	298
ファイルのコピー	298
ファイルの削除	299
ファイルの作成、表示、および抽出	300
フラッシュ ファイル システムに関するその他の関連資料	302
フラッシュファイルシステムの機能履歴	303

 第 13 章

工場出荷時の状態へのリセット (Factory Reset)	305
初期設定へのリセット実行の前提条件	305
工場出荷時の状態へのリセットの実行に関する制限	305
初期設定へのリセットに関する情報	306
工場出荷時の状態へのリセット方法	307
工場出荷時の状態へのリセットに関する機能履歴と情報	307

第 14 章	セキュアストレージの設定	309
	セキュアストレージについて	309
	セキュアストレージの有効化	309
	セキュアストレージの無効化	310
	暗号化のステータスの確認	311
	セキュアストレージの機能情報	311

第 15 章	条件付きデバッグとラジオアクティブトレース	313
	機能情報の確認	313
	条件付きデバッグの概要	313
	ラジオアクティブトレースの概要	314
	条件付きデバッグとラジオアクティブトレースの設定方法	315
	条件付きデバッグおよび放射線トレース	315
	トレースファイルの場所	315
	条件付きデバッグの設定	315
	L2 マルチキャストの放射線トレース	317
	トレースファイルの推奨ワークフロー	318
	ボックス外へのトレースファイルのコピー	318
	条件付きデバッグのモニターリング	319
	条件付きデバッグの設定例	319
	条件付きデバッグとラジオアクティブトレースに関するその他の関連資料	320
	条件付きデバッグとラジオアクティブトレースの機能履歴	321

第 16 章	ソフトウェア設定のトラブルシューティング	323
	ソフトウェア設定のトラブルシューティングに関する情報	323
	スイッチのソフトウェア障害	323
	のパスワードを紛失したか忘れた場合 デバイス	324
	Power over Ethernet (PoE) ポート	324
	電力消失によるポートの障害	325
	不正リンク アップによるポート障害	325

ping	325
レイヤ2 トレースルート	325
レイヤ2 の traceroute のガイドライン	326
IP トレースルート	327
debug コマンド	328
システム レポート	328
スイッチのオンボード障害ロギング	330
ファン障害	331
CPU 使用率が高い場合に起こりうる症状	332
ソフトウェア設定のトラブルシューティング方法	332
ソフトウェア障害からの回復	332
パスワードを忘れた場合の回復	336
パスワード回復がイネーブルになっている場合の手順	338
パスワード回復がディセーブルになっている場合の手順	339
スイッチ スタック問題の回避	341
自動ネゴシエーションの不一致の防止	342
SFP モジュールのセキュリティと識別に関するトラブルシューティング	342
SFP モジュール ステータスのモニタリング	343
ping の実行	343
温度のモニタリング	344
物理パスのモニタリング	344
IP traceroute の実行	344
デバッグおよびエラー メッセージ出力のリダイレクト	345
show platform forward コマンドの使用	345
show debug コマンドの使用方法	345
OBFL の設定	346
ソフトウェア設定のトラブルシューティングの確認	347
OBFL 情報の表示	347
例：高い CPU 使用率に関する問題と原因の確認	348
ソフトウェア設定のトラブルシューティングのシナリオ	349
Power over Ethernet (PoE) に関するトラブルシューティングのシナリオ	349

ソフトウェアのトラブルシューティングの設定例 354

例：IP ホストの ping 354

例：IP ホストに対する traceroute の実行 355

ソフトウェア設定のトラブルシューティングの機能履歴 356



第 1 章

デバイスの管理

- デバイスの管理に関する情報 (1 ページ)
- デバイスを管理する方法 (12 ページ)
- デバイス管理の設定例 (44 ページ)
- デバイス管理に関する追加情報 (47 ページ)
- デバイス管理の機能履歴 (48 ページ)

デバイスの管理に関する情報

システム日時の管理

device のシステム日時は、自動設定方式 (RTC および NTP) または手動設定方式を使用して管理できます。



(注) ここで使用するコマンドの構文および使用方法の詳細については、*Cisco.com* で、『*Cisco IOS Configuration Fundamentals Command Reference*』を参照してください。

システム クロック

時刻サービスの基本となるのはシステム クロックです。このクロックはシステムがスタートアップした瞬間から稼働し、日時を常時トラッキングします。

システム クロックは、次のソースにより設定できます。

- NTP
- 手動設定

システム クロックは、次のサービスに時刻を提供します。

- user **show** コマンド

- ログおよびデバッグ メッセージ

システム クロックは、グリニッジ標準時 (GMT) ととも呼ばれる協定世界時 (UTC) に基づいて内部的に時刻を追跡します。ローカルのタイムゾーンおよび夏時間に関する情報を設定することにより、時刻がローカルのタイムゾーンに応じて正確に表示されるようになります。

システムクロックは、時刻に信頼性があるかどうか (つまり、信頼できると見なされるタイムソースによって時刻が設定されているか) を常時トラッキングします。信頼性のない場合は、時刻は表示目的でのみ使用され、再配信されません。

ネットワーク タイム プロトコル

NTP は、ネットワーク上のデバイス間の時刻の同期化を目的に設計されています。NTP はユーザ データグラム プロトコル (UDP) で稼働し、UDP は IP 上で稼働します。NTP は RFC 1305 で規定されています。

NTP ネットワークは通常、タイム サーバに接続されたラジオクロックやアトミック クロックなど、正規の時刻源から時刻を取得します。NTP は、ネットワークにこの時刻を分配します。NTP はきわめて効率的で、1 分間に 1 パケットを使用するだけで、2 台のデバイスを 1 ミリ秒以内に同期化できます。

NTP では、信頼できるタイムソースから各マシンが何 NTP ホップ隔たっているかを表すために、ストラタムという概念が使用されます。ストラタム 1 タイム サーバには、ラジオクロックまたは原子時計が直接接続されており、ストラタム 2 タイム サーバは、NTP を使用してストラタム 1 タイム サーバから時刻を取得します (以降のストラタムも同様です)。NTP が稼働するデバイスは、タイムソースとして、NTP を使用して通信するストラタム番号が最小のデバイスを自動的に選択します。この方法によって、NTP 時刻配信の自動編成型ツリーが効率的に構築されます。

NTP では、同期化されていないデバイスと同期化しないことによって、時刻が正確でないデバイスとの同期化を防ぎます。また、NTP では、複数のデバイスから報告される時刻を比較して、ストラタムの番号が小さくても、時刻が他のデバイスと大幅に異なるデバイスとは同期化しません。

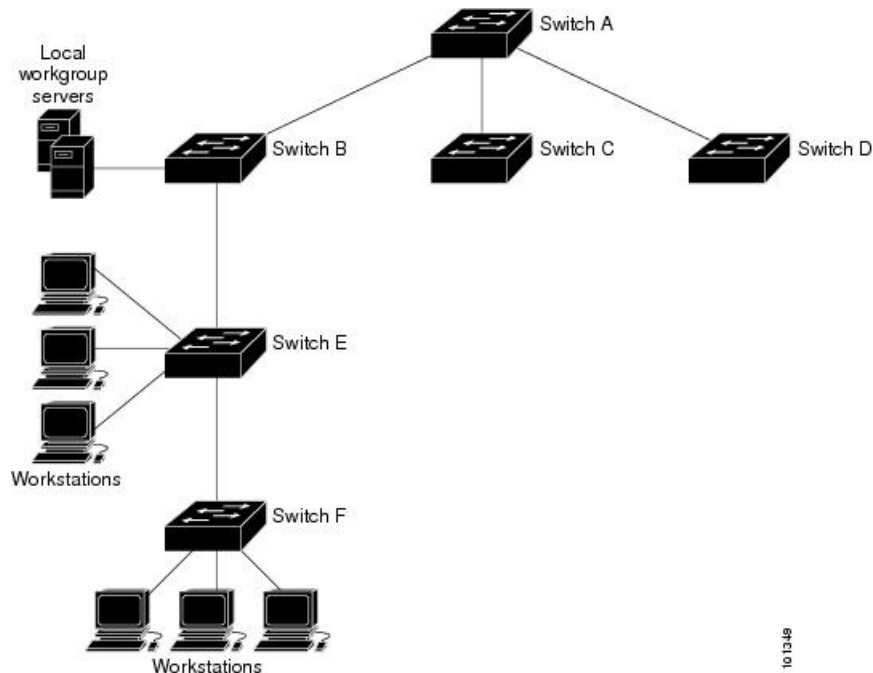
NTP が稼働するデバイス間の通信 (アソシエーション) は、通常静的に設定されます。各デバイスには、アソシエーションを作成すべきすべてのデバイスの IP アドレスが与えられます。アソシエーションのペアとなるデバイス間で NTP メッセージを交換することによって、正確な時刻の維持が可能になります。ただし、LAN 環境では、代わりに IP ブロードキャストメッセージを使用するように NTP を設定できます。各デバイスを、単にブロードキャストメッセージを送受信するように設定すればよいので、この代替手段によって設定の複雑さが緩和されます。ただし、情報の流れは一方向に限られます。

デバイス上で維持される時刻は、重要なリソースです。NTP のセキュリティ機能を使用して、不正確な時刻が誤って、あるいは意図的に設定されることがないようにしてください。その方法として、アクセスリストベースの制約方式と暗号化認証方式があります。

シスコによる NTP の実装では、ストラタム 1 サービスをサポートしていないため、ラジオクロックまたは原子時計に接続できません。ネットワークの時刻サービスは、IP インターネット上のパブリック NTP サーバから取得することを推奨します。

次の図に NTP を使用した一般的なネットワークの例を示します。A はプライマリ NTP、デバイス B、C、D が NTP サーバーモードに設定されている（デバイス A との間にはサーバーアソシエーションが設定されている）場合の NTP マスターです。デバイス E は、アップストリームデバイス（デバイス B）とダウンストリームデバイス（デバイス F）の NTP ピアとして設定されます。

図 1: 一般的な NTP ネットワークの構成



ネットワークがインターネットから切り離されている場合、シスコの NTP によって、実際には、他の方法で時刻を学習しているにもかかわらず、デバイスが NTP を使用して同期化しているように動作を設定できます。他のデバイスは、NTP によりこのデバイスと同期化されます。

複数のタイム ソースがある場合は、NTP は常に、より信頼性があると見なされます。NTP の時刻は、他の方法による時刻に優先します。

自社のホスト システムに NTP ソフトウェアを組み込んでいるメーカーが数社あり、UNIX システム用のバージョンやその派生ソフトウェアも一般に入手できます。このソフトウェアによって、ホストシステムも時間が同期化されます。

NTP ストラタム

NTP では、信頼できるタイム ソースから各マシンが何 NTP ホップ隔たっているかを表すために、ストラタムという概念が使用されます。ストラタム 1 タイム サーバには、ラジオクロックまたは原子時計が直接接続されており、ストラタム 2 タイム サーバは、NTP を使用してストラタム 1 タイム サーバから時刻を取得します（以降のストラタムも同様です）。NTP が稼働するデバイスは、タイム ソースとして、NTP を使用して通信するストラタム番号が最小の

デバイスを自動的に選択します。この方法によって、NTP時刻配信の自動編成型ツリーが効率的に構築されます。

NTPでは、同期化されていないデバイスと同期化しないことによって、時刻が正確でないデバイスとの同期化を防ぎます。また、NTPでは、複数のデバイスから報告される時刻を比較して、ストラタムの番号が小さくても、時刻が他のデバイスと大幅に異なるデバイスとは同期化しません。

NTP アソシエーション

NTPが稼働するデバイス間の通信（アソシエーション）は、通常静的に設定されます。各デバイスには、アソシエーションを作成すべきすべてのデバイスのIPアドレスが与えられます。アソシエーションのペアとなるデバイス間でNTPメッセージを交換することによって、正確な時刻の維持が可能になります。ただし、LAN環境では、代わりにIPブロードキャストメッセージを使用するようにNTPを設定できます。各デバイスを、単にブロードキャストメッセージを送受信するように設定すればよいので、この代替手段によって設定の複雑さが緩和されます。ただし、情報の流れは一方向に限られます。

ポーリングベースのNTPアソシエーション

NTPを実行しているネットワークデバイスには、時刻を基準時刻源と同期する際にさまざまなアソシエーションモードで動作するように設定できます。ネットワークデバイスは、2つの方法でネットワーク上の時刻情報を取得できます。それらは、ホストサービスのポーリングとNTPブロードキャストのリスニングです。ここでは、ポーリングベースのアソシエーションモードを中心に説明します。ブロードキャストベースのNTPアソシエーションの詳細については、「ブロードキャストベースのNTPアソシエーション」を参照してください。

最も一般的に使用される2つのポーリングベースのアソシエーションモードは次のとおりです。

- クライアントモード
- 対称アクティブモード

クライアントモードと対称アクティブモードは、高レベルの時刻の精度と信頼性を提供するためにNTPが必要になる場合に使用します。

クライアントモードで動作しているネットワークデバイスは、自身に割り当てられている時刻提供ホストをポーリングして現在の時刻を取得します。次に、ネットワークデバイスは、ポーリングされたすべてのタイムサーバーから、同期に使用するホストを選択します。この場合は、確立された関係がクライアントホスト関係なので、ホストがローカルクライアントデバイスから送信された時刻情報をキャプチャしたり使用したりすることはありません。このモードが最も適しているのは、他のローカルクライアントにどのような形式の時刻同期も提供する必要のない、ファイルサーバーおよびワークステーションのクライアントです。ネットワークデバイスを同期させるタイムサーバーを個別に指定し、クライアントモードで動作するようにネットワークデバイスを設定するには、**ntp server** コマンドを使用します。

対称アクティブモードで動作しているネットワークデバイスは、自身に割り当てられている時刻提供ホストをポーリングして現在の時刻を取得し、そのホストによるポーリングに応

答します。これはピアツーピアの関係なので、ホストは、通信相手のローカルネットワーク デバイスの時刻関連情報も保持します。このモードは、さまざまなネットワークパスを経由で多数の冗長サーバーが相互接続されている場合に使用します。インターネット上のほとんどの **Stratum 1** および **Stratum 2** サーバーは、この形式のネットワーク設定を採用しています。ネットワーク デバイスを同期させる時刻提供ホストを個別に指定し、対称アクティブモードで動作するようにネットワーク デバイスを設定するには、**ntp peer** コマンドを使用します。

各ネットワーク デバイスの設定モードを決定する際には、タイムキーピング デバイスとしてのそのデバイスの役割（サーバーかクライアントか）と、そのデバイスが **Stratum 1** タイムキーピング サーバーにどれだけ近いかを主に考慮してください。

ネットワーク デバイスは、クライアントモードでクライアントまたはホストとして動作する場合、または対称アクティブモードでピアとして動作する場合にポーリングに関与します。通常、ポーリングによってメモリおよび CPU リソース（帯域幅など）に負荷が生じることはありませんが、システム上で進行または同時実行しているポーリングの数がきわめて多い場合には、システムの性能に深刻な影響があったり、特定のネットワークの性能が低下したりする可能性があります。過剰な数のポーリングがネットワーク上で進行することを防止するには、直接的なピアツーピアアソシエーションまたはクライアントからサーバーへのアソシエーションを制限する必要があります。代わりに、局所的なネットワーク内に NTP ブロードキャストを使用して時刻情報を伝播することを検討します。

ブロードキャストベースの NTP アソシエーション

ブロードキャストベースの NTP アソシエーションは、時刻の精度および信頼性要件が適度であり、ネットワークが局所的であり、クライアント数が 20 を超える場合に使用します。また、帯域幅、システムメモリ、または CPU リソースが制限されているネットワークにおいても、ブロードキャストベースの NTP アソシエーションの使用をお勧めします。

ブロードキャストクライアントモードで動作しているネットワーク デバイスはポーリングに関与しません。代わりに、ブロードキャストタイムサーバーによって転送される NTP ブロードキャストパケットを待ち受けます。その結果、時刻情報の流れが一方向に限られるため、時刻の精度がわずかに低下する可能性があります。

ネットワークを通じて伝播される NTP ブロードキャストパケットをリッスンするようにネットワーク デバイスを設定するには、**ntp broadcast client** コマンドを使用します。ブロードキャストクライアントモードが動作するためには、ブロードキャストサーバーとそのクライアントが同じサブネット上に存在する必要があります。**ntp broadcast** コマンドを使用して、特定のデバイスのインターフェイスで NTP ブロードキャストパケットを送信するタイムサーバーを有効にする必要があります。

NTP セキュリティ

デバイス上で維持される時刻は、重要なリソースです。NTP のセキュリティ機能を使用して、不正確な時刻が誤って、あるいは意図的に設定されないようにしてください。その方法として、アクセスリストベースの制約方式と暗号化認証方式があります。

NTP アクセス グループ

アクセスリストベースの制限スキームを使用すると、ネットワーク全体、ネットワーク内のサブネット、またはサブネット内のホストに対し、特定のアクセス権限を許可または拒否できます。NTP アクセスグループを定義するには、グローバルコンフィギュレーションモードで `ntp access-group` コマンドを使用します。

アクセスグループのオプションは、次の順序で制限の緩いものから厳しいものへとスキャンされます。

1. `ipv4` : IPv4 アクセスリストを設定します。
2. `ipv6` : IPv6 アクセスリストを設定します。
3. `peer` : 時刻要求と NTP 制御クエリを許可し、システムがアクセスリストの基準を満たすアドレスを持つ別のシステムに同期することを許可します。
4. `serve` : 時刻要求と NTP 制御クエリを許可しますが、システムがアクセスリストの基準を満たすアドレスを持つ別のシステムに同期することは許可しません。
5. `serve-only` : アクセスリストの条件を満たすアドレスを持つシステムからの時刻要求のみを許可します。
6. `query-only` : アクセスリストの基準を満たすアドレスを持つ別のシステムからの NTP 制御クエリのみを許可します。

送信元 IP アドレスが複数のアクセス タイプのアクセス リストに一致する場合は、最初のアクセス タイプのアクセスが認可されます。アクセス グループが指定されていない場合は、すべてのシステムへのアクセスがすべてのアクセスタイプに対して認可されます。アクセスグループが指定されている場合は、指定されたアクセス タイプに対してのみアクセスが認可されます。

NTP 制御クエリの詳細については、RFC 1305 を参照してください。

信頼できる形式のアクセス コントロールが必要な場合は、暗号化された NTP 認証方式を使用する必要があります。IP アドレスに基づくアクセス リストベースの制約方式とは異なり、暗号化認証方式では、認証キーと認証プロセスを使用して、ローカルネットワーク上の指定されたピアまたはサーバーによって送信された NTP 同期パケットが信頼できると見なされるかどうかを、一緒に伝送された時刻情報を受け入れる前に判断します。

認証プロセスは、NTP パケットが作成されるとすぐに開始されます。暗号チェックサム キーは、Message-Digest Algorithm 5 (MD5) を使用して生成され、受信側クライアントに送信される NTP 同期パケットに埋め込まれます。パケットがクライアントによって受信されると、暗号チェックサムキーが復号され、信頼できるキーのリストに対してチェックされます。一致する認証キーがパケットに含まれる場合、受信側クライアントは、パケットに含まれるタイムスタンプ情報を受け入れます。一致するオーセンティケーター キーが含まれていない NTP 同期パケットは無視されます。



- (注) 信頼できるキーを多数設定する必要がある大規模なネットワークでは、信頼できるキーの範囲設定機能を使用して複数のキーを同時に有効にすることができます。

NTP 認証で使用される暗号化および復号化プロセスでは、CPU に非常に大きな負荷がかかる場合があります。ネットワーク内で伝播される時刻の精度が大きく低下する可能性があることに注意してください。より包括的なアクセス コントロール モデルを使用できるネットワーク構成の場合は、アクセス リスト ベースのコントロール方式を使用することを検討してください。

NTP 認証が適切に設定されると、ネットワーキングデバイスは、信頼できる時刻源と同期し、信頼できる時刻源だけに同期を提供します。

特定のインターフェイス上の NTP サービス

Network Time Protocol (NTP) サービスは、デフォルトではすべてのインターフェイスで無効になっています。なんらかの NTP コマンドを入力すると、NTP がグローバルに有効になります。特定のインターフェイスを通じて特定の NTP パケットを受信しないように設定するには、インターフェイス コンフィギュレーション モードで **ntp disable** コマンドを使用します。

NTP パケットの送信元 IP アドレス

システムが NTP パケットを送信すると、通常、送信元 IP アドレスは、その NTP パケットの送信元であるインターフェイスのアドレスに設定されます。IP 送信元アドレスの取得元のインターフェイスを設定するには、グローバル コンフィギュレーション モードで **ntp source interface** コマンドを使用します。

このインターフェイスは、すべての宛先に送信されるすべてのパケットの送信元アドレスに使用されます。特定のアソシエーションに送信元アドレスを使用する場合は、**ntp peer** コマンドまたは **ntp server** コマンドで **source** キーワードを使用します。

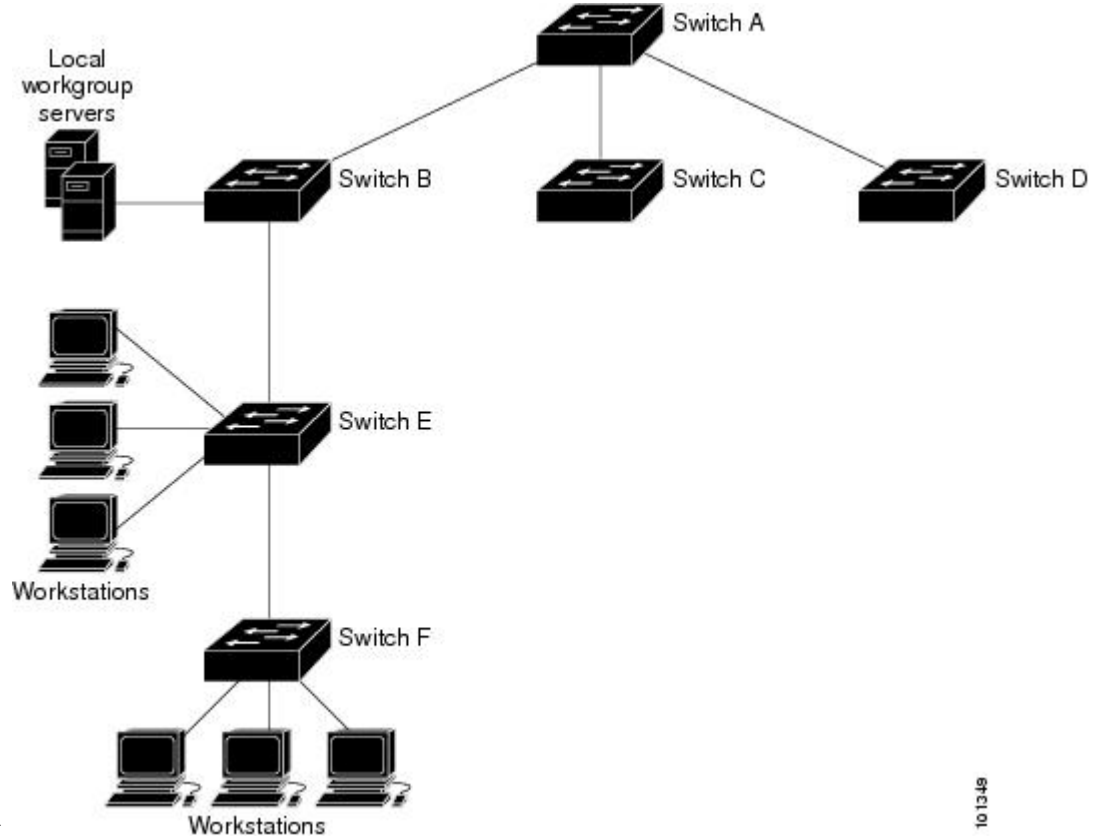
NTP の実装

NTP の実装では、ストラタム 1 サービスがサポートされないため、ラジオクロックまたは原子時計に接続できません。ネットワークの時刻サービスは、IP インターネット上のパブリック NTP サーバから取得することを推奨します。

図 2: 一般的な NTP ネットワークの構成

次の図は NTP を使用した一般的なネットワークの例を示します。スイッチ A は、スイッチ B、C、D が NTP サーバーモードに設定されている（スイッチ A との間にサーバーアソシエーションが設定されている）場合のプライマリ NTP です。スイッチ E は、アップストリームスイッ

チ (スイッチ B) とダウンストリームスイッチ (スイッチ F) の NTP ピアとして設定されま



す。

ネットワークがインターネットから切り離されている場合、NTPによって、実際には、他の方法で時刻を取得している場合でも、NTPを使用した同期化と同様にデバイスの動作を設定できます。他のデバイスは、NTPによりこのデバイスと同期化されます。

複数のタイム ソースがある場合は、NTP は常に、より信頼性があると見なされます。NTP の時刻は、他の方法による時刻に優先します。

自社のホストシステムに NTP ソフトウェアを組み込んでいるメーカーが数社あり、UNIX システム用のバージョンやその派生ソフトウェアも一般に入手できます。このソフトウェアによって、ホストシステムも時間が同期化されます。

システム名およびシステム プロンプト

デバイスを識別するシステム名を設定します。デフォルトでは、システム名およびプロンプトは *Switch* です。

システム プロンプトを設定していない場合は、システム名の最初の 20 文字がシステム プロンプトとして使用されます。大なり記号 (>) が付加されます。システム名が変更されると、プロンプトは更新されます。

ここで使用するコマンドの構文および使用方法の詳細については、『Cisco IOS Configuration Fundamentals Command Reference, Release 12.4』および『Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols, Release 12.4』を参照してください。

スタックのシステム名およびシステム プロンプト

アクティブスイッチを介してスタックメンバーにアクセスする場合は、**session stack-member-number** 特権 EXEC コマンドを使用する必要があります。スタック メンバ番号の有効範囲は 1 ~ 8 です。このコマンドを使用すると、スタック メンバの番号がシステム プロンプトの末尾に追加されます。たとえば、**Switch-2#** はスタック メンバ 2 の特権 EXEC モードのプロンプトであり、スイッチスタックのシステムプロンプトは **Switch** です。

デフォルトのシステム名とプロンプトの設定

デフォルトのスイッチのシステム名およびプロンプトは **Switch** です。

DNS

DNS プロトコルは、ドメインネームシステム (DNS) を制御します。DNS とは分散型データベースであり、ホスト名を IP アドレスにマッピングできます。device に DNS を設定すると、**ping**、**telnet**、**connect** などのすべての IP コマンドを使用する場合や、関連する Telnet サポート操作時に、IP アドレスの代わりにホスト名を使用できます。

IP によって定義される階層型の命名方式では、デバイスを場所またはドメインで特定できます。ドメイン名は、ピリオド (.) を区切り文字として使用して構成されています。たとえば、シスコは、IP で **com** というドメイン名に分類される商業組織なので、ドメイン名は **cisco.com** となります。このドメイン内の特定のデバイス、たとえばファイル転送プロトコル (FTP) システムは、**ftp.cisco.com** で表されます。

IP ではドメイン名をトラッキングするために、ドメインネームサーバという概念が定義されています。ドメインネームサーバの役割は、名前から IP アドレスへのマッピングをキャッシュ (またはデータベース) に保存することです。ドメイン名を IP アドレスにマッピングするには、まず、ホスト名を明示し、ネットワーク上に存在するネームサーバを指定し、DNS をイネーブルにします。

DNS のデフォルト設定値

表 1: DNS のデフォルト設定値

機能	デフォルト設定
DNS イネーブル ステート	イネーブル
DNS デフォルト ドメイン名	未設定
DNS サーバ	ネームサーバのアドレスが未設定

ログインバナー

Message-of-The-Day (MoTD) バナーおよびログインバナーを作成できます。MoTD バナーはログイン時に、接続されたすべての端末に表示されます。すべてのネットワークユーザに影響するメッセージ（差し迫ったシステム シャットダウンの通知など）を送信する場合に便利です。

ログインバナーも接続されたすべての端末に表示されます。表示されるのは、MoTDバナーの後で、ログインプロンプトが表示される前です。



(注) ここで使用するコマンドの構文および使用方法の詳細については、『Cisco IOS Configuration Fundamentals Command Reference, Release 12.4』を参照してください。

バナーのデフォルト設定

MoTD およびログインバナーは設定されません。

MAC アドレス テーブル

MAC アドレス テーブルには、deviceがポート間のトラフィック転送に使用するアドレス情報が含まれています。このアドレス テーブルに登録されたすべての MAC アドレスは、1 つまたは複数のポートに対応しています。アドレス テーブルに含まれるアドレス タイプには、次のものがあります。

- ダイナミックアドレス：deviceが取得し、使用されなくなった時点で期限切れとなる送信元の MAC アドレス
- スタティックアドレス：手動で入力され、期限切れにならず、deviceのリセット時にも消去されないユニキャスト アドレス

アドレス テーブルは、宛先 MAC アドレス、対応する VLAN（仮想 LAN）ID、アドレスに対応付けられたポート番号、およびタイプ（スタティックまたはダイナミック）のリストです。



(注) ここで使用するコマンドの構文および使用方法の詳細については、このリリースに対応するコマンドリファレンスを参照してください。

MAC アドレス テーブルの作成

すべてのポートでサポートされる複数の MAC アドレスを使用して、他のネットワーク デバイスにdevice上のすべてのポートを接続できます。deviceは、各ポートで受信するパケットの送信元アドレスを取得し、アドレス テーブルにアドレスとそれに関連付けられたポート番号を追加することによって、動的なアドレス指定を行います。ネットワークでデバイスの追加または削

除が行われると、**device**によってアドレステーブルが更新され、新しいダイナミックアドレスが追加され、使用されていないアドレスは期限切れになります。

エージングインターバルは、グローバルに設定されています。ただし、**device**はVLANごとにアドレステーブルを維持し、STPによってVLAN単位で有効期間を短縮できます。

deviceは、受信したパケットの宛先アドレスに基づいて、任意の組み合わせのポート間でパケットを送信します。**device**は、MACアドレステーブルを使用することによって、宛先アドレスに関連付けられたポートに限定してパケットを転送します。宛先アドレスがパケットを送信したポート上にある場合は、パケットはフィルタリング処理され、転送されません。**device**は、常にストアアンドフォワード方式を使用します。このため、完全なパケットをいったん保存してエラーがないか検査してから転送します。

MAC アドレスおよび VLAN

すべてのアドレスはVLANと関連付けられます。1つのアドレスを複数のVLANに対応付け、それぞれで異なる宛先を設定できます。たとえば、ユニキャストアドレスをVLAN 1のポート1およびVLAN 5のポート9、10、1に転送するといったことが可能です。

VLANごとに、独自の論理アドレステーブルが維持されます。あるVLANで認識されているアドレスが別のVLANで認識されるには、別のVLAN内のポートによって学習されるか、または別のVLAN内のポートにスタティックに対応付けられる必要があります。

MAC アドレスおよびデバイスのスタック

すべてのスタックメンバにあるMACアドレステーブルでは、同期が取られます。いかなる時点でも、各スタックメンバには、各VLANのアドレステーブルの同じコピーがあります。アドレスがエージングアウトすると、アドレスは、すべてのスタックメンバにあるアドレステーブルから削除されます。デバイスがスイッチスタックに参加すると、そのデバイスでは、他のスタックメンバで学習された各VLANのアドレスを受信します。スタックメンバがスイッチスタックに残っているときには、残りのスタックメンバは、エージングアウトするか、前のスタックメンバによってラーニングされたすべてのアドレスが削除されます。

MAC アドレス テーブルのデフォルト設定

次の表に、MACアドレステーブルのデフォルト設定を示します。

表 2: MAC アドレスのデフォルト設定

機能	デフォルト設定
エージング タイム	300 秒
ダイナミック アドレス	自動学習
スタティック アドレス	未設定

ARP テーブルの管理

デバイスと通信するには（イーサネット上のデバイスなど）、ソフトウェアは最初にそのデバイスの 48 ビット MAC アドレスまたはローカル データ リンク アドレスを学習する必要があります。IP アドレスからローカル データ リンク アドレスを学習するプロセスを、アドレス解決といいます。

アドレス解決プロトコル（ARP）は、ホスト IP アドレスを、該当するメディアまたは MAC アドレスおよび VLAN ID に対応付けます。IP アドレスを使用して、ARP は対応する MAC アドレスを見つけます。MAC アドレスが見つかると、IP と MAC アドレスとの対応を ARP キャッシュに格納し、すばやく検索できるようにします。その後、IP データグラムがリンク層フレームにカプセル化され、ネットワークを通じて送信されます。イーサネット以外の IEEE 802 ネットワークにおける IP データグラムのカプセル化および ARP 要求/応答については、サブネットワーク アクセス プロトコル（SNAP）で規定されています。IP インターフェイスでは、標準的なイーサネット形式の ARP カプセル化（**arpa** キーワードで表される）がデフォルトでイーサネットに設定されています。

手動でテーブルに追加された ARP エントリは期限切れにならないので、手動で削除する必要があります。

CLI（コマンドライン インターフェイス）の手順については、*Cisco.com* で Cisco IOS Release 12.4 のマニュアルを参照してください。

デバイスを管理する方法

手動による日付と時刻の設定

正確なシステム時刻は再開と再起動により保持されますが、日付と時刻はシステムが再開してから手動で設定できます。

手動設定は必要な場合にのみ使用することを推奨します。**device**が同期できる外部ソースがある場合は、システムクロックを手動で設定する必要はありません。

システムクロックの設定

ネットワーク上に、NTP サーバなどの時刻サービスを提供する外部ソースがある場合、手動でシステムクロックを設定する必要はありません。

システムクロックを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. 次のいずれかを使用します。
 - **clock set** *hh:mm:ss day month year*
 - **clock set** *hh:mm:ss month day year*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <p>デバイス> enable</p>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>次のいずれかを使用します。</p> <ul style="list-style-type: none"> • clock set hh:mm:ss day month year • clock set hh:mm:ss month day year <p>例 :</p> <p>デバイス# clock set 13:32:00 23 March 2013</p>	<p>次のいずれかの書式を使ってシステムクロックを手動で設定します。</p> <ul style="list-style-type: none"> • hh:mm:ss : 時間 (24 時間形式)、分、秒を指定します。指定された時刻は、設定されたタイムゾーンに基づきます。 • day : 月の日で日付を指定します。 • month : 月を名前で指定します。 • year : 年を指定します (略式表記で指定しないでください)。

タイムゾーンの設定

タイムゾーンを手動で設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **clock timezone zone hours-offset [minutes-offset]**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <p>デバイス> enable</p>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p>	<p>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。</p>

	コマンドまたはアクション	目的
	デバイス# <code>configure terminal</code>	
ステップ 3	<p>clock timezone zone hours-offset [minutes-offset]</p> <p>例 :</p> <p>デバイス(config)# <code>clock timezone AST -3 30</code></p>	<p>時間帯を設定します。</p> <p>内部時間は、協定世界時 (UTC) で維持されるため、このコマンドは表示専用で、時刻を手動で設定するときだけに使用されます。</p> <ul style="list-style-type: none"> • <i>zone</i> : 標準時が適用されているときに表示されるタイムゾーンの名前を入力します。デフォルトは UTC です。 • <i>hours-offset</i> : UTC からのオフセット時間数を入力します。 • (任意) <i>minutes-offset</i> : UTC からのオフセット分数を入力します。ローカルタイムゾーンが UTC と 1 時間の差の割合である場合に指定できます。
ステップ 4	<p>end</p> <p>例 :</p> <p>デバイス(config)# <code>end</code></p>	特権 EXEC モードに戻ります。
ステップ 5	<p>show running-config</p> <p>例 :</p> <p>デバイス# <code>show running-config</code></p>	入力を確認します。
ステップ 6	<p>copy running-config startup-config</p> <p>例 :</p> <p>デバイス# <code>copy running-config startup-config</code></p>	(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。

夏時間の設定

毎年特定の日に夏時間が開始および終了する地域に夏時間を設定するには、次の作業を行います。

手順の概要

1. enable

2. **configure terminal**
3. **clock summer-time zone date date month year hh:mm date month year hh:mm [offset]**
4. **clock summer-time zone recurring [week day month hh:mm week day month hh:mm [offset]]**
5. **end**
6. **show running-config**
7. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 ・パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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	clock summer-time zone date date month year hh:mm date month year hh:mm [offset] 例： デバイス(config)# clock summer-time PDT date 10 March 2013 2:00 3 November 2013 2:00	毎年指定された日に開始および終了する夏時間を設定します。
ステップ 4	clock summer-time zone recurring [week day month hh:mm week day month hh:mm [offset]] 例： デバイス(config)# clock summer-time PDT recurring 10 March 2013 2:00 3 November 2013 2:00	毎年指定された日に開始および終了する夏時間を設定します。すべての時刻は、現地のタイムゾーンを基準にしています。開始時間は標準時を基準にしています。 終了時間は夏時間を基準にしています。夏時間はデフォルトでディセーブルに設定されています。パラメータなしで clock summer-time zone recurring を指定すると、夏時間のルールは米国のルールにデフォルト設定されます。 開始月が終了月より後の場合は、システムでは南半球にいると見なされます。 ・ <i>zone</i> : 夏時間が有効な場合に表示される時間帯名 (PDT など) を指定します。 ・ (任意) <i>week</i> : 月の週 (1 ~ 4、 first 、または last) を指定します。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • (任意) <i>day</i> : 曜日 (Sunday、Monday など) を指定します。 • (任意) <i>month</i> : 月 (January、February など) を指定します。 • (任意) <i>hh:mm</i> : 時および分単位で時間 (24 時間形式) を指定します。 • (任意) <i>offset</i> : 夏時間中に追加する分数を指定します。デフォルトは 60 です。
ステップ 5	end 例 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 6	show running-config 例 : デバイス # show running-config	入力を確認します。
ステップ 7	copy running-config startup-config 例 : デバイス # 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。

ユーザーの居住地の夏時間が定期的なパターンに従わない (次の夏時間のイベントの正確な日時を設定する) 場合は、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **clock summer-time zone date [month date year hh:mm month date year hh:mm [offset]] or clock summer-time zone date [date month year hh:mm date month year hh:mm [offset]]**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <p>デバイス> enable</p>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <p>デバイス# configure terminal</p>	<p>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。</p>
ステップ 3	<p>clock summer-time zone date [<i>month date year hh:mm month date year hh:mm [offset]</i>] or clock summer-time zone date [<i>date month year hh:mm date month year hh:mm [offset]</i>]</p>	<p>最初の日付で夏時間開始の日付を、2 番目の日付で終了の日付を設定します。</p> <p>夏時間はデフォルトでディセーブルに設定されています。</p> <ul style="list-style-type: none"> • <i>zone</i> には、夏時間が施行されているときに表示されるタイムゾーンの名前 (たとえば PDT) を入力します。 • (任意) <i>week</i> には、月の何週目かを指定します (1 ~ 5、または last)。 • (任意) <i>day</i> には、曜日を指定します (Sunday、Monday など)。 • (任意) <i>month</i> には、月を指定します (January、February など)。 • (任意) <i>hh:mm</i> には、時刻を時間 (24 時間形式) と分で指定します。 • (任意) <i>offset</i> には、夏時間の間、追加する分数を指定します。デフォルトは 60 です。
ステップ 4	<p>end</p> <p>例 :</p> <p>デバイス (config) # end</p>	<p>特権 EXEC モードに戻ります。</p>
ステップ 5	<p>show running-config</p> <p>例 :</p>	<p>入力を確認します。</p>

	コマンドまたはアクション	目的
	デバイス# <code>show running-config</code>	
ステップ 6	copy running-config startup-config 例： デバイス# <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。

NTP の設定

デバイスはハードウェアサポートクロックを備えておらず、外部 NTP ソースが利用できないときに、ピアが自身を同期化するための NTP プライマリクロックとして機能することはできません。デバイスは、カレンダーに対するハードウェアサポートも備えていません。そのため、グローバル コンフィギュレーションモードで `ntp update-calendar` コマンドと `ntp master` コマンドを使用することはできません。

NTP の設定情報については、次のセクションを参照してください。

NTP のデフォルト設定

NTP のデフォルト設定を示します。

表 3: NTP のデフォルト設定

機能	デフォルト設定
NTP 認証	ディセーブル認証キーは指定されていません。
NTP ピアまたはサーバー アソシエーション	未設定
NTP ブロードキャスト サービス	ディセーブル。どのインターフェイスも NTP ブロードキャストパケットを送受信しません。
NTP アクセス制限	アクセスコントロールは指定されていません。
NTP パケット送信元 IP アドレス	送信元アドレスは、発信インターフェイスによって設定されます。

NTP は、すべてのインターフェイスでデフォルトでイネーブルに設定されています。すべてのインターフェイスは、NTP パケットを受信します。

NTP 認証の設定

NTP 認証を設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **[no] ntp authenticate**
4. **[no] ntp authentication-key number md5 value**
5. **[no] ntp trusted-key key-number**
6. **[no] ntp server ip-address key key-id [prefer]**
7. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# 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	[no] ntp authenticate 例： Device(config)# ntp authenticate	NTP 認証をイネーブルにします。 NTP 認証を無効にするには、このコマンドの no 形式を使用します。
ステップ 4	[no] ntp authentication-key number md5 value 例： Device(config)# ntp authentication-key 42 md5 aNiceKey	認証キーを定義します。 • キーごとに、キー番号、タイプ、および値を 1 つずつ指定します。 SNTP の認証キーを削除する場合は、このコマンドの no 形式を使用します。
ステップ 5	[no] ntp trusted-key key-number 例： Device(config)# ntp trusted-key 42	このデバイスと同期できるようにするために、ピア NTP デバイスが NTP パケットで提供する必要がある信頼できる認証キーを定義します。 信頼できる認証を無効にするには、このコマンドの no 形式を使用します。
ステップ 6	[no] ntp server ip-address key key-id [prefer] 例：	NTP タイム サーバーによってソフトウェア クロックが同期されるように設定します。

	コマンドまたはアクション	目的
	Device(config)# ntp server 172.16.22.44 key 42	<ul style="list-style-type: none"> • ip-address : クロック同期を提供するタイムサーバーの IP アドレス。 • key-id : ntp authentication-key コマンドで定義された認証キー。 • prefer : このピアを、同期を提供する優先ピアにします。このキーワードは、ピア間のクロックホップを減らします。 <p>サーバーアソシエーションを解除するには、このコマンドの no 形式を入力します。</p>
ステップ 7	end 例 : Device(config)# end	グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。

ポーリングベースの NTP アソシエーションの設定

ポーリングベースの NTP アソシエーションを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **[no] ntp peer ip-address [version number] [key key-id] [source interface] [prefer]**
4. **[no] ntp server ip-address [version number] [key key-id] [source interface] [prefer]**
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します (要求された場合) 。
ステップ 2	configure terminal 例 : Device# 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。

	コマンドまたはアクション	目的
<p>ステップ 3</p>	<p>[no] ntp peer ip-address [version number] [key key-id] [source interface] [prefer]</p> <p>例 :</p> <pre>Device(config)# ntp peer 172.16.22.44 version 2</pre>	<p>ピアを同期化するか、またはピアによって同期化されるように、デバイスのシステムクロックを設定します (ピアアソシエーション)。</p> <ul style="list-style-type: none"> • <i>ip-address</i> : クロック同期を提供する、またはクロック同期を提供されるピアの IP アドレス。 • <i>number</i> : NTP バージョン番号。指定できる範囲は 1 ~ 4 です。デフォルトでは、バージョン 4 が選択されています。 • <i>key-id</i> : ntp authentication-key コマンドで定義された認証キー。 • <i>interface</i> : IP の送信元アドレスを取得するインターフェイス。デフォルトでは、送信元 IP アドレスは発信インターフェイスから取得します。 • prefer : このピアを、同期を提供する優先ピアにします。このキーワードにより、ピア間の切り替えが減少します。 <p>ピアアソシエーションを解除するには、このコマンドの no 形式を使用します。</p>
<p>ステップ 4</p>	<p>[no] ntp server ip-address [version number] [key key-id] [source interface] [prefer]</p> <p>例 :</p> <pre>Device(config)# ntp server 172.16.22.44 version 2</pre>	<p>タイムサーバーによって同期化されるように、デバイスのシステムクロックを設定します (サーバーアソシエーション)。</p> <ul style="list-style-type: none"> • <i>ip-address</i> : クロック同期を提供するタイムサーバーの IP アドレス。 • <i>number</i> : NTP バージョン番号。指定できる範囲は 1 ~ 4 です。デフォルトでは、バージョン 4 が選択されています。 • <i>key-id</i> : ntp authentication-key コマンドで定義された認証キー。 • <i>interface</i> : IP の送信元アドレスを取得するインターフェイス。デフォルトでは、送信元 IP アドレスは発信インターフェイスから取得します。 • prefer : このピアを、同期を提供する優先ピアにします。このキーワードは、ピア間のクロックホップを減らします。 <p>サーバーアソシエーションを解除するには、このコマンドの no 形式を入力します。</p>

	コマンドまたはアクション	目的
ステップ 5	end 例 : Device (config) # end	特権 EXEC モードに戻ります。

ブロードキャストベースの NTP アソシエーションの設定

ブロードキャストベースの NTP アソシエーションを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface interface-id**
4. **[no] ntp broadcast [version number] [key key-id] [destination-address]**
5. **[no] ntp broadcast client**
6. **exit**
7. **[no] ntp broadcastdelay microseconds**
8. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 : Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例 : Device# 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	interface interface-id 例 : Device (config) # interface gigabitethernet1/0/1	インターフェイスを設定し、インターフェイス コンフィギュレーション モードを開始します。
ステップ 4	[no] ntp broadcast [version number] [key key-id] [destination-address] 例 :	NTPブロードキャストパケットをピアに送信するインターフェイスをイネーブルにします。

	コマンドまたはアクション	目的
	<pre>Device(config-if)# ntp broadcast version 2</pre>	<ul style="list-style-type: none"> • <i>number</i> : NTPバージョン番号。指定できる範囲は1～4です。デフォルトでは、バージョン4が使用されます。 • <i>key-id</i> : 認証キー。 • <i>destination-address</i> : このスイッチに対してクロックを同期しているピアのIPアドレス。 <p>インターフェイスでのNTPブロードキャストパケットの送信を無効にするには、このコマンドの no 形式を使用します。</p>
ステップ 5	<p>[no] ntp broadcast client</p> <p>例 :</p> <pre>Device(config-if)# ntp broadcast client</pre>	<p>インターフェイスがNTPブロードキャストパケットを受信できるようにします。</p> <p>インターフェイスでのNTPブロードキャストパケットの受信を無効にするには、このコマンドの no 形式を使用します。</p>
ステップ 6	<p>exit</p> <p>例 :</p> <pre>Device(config-if)# exit</pre>	<p>特権 EXEC モードに戻ります。</p>
ステップ 7	<p>[no] ntp broadcastdelay <i>microseconds</i></p> <p>例 :</p> <pre>Device(config)# ntp broadcastdelay 100</pre>	<p>(任意) デバイスとNTPブロードキャストサーバー間のラウンドトリップ遅延の予測値を変更します。</p> <p>デフォルトは3000マイクロ秒です。範囲は1～999999です。</p> <p>インターフェイスでのNTPブロードキャストパケットの受信を無効にするには、このコマンドの no 形式を使用します。</p>
ステップ 8	<p>end</p> <p>例 :</p> <pre>Device(config)# end</pre>	<p>特権 EXEC モードに戻ります。</p>

NTP アクセス制限の設定

以降で説明するように、2つのレベルでNTPアクセスを制御できます。

アクセスグループの作成と基本 IP アクセスリストの割り当て

アクセスグループを作成して基本 IP アクセスリストを割り当てるには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **[no] ntp access-group {query-only | serve-only | serve | peer} access-list-number**
4. **access-list access-list-number permit source [source-wildcard]**
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# 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	[no] ntp access-group {query-only serve-only serve peer} access-list-number 例： Device(config)# ntp access-group peer 99	アクセスグループを作成し、基本 IP アクセスリストを割り当てます。 <ul style="list-style-type: none"> • query-only : NTP 制御クエリ。 • serve-only : 時間要求。 • serve : 時刻要求と NTP 制御クエリは許可しますが、リモートデバイスに対するデバイスの同期化は許可しません。 • peer : 時刻要求と NTP 制御クエリ、およびリモートデバイスに対するデバイスの同期化を許可します。 • access-list-number : IP アクセスリスト番号。指定できる範囲は 1 ~ 99 です。 スイッチ NTP サービスに対するアクセス制御を削除するには、このコマンドの no 形式を使用します。

	コマンドまたはアクション	目的
ステップ 4	<p>access-list <i>access-list-number</i> permit <i>source</i> [<i>source-wildcard</i>]</p> <p>例 :</p> <pre>Device (config)# access-list 99 permit 172.20.130.5</pre>	<p>アクセス リストを作成します。</p> <ul style="list-style-type: none"> • <i>access-list-number</i> : IP アクセスリスト番号。指定できる範囲は 1 ~ 99 です。 • permit : 条件が一致した場合にアクセスを許可します。 • <i>source</i> : デバイスへのアクセスが許可されているデバイスの IP アドレス。 • <i>source-wildcard</i> : 送信元アドレスに適用されるワイルドカードビット。 <p>(注) アクセス リストを作成する際は、アクセス リストの末尾に暗黙の deny ステートメントがデフォルトで存在し、ACL の終わりに到達するまで一致が見つからなかったすべてのパケットに適用されることに注意してください。</p> <p>SNTP の認証キーを削除する場合は、このコマンドの no 形式を使用します。</p>
ステップ 5	<p>end</p> <p>例 :</p> <pre>Device (config)# end</pre>	<p>特権 EXEC モードに戻ります。</p>

特定のインターフェイス上の NTP サービスのディセーブル化

インターフェイスで NTP パケットの受信を無効にするには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **interface** *interface-id*
4. [**no**] **ntp disable**
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# 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	interface interface-id 例： Device(config)# interface gigabitethernet1/0/1	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 4	[no] ntp disable 例： Device(config-if)# ntp disable	インターフェイスでNTPパケットの受信をディセーブルにします。 インターフェイスでNTPパケットの受信を再度有効にするには、このコマンドの no 形式を使用します。
ステップ 5	end 例： Device(config-if)# end	特権 EXEC モードに戻ります。

システム名の設定

システム名を手動で設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **hostname name**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 ・パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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	hostname name 例： デバイス (config)# hostname remote-users	システム名を設定します。システム名を設定すると、システム プロンプトとしても使用されます。 デフォルト設定は Switch です。 名前は ARPANET ホスト名のルールに従う必要があります。このルールではホスト名は文字で始まり、文字または数字で終わり、その間には文字、数字、またはハイフンしか使用できません。名前には 63 文字まで使用できます。
ステップ 4	end 例： remote-users (config)# end remote-users#	特権 EXEC モードに戻ります。
ステップ 5	show running-config 例： デバイス# show running-config	入力を確認します。
ステップ 6	copy running-config startup-config 例： デバイス# 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。

DNS の設定

deviceの IP アドレスをホスト名として使用する場合、この IP アドレスが使用されるため、DNS クエリは発生しません。ピリオド (.) なしでホスト名を設定すると、ピリオドと、それに続く

デフォルトのドメイン名がホスト名に追加され、その後で DNS クエリーが行われ、名前を IP アドレスにマッピングします。デフォルトのドメイン名は、**ip domain-name** グローバル コンフィギュレーション コマンドによって設定される値です。ホスト名にピリオド (.) がある場合は、Cisco IOS ソフトウェアは、ホスト名にデフォルトのドメイン名を追加せずに IP アドレスを検索します。

DNS を使用するようにスイッチを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip domain-name name**
4. **ip name-server server-address1 [server-address2 ... server-address6]**
5. **ip domain-lookup [nsap | source-interface interface]**
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <p>デバイス> enable</p>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <p>デバイス# configure terminal</p>	<p>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。</p>
ステップ 3	<p>ip domain-name name</p> <p>例 :</p> <p>デバイス(config)# ip domain-name Cisco.com</p>	<p>非完全修飾ホスト名 (ドット付き 10 進表記ドメイン名のない名前) を完成させるためにソフトウェアが使用する、デフォルトのドメイン名を定義します。</p> <p>ドメイン名を未修飾の名前から区切るために使用される最初のピリオドは入れないでください。</p> <p>ブート時にはドメイン名は設定されていませんが、device の設定が BOOTP または Dynamic Host Configuration Protocol (DHCP) サーバーから行われている場合、BOOTP または DHCP サーバーによってデフォルトのドメイン名が設定されることがあります (この情報がサーバーに設定されている場合)。</p>

	コマンドまたはアクション	目的
ステップ 4	<p>ip name-server <i>server-address1</i> [<i>server-address2</i> ... <i>server-address6</i>]</p> <p>例 :</p> <pre>デバイス(config)# ip name-server 192.168.1.100 192.168.1.200 192.168.1.300</pre>	<p>名前とアドレスの解決に使用する 1 つまたは複数のネーム サーバのアドレスを指定します。</p> <p>最大 6 つのネーム サーバを指定できます。各サーバアドレスはスペースで区切ります。最初に指定されたサーバが、プライマリ サーバです。device は、最初にプライマリ サーバへ DNS クエリを送信します。そのクエリが失敗した場合は、バックアップサーバにクエリが送信されます。</p>
ステップ 5	<p>ip domain-lookup [<i>nsap</i> source-interface <i>interface</i>]</p> <p>例 :</p> <pre>デバイス(config)# ip domain-lookup</pre>	<p>(任意) device 上で、DNS に基づくホスト名からアドレスへの変換をイネーブルにします。この機能は、デフォルトでイネーブルにされています。</p> <p>ユーザのネットワークデバイスが、名前の割り当てを制御できないネットワーク内のデバイスと接続する必要がある場合、グローバルなインターネットのネーミング方式 (DNS) を使用して、ユーザのデバイスを一意に識別するデバイス名を動的に割り当てることができます。</p>
ステップ 6	<p>end</p> <p>例 :</p> <pre>デバイス(config)# end</pre>	<p>特権 EXEC モードに戻ります。</p>
ステップ 7	<p>show running-config</p> <p>例 :</p> <pre>デバイス# show running-config</pre>	<p>入力を確認します。</p>
ステップ 8	<p>copy running-config startup-config</p> <p>例 :</p> <pre>デバイス# copy running-config startup-config</pre>	<p>(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。</p>

Message-of-the-Day ログインバナーの設定

device にログインしたときに画面に表示される 1 行以上のメッセージバナーを作成できます。

MOTD ログインバナーを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **banner motd c message c**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <p>デバイス> enable</p>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <p>デバイス# configure terminal</p>	<p>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。</p>
ステップ 3	<p>banner motd c message c</p> <p>例 :</p> <p>デバイス (config)# banner motd # This is a secure site. Only authorized users are allowed. For access, contact technical support. #</p>	<p>MoTD を指定します。</p> <p><i>c</i> : ポンド記号 (#) など、目的のデリミタを入力して Return キーを押します。区切り文字はバナー テキストの始まりと終わりを表します。終わりの区切り文字の後ろの文字は廃棄されます。</p> <p><i>message</i> : 255 文字までのバナーメッセージを入力します。メッセージ内には区切り文字を使用できません。</p>
ステップ 4	<p>end</p> <p>例 :</p> <p>デバイス (config)# end</p>	<p>特権 EXEC モードに戻ります。</p>
ステップ 5	<p>show running-config</p> <p>例 :</p> <p>デバイス# show running-config</p>	<p>入力を確認します。</p>

	コマンドまたはアクション	目的
ステップ 6	copy running-config startup-config 例： デバイス# 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。

ログインバナーの設定

接続されたすべての端末でログインバナーが表示されるように設定できます。バナーが表示されるのは、MoTD バナーの後で、ログインプロンプトが表示される前です。

ログインバナーを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **banner login c message c**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例： デバイス# 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	banner login c message c 例： デバイス(config)# banner login \$ Access for authorized users only. Please enter your username and password.	ログイン メッセージを指定します。 c : ポンド記号 (#) など、目的のデリミタを入力して Return キーを押します。区切り文字はバナー テキストの始まりと終わりを表します。終わりの区切り文字の後ろの文字は廃棄されます。

	コマンドまたはアクション	目的
	\$	<i>message</i> : 255 文字までのログインメッセージを入力します。メッセージ内には区切り文字を使用できません。
ステップ 4	end 例 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 5	show running-config 例 : デバイス # show running-config	入力を確認します。
ステップ 6	copy running-config startup-config 例 : デバイス # 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。

MAC アドレス テーブルの管理

アドレス エージング タイムの変更

ダイナミックアドレステーブルのエージングタイムを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **mac address-table aging-time [0 | 10-1000000] [routed-mac | vlan vlan-id]**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 :	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。

	コマンドまたはアクション	目的
	デバイス> <code>enable</code>	
ステップ 2	configure terminal 例： デバイス# <code>configure terminal</code>	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	mac address-table aging-time [0 10-1000000] [routed-mac vlan vlan-id] 例： デバイス(config)# <code>mac address-table aging-time 500 vlan 2</code>	ダイナミック エントリが使用または更新された後、MAC アドレス テーブル内に保持される時間を設定します。 指定できる範囲は 10 ~ 1000000 秒です。デフォルトは 300 です。0 を入力して期限切れをディセーブルにすることもできます。スタティック アドレスは、期限切れになることもテーブルから削除されることはありません。 <i>vlan-id</i> : 有効な ID は 1 ~ 4094 です。
ステップ 4	end 例： デバイス(config)# <code>end</code>	特権 EXEC モードに戻ります。
ステップ 5	show running-config 例： デバイス# <code>show running-config</code>	入力を確認します。
ステップ 6	copy running-config startup-config 例： デバイス# <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。

MAC アドレス変更通知トラップの設定

NMSホストにMACアドレス変更通知トラップを送信するようにスイッチを設定するには、次の手順を実行します。

手順の概要

1. `enable`

2. **configure terminal**
3. **snmp-server host** *host-addr community-string notification-type* { **informs** | **traps** } {**version** {**1** | **2c** | **3**}} { **vrf** *vrf instance name*}
4. **snmp-server enable traps mac-notification change**
5. **mac address-table notification change**
6. **mac address-table notification change** [**interval** *value*] [**history-size** *value*]
7. **interface** *interface-id*
8. **snmp trap mac-notification change** {**added** | **removed**}
9. **end**
10. **show running-config**
11. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>デバイス> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <pre>デバイス# configure terminal</pre>	<p>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。</p>
ステップ 3	<p>snmp-server host <i>host-addr community-string notification-type</i> { informs traps } {version {1 2c 3}} { vrf <i>vrf instance name</i>}</p> <p>例 :</p> <pre>デバイス(config)# snmp-server host 172.20.10.10 traps private mac-notification</pre>	<p>トラップ メッセージの受信側を指定します。</p> <ul style="list-style-type: none"> • host-addr : NMS の名前またはアドレスを指定します。 • traps (デフォルト) : ホストに SNMP トラップを送信します。 • informs : ホストに SNMP 情報を送信します。 • version : サポートする SNMP バージョンを指定します。informs にはバージョン 1 (デフォルト) を使用できません。 • community-string : 通知処理で送信する文字列を指定します。snmp-server host コマンドを使用してこの文字列を設定できますが、この文字列を定義するには、snmp-server community 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から、snmp-server host コマンドを使用することを推奨します。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • <i>notification-type</i> : mac-notification キーワードを使用します。 • <i>vrf vrf</i> インスタンス名 : このホストの VPN ルーティング/転送インスタンスを指定します。
ステップ 4	snmp-server enable traps mac-notification change 例 : デバイス (config) # snmp-server enable traps mac-notification change	device が MAC アドレス変更通知トラップを送信できるようにします。
ステップ 5	mac address-table notification change 例 : デバイス (config) # mac address-table notification change	MAC アドレス変更通知機能をイネーブルにします。
ステップ 6	mac address-table notification change [interval value] [history-size value] 例 : デバイス (config) # mac address-table notification change interval 123 デバイス (config) # mac address-table notification change history-size 100	トラップ インターバル タイムと履歴テーブルのサイズを入力します。 <ul style="list-style-type: none"> • (任意) interval value : NMS に生成されるトラップの各セット間の通知トラップインターバルを秒単位で指定します。指定できる範囲は 0 ~ 2147483647 秒です。デフォルトは 1 秒です。 • (任意) history-size value : MAC 通知履歴テーブルの最大エントリ数を指定します。指定できる範囲は 0 ~ 500 です。デフォルトは 1 です。
ステップ 7	interface interface-id 例 : デバイス (config) # interface gigabitethernet1/0/2	インターフェイス コンフィギュレーション モードを開始し、SNMP MAC アドレス通知トラップをイネーブルにするレイヤ 2 インターフェイスを指定します。
ステップ 8	snmp trap mac-notification change {added removed} 例 : デバイス (config-if) # snmp trap mac-notification change added	インターフェイス上で MAC アドレス変更通知トラップをイネーブルにします。 <ul style="list-style-type: none"> • MAC アドレスがインターフェイスに added された場合にトラップをイネーブルにします。 • MAC アドレスがインターフェイスに removed された場合にトラップをイネーブルにします。

	コマンドまたはアクション	目的
ステップ 9	end 例： デバイス(config)# end	特権 EXEC モードに戻ります。
ステップ 10	show running-config 例： デバイス# show running-config	入力を確認します。
ステップ 11	copy running-config startup-config 例： デバイス# 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。

MAC アドレス移動通知トラップの設定

MAC 移動通知を設定する場合は、MAC アドレスが、同じ VLAN 内のあるポートから別のポートに移動すると常に、SNMP 通知が生成されてネットワーク管理システムに送信されます。

次の手順に従い、device を設定し、NMS ホストに MAC アドレス移動通知トラップを送信するようにします。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server host host-addr {traps | informs} {version {1 | 2c | 3}} community-string notification-type**
4. **snmp-server enable traps mac-notification move**
5. **mac address-table notification mac-move**
6. **end**
7. **show running-config**
8. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例：	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。

	コマンドまたはアクション	目的
	<pre>デバイス> enable</pre>	
ステップ 2	<p>configure terminal</p> <p>例 :</p> <pre>デバイス# configure terminal</pre>	<p>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。</p>
ステップ 3	<p>snmp-server host host-addr {traps informs} {version {1 2c 3}} community-string notification-type</p> <p>例 :</p> <pre>デバイス(config)# snmp-server host 172.20.10.10 traps private mac-notification</pre>	<p>トラップ メッセージの受信側を指定します。</p> <ul style="list-style-type: none"> • host-addr : NMS の名前またはアドレスを指定します。 • traps (デフォルト) : ホストに SNMP トラップを送信します。 • informs : ホストに SNMP 情報を送信します。 • version : サポートする SNMP バージョンを指定します。informs にはバージョン 1 (デフォルト) を使用できません。 • community-string : 通知処理で送信する文字列を指定します。snmp-server host コマンドを使用してこの文字列を設定できますが、この文字列を定義するには、snmp-server community 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から、snmp-server host コマンドを使用することを推奨します。 • notification-type : mac-notification キーワードを使用します。
ステップ 4	<p>snmp-server enable traps mac-notification move</p> <p>例 :</p> <pre>デバイス(config)# snmp-server enable traps mac-notification move</pre>	<p>deviceが NMS に MAC アドレス移動通知トラップを送信できるようにします。</p>
ステップ 5	<p>mac address-table notification mac-move</p> <p>例 :</p> <pre>デバイス(config)# mac address-table notification mac-move</pre>	<p>MAC アドレス移動通知機能をイネーブルにします。</p>

	コマンドまたはアクション	目的
ステップ 6	end 例： デバイス(config)# end	特権 EXEC モードに戻ります。
ステップ 7	show running-config 例： デバイス# show running-config	入力を確認します。
ステップ 8	copy running-config startup-config 例： デバイス# 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。

次のタスク

MAC アドレス移動通知トラップの送信をディセーブルにするには、**no snmp-server enable traps mac-notification move** 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。MAC アドレス移動通知機能をディセーブルにするには、**no mac address-table notification mac-move** 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。

設定を確認するには、**show mac address-table notification mac-move** 特権 EXEC コマンドを入力します。

MAC しきい値通知トラップの設定

MAC しきい値通知を設定する場合は、MAC アドレス テーブルのしきい値の制限値に達するか、その値を超えると、SNMP通知が生成されてネットワーク管理システムに送信されます。

NMS ホストに MAC アドレス テーブルしきい値通知トラップを送信するようにスイッチを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **snmp-server host host-addr { traps / informs } { version { 1 | 2c | 3 } } community-string notification-type**
4. **snmp-server enable traps mac-notification threshold**
5. **mac address-table notification threshold**
6. **mac address-table notification threshold [limit percentage] [interval time]**
7. **end**

- 8. show running-config
- 9. copy running-config startup-config

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>デバイス> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <pre>デバイス# configure terminal</pre>	<p>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。</p>
ステップ 3	<p>snmp-server host <i>host-addr</i> { traps / informs } { version { 1 2c 3 } } <i>community-string</i> <i>notification-type</i></p> <p>例 :</p> <pre>デバイス(config)# snmp-server host 172.20.10.10 traps private mac-notification</pre>	<p>トラップ メッセージの受信側を指定します。</p> <ul style="list-style-type: none"> • host-addr : NMS の名前またはアドレスを指定します。 • traps (デフォルト) : ホストに SNMP トラップを送信します。 • informs : ホストに SNMP 情報を送信します。 • version : サポートする SNMP バージョンを指定します。informs にはバージョン 1 (デフォルト) を使用できません。 • community-string : 通知処理で送信する文字列を指定します。snmp-server host コマンドを使用してこの文字列を設定できますが、この文字列を定義するには、snmp-server community 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から、snmp-server host コマンドを使用することを推奨します。 • notification-type : mac-notification キーワードを使用します。
ステップ 4	<p>snmp-server enable traps mac-notification threshold</p> <p>例 :</p> <pre>デバイス(config)# snmp-server enable traps mac-notification threshold</pre>	<p>NMS への MAC しきい値通知トラップをイネーブルにします。</p>

	コマンドまたはアクション	目的
ステップ 5	mac address-table notification threshold 例 : デバイス (config) # mac address-table notification threshold	MAC アドレスしきい値通知機能をイネーブルにします。
ステップ 6	mac address-table notification threshold [limit percentage] [interval time] 例 : デバイス (config) # mac address-table notification threshold interval 123 デバイス (config) # mac address-table notification threshold limit 78	MAC アドレスしきい値使用状況モニタリングのしきい値を入力します。 <ul style="list-style-type: none"> • (任意) limit percentage : MAC アドレステーブルの使用率を指定します。有効値は 1 ~ 100% ですデフォルト値は 50% です。 • (任意) interval time : 通知の間隔を指定します。有効値は 120 秒以上です。デフォルトは 120 秒です。
ステップ 7	end 例 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 8	show running-config 例 : デバイス # show running-config	入力を確認します。
ステップ 9	copy running-config startup-config 例 : デバイス # 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。

スタティック アドレス エントリの追加および削除

スタティック アドレスを追加するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **mac address-table static mac-addr vlan vlan-id interface interface-id**
4. **show running-config**

5. copy running-config startup-config

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <p>デバイス> enable</p>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <p>デバイス# configure terminal</p>	<p>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。</p>
ステップ 3	<p>mac address-table static mac-addr vlan vlan-id interface interface-id</p> <p>例 :</p> <p>デバイス(config)# mac address-table static c2f3.220a.12f4 vlan 4 interface gigabitethernet 1/0/1</p>	<p>MAC アドレス テーブルにスタティック アドレスを追加します。</p> <ul style="list-style-type: none"> • mac-addr : アドレス テーブルに追加する宛先 MAC ユニキャスト アドレスを指定します。この宛先アドレスを持つパケットが指定した VLAN に着信すると、指定したインターフェイスに転送されます。 • vlan-id : 指定された MAC アドレスを持つパケットを受信する VLAN を指定します。指定できる VLAN ID の範囲は 1 ~ 4094 です。 • interface-id : 受信パケットが転送されるインターフェイスを指定します。有効なインターフェイスは、物理ポートまたはポートチャンネルです。スタティック マルチキャスト アドレスの場合、複数のインターフェイス ID を入力できます。スタティック ユニキャスト アドレスの場合、インターフェイスは同時に 1 つしか入力できません。ただし、同じ MAC アドレスおよび VLAN ID を指定すると、コマンドを複数回入力できます。
ステップ 4	<p>show running-config</p> <p>例 :</p> <p>デバイス# show running-config</p>	<p>入力を確認します。</p>

	コマンドまたはアクション	目的
ステップ 5	copy running-config startup-config 例： デバイス# 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。

ユニキャスト MAC アドレス フィルタリングの設定

デバイスが送信元または宛先ユニキャストスタティックアドレスをドロップするよう設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **mac address-table static mac-addr vlan vlan-id drop**
4. **end**
5. **show running-config**
6. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例： デバイス# 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	mac address-table static mac-addr vlan vlan-id drop 例： デバイス (config)# mac address-table static c2f3.220a.12f4 vlan 4 drop	ユニキャスト MAC アドレス フィルタリングをイネーブルにし、 device が指定した送信元または宛先ユニキャストスタティックアドレスを持つパケットをドロップするように設定します。 • mac-addr : 送信元または宛先ユニキャスト MAC アドレス (48ビット) を指定します。この MAC アドレスを持つパケットはドロップされます。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • <i>vlan-id</i> : 指定された MAC アドレスを持つパケットを受信する VLAN を指定します。指定できる VLAN ID の範囲は 1 ~ 4094 です。
ステップ 4	end 例 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 5	show running-config 例 : デバイス # show running-config	入力を確認します。
ステップ 6	copy running-config startup-config 例 : デバイス # 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。

デバイスのモニターリングおよび保守の管理

コマンド	目的
clear mac address-table dynamic	すべてのダイナミックエントリを削除します。
clear mac address-table dynamic address <i>mac-address</i>	特定の MAC アドレスを削除します。
clear mac address-table dynamic interface <i>interface-id</i>	指定された物理ポートまたはポート チャネル上のすべてのアドレスを削除します。
clear mac address-table dynamic vlan <i>vlan-id</i>	指定された VLAN 上のすべてのアドレスを削除します。
show clock [<i>detail</i>]	時刻と日付の設定を表示します。
show ip igmp snooping groups	すべての VLAN または指定された VLAN に対するレイヤ 2 マルチキャスト エントリを表示します。
show mac address-table address <i>mac-address</i>	指定された MAC アドレスの MAC アドレステーブル情報を表示します。

コマンド	目的
show mac address-table aging-time	すべての VLAN または指定された VLAN の エージング タイムを表示します。
show mac address-table count	すべての VLAN または指定された VLAN で存在しているアドレス数を表示します。
show mac address-table dynamic	ダイナミック MAC アドレス テーブル エントリのみを表示します。
show mac address-table interface <i>interface-name</i>	指定されたインターフェイスの MAC アドレス テーブル情報を表示します。
show mac address-table move update	MAC アドレス テーブル 移動更新情報を表示します。
show mac address-table multicast	マルチキャストの MAC アドレスのリストを表示します。
show mac address-table notification {change mac-move threshold}	MAC 通知パラメータおよび履歴テーブルを表示します。
show mac address-table secure	セキュア MAC アドレスを表示します。
show mac address-table static	スタティック MAC アドレス テーブル エントリだけを表示します。
show mac address-table vlan <i>vlan-id</i>	指定された VLAN の MAC アドレス テーブル情報を表示します。

デバイス管理の設定例

例：システムクロックの設定

次の例は、システムクロックを手動で設定する方法を示しています。

```
デバイス# clock set 13:32:00 23 July 2013
```

例：サマータイムの設定

次に、サマータイムが3月10日の02:00に開始し、11月3日の02:00に終了する場合の設定を例として示します。

```
デバイス(config)# clock summer-time PDT recurring PST date
```

```
10 March 2013 2:00 3 November 2013 2:00
```

次に、サマータイムの開始日と終了日を設定する例を示します。

```
デバイス(config)#clock summer-time PST date  
20 March 2013 2:00 20 November 2013 2:00
```

例：MOTD バナーの設定

次の例は、開始および終了デリミタにポンド記号 (#) を使用して、MOTD バナーを設定する方法を示しています。

```
デバイス(config)# banner motd #  
  
This is a secure site. Only authorized users are allowed.  
For access, contact technical support.  
  
#  
  
デバイス(config)#
```

次に、前の設定により表示されたバナーの例を示します。

```
Unix> telnet 192.0.2.15  
  
Trying 192.0.2.15...  
  
Connected to 192.0.2.15.  
  
Escape character is '^'.  
  
This is a secure site. Only authorized users are allowed.  
  
For access, contact technical support.  
  
User Access Verification  
  
Password:
```

例：ログインバナーの設定

次の例は、開始および終了デリミタにドル記号 (\$) を使用して、ログインバナーを設定する方法を示しています。

```
デバイス(config)# banner login $  
  
Access for authorized users only. Please enter your username and password.  
  
$
```

```
デバイス(config)#
```

例：MAC アドレス変更通知トラップの設定

次に、NMS として 172.20.10.10 を指定し、NMS への MAC アドレス通知トラップの送信をイネーブルにし、MAC アドレス変更通知機能をイネーブルにし、インターバルタイムを 123 秒に設定し、履歴サイズを 100 エントリに設定し、特定のポートで MAC アドレスが追加された場合のトラップをイネーブルにする例を示します。

```
デバイス(config)# snmp-server host 172.20.10.10 traps private mac-notification
デバイス(config)# snmp-server enable traps mac-notification change
デバイス(config)# mac address-table notification change
デバイス(config)# mac address-table notification change interval 123
デバイス(config)# mac address-table notification change history-size 100
デバイス(config)# interface gigabitethernet1/2/1
デバイス(config-if)# snmp trap mac-notification change added
```

例：MAC しきい値通知トラップの設定

次に、NMS として 172.20.10.10 を指定し、MAC アドレスしきい値通知機能をイネーブルにし、インターバルタイムを 123 秒に設定し、制限を 78% に設定する例を示します。

```
デバイス(config)# snmp-server host 172.20.10.10 traps private mac-notification
デバイス(config)# snmp-server enable traps mac-notification threshold
デバイス(config)# mac address-table notification threshold
デバイス(config)# mac address-table notification threshold interval 123
デバイス(config)# mac address-table notification threshold limit 78
```

例：MAC アドレス テーブルへのスタティック アドレスの追加

次の例では、MAC アドレス テーブルにスタティック アドレス c2f3.220a.12f4 を追加する方法を示します。VLAN4でこのMACアドレスを宛先アドレスとして持つパケットを受信すると、パケットは指定されたポートに転送されます。



(注) 複数のインターフェイスに同じ静的 MAC アドレスを関連付けることはできません。コマンドを別のインターフェイスで再度実行すると、新しいインターフェイス上で静的 MAC アドレスが上書きされます。

```
デバイス(config)# mac address-table static c2f3.220a.12f4 vlan 4 interface
gigabitethernet1/1/1
```

例：ユニキャスト MAC アドレス フィルタリングの設定

次に、ユニキャスト MAC アドレス フィルタリングをイネーブルにし、c2f3.220a.12f4 の送信元または宛先アドレスを持つドロップパケットを設定する例を示します。送信元または宛先としてこの MAC アドレスを持つパケットが VLAN4 上で受信された場合、パケットがドロップされます。

```
デバイス(config)# mac address-table static c2f3.220a.12f4 vlan 4 drop
```

デバイス管理に関する追加情報

関連資料

関連項目	マニュアル タイトル
システム管理コマンド	<i>Command Reference (Catalyst 9300 Series Switches)</i>

MIB

MIB	MIB のリンク
本リリースでサポートするすべての MIB	<p>選択したプラットフォーム、Cisco IOS リリース、およびフィッチャセットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。</p> <p>http://www.cisco.com/go/mibs</p>

シスコのテクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	<p>http://www.cisco.com/support</p>

デバイス管理の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	デバイス管理	デバイス管理では、システムの日時、システム名、ログインバナーを設定し、DNSを設定できます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 2 章

ブート整合性の可視性

- [ブート整合性の可視性について \(49 ページ\)](#)
- [ソフトウェアイメージとハードウェアの確認 \(49 ページ\)](#)
- [プラットフォーム ID とソフトウェア整合性の確認 \(50 ページ\)](#)
- [ブート整合性の可視性に関する追加情報 \(52 ページ\)](#)
- [ブート整合性の可視性の機能履歴 \(53 ページ\)](#)

ブート整合性の可視性について

ブート整合性の可視性によって、シスコのプラットフォーム ID とソフトウェアの整合性情報が可視化され、実用可能になります。プラットフォーム ID は、プラットフォームの製造元でインストールされた ID を提供します。ソフトウェアの整合性ではブート整合性の測定値が明らかになり、それを使用してプラットフォームが信頼できるコードを起動しているかどうかを評価できます。

ブートプロセス中に、ソフトウェアはブートローダーアクティビティの各ステージのチェックサムレコードを作成します。

このレコードを取得して、シスコ認定レコードと比較し、ソフトウェアイメージが正規かどうかを確認できます。チェックサム値が一致していない場合は、シスコによって認定されていない、または未承認パーティによって改ざんされているソフトウェアイメージを実行している可能性があります。



- (注) ブート整合性の可視性は、アクティブなスーパーバイザでのみサポートされます。高可用性シナリオはサポートしていません。

ソフトウェアイメージとハードウェアの確認

このタスクでは、スイッチの起動時に作成されたチェックサムレコードを取得する方法について説明します。特権 EXEC モードで次のコマンドを入力します。



- (注) 次のコマンドを実行した後で、メッセージ **% Please Try After Few Seconds** が CLI に表示されることがあります。これは CLI の障害を示すものではありませんが、必要な出力を取得するために必要な基盤となるインフラストラクチャの設定を示します。数分間待機して、コマンドを再度試すことをお勧めします。

メッセージ **% Error retrieving SUDI certificate** および **% Error retrieving integrity data** は、実際の CLI 障害を示します。

手順の概要

1. `show platform sudi certificate [sign [nonce nonce]]`
2. `show platform integrity [sign [nonce nonce]]`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<code>show platform sudi certificate [sign [nonce nonce]]</code> 例： デバイス# <code>show platform sudi certificate sign nonce 123</code>	特定の SUDI のチェックサム レコードを表示します。 <ul style="list-style-type: none"> • (オプション) sign : 署名を示します • (オプション) nonce : ナンス値を入力します
ステップ 2	<code>show platform integrity [sign [nonce nonce]]</code> 例： デバイス# <code>show platform integrity sign nonce 123</code>	ブート段階のチェックサム レコードを表示します。 <ul style="list-style-type: none"> • (オプション) sign : 署名を示します • (オプション) nonce : ナンス値を入力します

プラットフォーム ID とソフトウェア整合性の確認

プラットフォーム ID の確認

次に、PEM 形式でセキュアな固有デバイス識別子 (SUDI) チェーンを表示する例を示します。最初の証明書は Cisco Root CA 2048 で、2 番目はシスコの下位 CA (ACT2 SUDI CA) です。どちらの証明書も、<https://www.cisco.com/security/pki/> で公開されているものと一致しているかを確認できます。3 番目は SUDI 証明書です。

```
Device# show platform sudi certificate sign nonce 123
-----BEGIN CERTIFICATE-----
MIIDQzCCAiugAwIBAgIQX/h7KctU3I1CoxW1aMmt/zANBgkqhkiG9w0BAQUFADA1
MRYwFAYDVQQKEw1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDEsJDaXNjbyBSb290IENB
IDlwNDgwHhcNMDQwNTE0MjAxNzEyWhcNMjkwNTE0MjAyNTQyWjAlMRYwFAYDVQQK
Ew1DaXNjbyBTeXN0ZW1zMRswGQYDVQQDEsJDaXNjbyBSb290IENBIDlwNDgwggEg
MA0GCSqGSIb3DQEBAQUAA4IBDQAwggEIAoIBAQCwmrmrp68Kd6ficba0ZmKueIhH
```

```

xmJvHEAyv8CrLqUccda8bnuoqrpu0hWISEWdovyD0My5j0AmaHBKeN8hF570YQXJ
FcjPftolYmUQ6lEqDGYeJu5Tm8sUxJszR2tKyS7McQr/4NEb7Y9JHcJ6r8qqB9q
VvYgDxFUL4F1pyXOWWqCZe+36ufijXWlLvLdT6ZeYpZPEApk0E5tzivMW/VgpSdh
jWn0f84bcN5wGyDWbs2mAg8EtKp6BrXruOIIt6ke0la06g58QBdKhTCytKmg9l
Eg6CTY5j/e/rmxrbU6YTYK/CfdHbBcl1HP7R2RQgYUCUOG/rksc35LTLgXfAgED
o1EwTzALBGNVHQ8EBAMCAYwDwYDVR0TAQH/BAUwAwEB/zAdBgNVHQ4EFgQUJ/PI
FR5umgIJFq0roIlgX9p7L6owEAYJKwYBBAGCNxUBBAMCAQAwDQYJKoZIhvcNAQEF
BQADggEBAJ2dhISjQal8dwy3U8pORFbi71R803UXHOjgXkLtv5M0hmBvRbW7hmW
Yqpa02TB9k5UM8Z3/sUcuuVdJcr18JOagEu5sv4dEX+5wW4q+ffy0vhN4TauYuX
cB7w4ovXsNgOnbFp1iqRe6lJT37mjpXYgyc81WhJdTsd9i7rp77rMKsS0t8lasz
Bvt9YaretIpsjYp8qS5UwGH0GikJ3+r/+n6yUA4iGe0OcaEblfJU9u6ju7AQ7L4
CYNu/2bPPu8XslgYJQkOXuPL1hs27PKsb3TkL4Eq1ZKR4OCXPdJoBYVL0fdX41Id
kxpUnwVwwEpxYB5DC2Ae/qPOgRnhCzU=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIEPDCAySgAwIBAgIKYQ1ufQAAAAAADANBgkqhkiG9w0BAQUFADA1MRYwFAyD
VQQKEw1DaXNjbyBTeXNOZW1zMRswGQYDVQDEw1DaXNjbyBzSb290IENBIDwNDGw
HhcNMTEwNjMwMjE1NjU3W3hcnmJkNTE0MjA5NTQyWjAnMQ4wDAYDVQQKEw1DaXNj
bzEVMBMGA1UEAxMMQUNUMiBTUURJIEENBmIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8A
MIIBGgKCAQEA0m513THlxA9tN/hS5qR/6UZRpdd+9aE2JbFknjht6gfHKd477AkS
5XAtUs5oxDYVt/zEbslZq3+LR6grqKQVu6JYvh05UYLBqCj38s76NLk53905Wzp
9pRcmRCPUx+a6tHF/qRuoiJ44mdeDYzo3qPCpZpXrWJDPclM4iYKHumMQMqmgmg+
xghHiooWS80BOcdiynEbeP5rZ7qRuewKmpl11TiI3WdBNJzjnpfjg66F+P4SaDkGb
BXdGj13oVeF+EyFWLrFj97fL2+8oauV43Qrvnf3d/GfqXj7ew+z/sXlXtEojSXJ
URsYMEj53Rdd9tJwHky8neapszS+r+kdVQIDAQABo4IBWjCCAAYwCwYDVR0PBAQD
AgHGMB0GA1UdDgQWBRI2PHxwnDVW7t8cwmTr7i4MAP4fzAfBgNVHSMEGDAwBQn
88gVHm6aAgkWrSugiWBF2nsvqjBDBGNVHR8EPDA6MDiGNgA0hjJodHRWoi8vd3d3
LmNpc2NvLmNvbS9zZWN1cm10eS9wa2kvY3JsL2NyY2EyMDQ4LmNybDBQBggrBgEF
BQcBAQREMEIwQAYIKwYBBQUHMAKGNh0dHA6Ly93d3cuY2l2y28uY29tL3N1Y3VyaXR5
aXR5L3BraS9jZXJ0cy9jcmNhMjA0OC5jZXIwXAYDVDR0GgBFUwUzBRBgorBgEAAQkV
AQwAMEMwQQYIKwYBBQUHAgEWNWh0dHA6Ly93d3cuY2l2y28uY29tL3N1Y3VyaXR5
L3BraS9wb2xpY2l2cy9pbmRleC5odG1sMBIGA1UdEwEB/wQIMAYBAf8CAQAwdQYJ
KoZIhvcNAQEFBQADggEBAGh1qclr9tx4hzWgDERm371yeuEmqcIffi9b9+GbsMjbi
ZHc/Cc10lJu0a9zTXA9w47H9/t6leduGxb4WeLxcwCiUgvFtCa51Iklt8nNbcKY
/4dw1ex+7amATUQ04QggIE67wVIPu6bgAE3Ja/nRS3xKYSnj8H5TehimBSv6TECi
i5jUhoWryAK4dVo8hCjkjEkzu3ufBTJapnv89g9OE+H3VKM4L+/KdkUO+52djFkn
hyl47d7cZR4DY4LlUfM2P1As8YyjoNpK/urSRI14WdI1plRlnH7KND15618yFVP
0IFJZBGrooCRBjOsFv8cpWcbmWdPaCQT2nwIjTfy8c=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIDezCCAmOgAwIBAgIEAc+JiTANBgkqhkiG9w0BAQsFADANMQ4wDAYDVQQKEwVD
aXNjbyBzEVMBMGA1UEAxMMQUNUMiBTUURJIEENBmB4XDTE3MDgxoTEwNDMzOVoxDTI3
MDgxOTEwNDMzOVowZzEmMCQGA1UEBRMUE1EOKm5MzAwLTI0VVggU046RkNxmJez
NEwwMEMxZjAMBGNVBAoTBUNpc2NvMRgwFgYDVQLLEw9BQ1QtMiBmaXRlIFNVREkx
EzARBGNVBAMTCkM5MzAwLTI0VVggwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQDdav5txv4THsqXwWC7AxzHm5MZ28Feqk8FA3tXAv0tV8RxtY4Z9I9XgRzWz
Yw8chknh8LuDMcmGmk8DP+ct++vAF4nkVeIeBeOHnx2RuC9rcR8tuKjCimamDk0M
Jhk12w/9+TbdKdNBEy6SuehlRPVbuSk1oQLQcOYW7CsYc5tI1GkJKfk1nGEK3ni3
ztIpsi7QHyp6k59ycnbnzXSdwoBrtpIIEYk/iHwFRQdlMUunnfIshI7yPne07V0
NnPc08wk+CA+8XeXk/fnDeGAswKRK1tW9jDF/sY1YubBJNj4ToqQpG6W/hbNvu3Y
NyS24osSvnn5Bp7on3Rf7eHq9hNjAgMBAAGjbjBtMA4GA1UdDwEB/wQEAwIF4DAM
BgNVHRMBAf8EAjAAAME0GA1UdEQRGMEsGQgYJKwYBBAEJFQIDoDUTM0NoaXBjRd1V
WUpPVkVZNEZRT0xSbkpwSUUxaGNpQXhNQ0F4Tnpvd05Eb3hNeUfiY1FjPTANBgkq
hkiG9w0BAQsFAAOCAQEASXX+iZLMvHQIR1/s1Pobm0kP/bYeHsgDTRQPRHbCM1HH
ROfjJDaJMHCspB17XtcLkNNFOWyUEkjoePyHjpxxhekGIqgD6Xt4rW6v/058Haw6
QbAhJFGZriVxFOBvW20VQ4ezyaGoqA+0I2GzQD/ZggUy6zsVwKme6inoEgXcYap
5GqF4weEoty9u+OKqr3ppWU4751xnNm/h+WHbNtunL6r7wZfe5dFQIXR5QP5gWra
svpSsCoKm6PiuUhw25CvtZ9NTg0tu5t5D7aVcxLeR8XbAlpjfgxw/RtSsjNse3+
ZkOgJUESqlxwzxcGULY+vDINyRQ/sP6y7cT+niT00A==
-----END CERTIFICATE-----

```

Signature version: 1
Signature:

ソフトウェア整合性の確認

次に、ブート段階のチェックサムレコードを表示する例を示します。ハッシュ測定値は、連続してブートされたソフトウェアの3つの段階それぞれについて表示されます。これらのハッシュをシスコが提供する基準値と比較できます。出力に署名するオプションを使用すると、出力は正規であり改ざんされていないことを保証する機能が検証ツールに付与されます。リプレイ攻撃から保護するために、ナンスを提供できます。

```
Device# show platform integrity sign nonce 123
Platform: C9300-24P
Boot 0 Version: MA0081R06.1307262016
Boot 0 Hash: A99EF9F31CE3F3F8533055407F1C88C62176E667E4E1DA0649EAA7A1282F205E0A
Boot Loader Version: System Bootstrap, Version 16.8.0.3, RELEASE SOFTWARE (P)
Boot Loader Hash:
F82826514658055C3993AE95F53512341EF20F3CC7D4083C980450EA6CD84608EE6385B15D13414203CED35603F01974B8676C6AC6F9DC45E25CD1039E686C40A
OS Version: BLD_POLARIS_DEV_LATEST_20171213_030750
OS Hash:
E7336A416FE232CA87C73C5C6387EB7244560FBE9F977207D8783C113217E3DD4CA16C40B16A8CC9841100264D04CAFE3AE863EB94FE561F9851AB167E913830A
PCR0: 9745B571B66D79F0936F4D292B5672B50F50FD1E56E74248D48A33582E992574
PCR8: 1CC295C233DA41BD3530A6F09C21991E8406BFFC88249D7778CA4BB0B9E71EB7
Signature version: 1
Signature:
```

ブート整合性の可視性に関する追加情報

関連資料

関連項目	マニュアルタイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	<i>Command Reference (Catalyst 9300 Series Switches)</i>

MIB

MIB	MIB のリンク
本リリースでサポートするすべての MIB	<p>選択したプラットフォーム、Cisco IOS リリース、およびフィチャセットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。</p> <p>http://www.cisco.com/go/mibs</p>

シスコのテクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	http://www.cisco.com/support

ブート整合性の可視性の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Fuji 16.8.1a	ブート整合性の可視性	ブート整合性の可視性によって、シスコのプラットフォーム ID とソフトウェアの整合性情報が可視化され、実用可能になります。プラットフォーム ID は、プラットフォームの製造元でインストールされた ID を提供します。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 3 章

デバイスのセットアップ設定の実行

- ソフトウェア インストールの制約事項 (55 ページ)
- デバイスセットアップ設定の実行に関する情報 (55 ページ)
- デバイスセットアップ設定の実行方法 (68 ページ)
- デバイスのセットアップ設定のモニターリング (84 ページ)
- デバイスのセットアップを実行する場合の設定例 (89 ページ)
- デバイスセットアップの実行に関する追加情報 (104 ページ)
- デバイスセットアップ設定の実行に関する機能履歴 (104 ページ)

ソフトウェア インストールの制約事項

- サブパッケージのインストールはサポートされていません。

デバイスセットアップ設定の実行に関する情報

IP アドレスの割り当ておよび DHCP 自動設定を含む初期 device 設定タスクを実行する前に、このモジュールのセクションを確認します。

デバイスブート プロセス

device を起動するには、ハードウェア 設置ガイドの手順に従って device を設置して電源投入し、device の初期設定を行う必要があります。

通常の起動プロセスにはブートローダソフトウェアの動作が含まれ、以下のアクティビティが実行されます。

- 下位レベルの CPU 初期化を行います。CPU レジスタを初期化することにより、物理メモリがマッピングされる場所、容量、速度などを制御します。
- CPU サブシステムの電源投入時セルフ テスト (POST) を実行し、システム DRAM をテストします。

- システム ボード上のファイル システムを初期化します。
- デフォルトのオペレーティング システム ソフトウェア イメージをメモリにロードし、`device`を起動します。

ブート ロードにより、オペレーティング システムがロードされる前に、ファイル システムにアクセスすることができます。ブート ロードの使用目的は通常、オペレーティング システムのロード、展開、および起動に限定されます。オペレーティング システムが CPU を制御できるようになると、ブートローダは、次にシステムがリセットされるか電源が投入されるまでは非アクティブになります。

`device`情報を割り当てるには、PC または端末をコンソール ポートに接続するか、PC をイーサネット管理ポートに接続して、PC または端末エミュレーション ソフトウェアのボーレートおよびキャラクタ フォーマットを `device`のコンソールポートの設定と一致させておく必要があります。

- デフォルトのボーレートは 9600 です。
- デフォルトのデータ ビットは 8 です。



(注) データ ビット オプションを 8 に設定した場合、パリティ オプションは「なし」に設定します。

- デフォルトのストップ ビットは 2 (マイナー) です。
- デフォルトのパリティ設定は「なし」です。

ソフトウェア インストールの概要

ソフトウェア インストール機能では、イメージの完全インストール、ソフトウェア メンテナンス アップグレード (SMU)、インサービス ソフトウェア アップグレード (ISSU)、およびインサービス モデル アップグレード (データ モデル パッケージ) など、さまざまなタイプのアップグレードを同じように実行できます。

ソフトウェア インストール機能は、インストール モードでソフトウェアを 1 つのバージョンから別のバージョンへと移行する際に役立ちます。`install` コマンドを特権 EXEC モードで使用して、ソフトウェア イメージをインストールまたはアップグレードします。また、インストール モードを使用して以前のバージョンのソフトウェア イメージにダウングレードすることもできます。

Cisco IOS XE ソフトウェアをアップグレードするために使用する方式は、スイッチが動作しているのがインストール モードかバンドル モードかによって異なります。バンドル モードまたは統合ブートモードでは、ローカルまたはリモート ロケーションから `.bin image` ファイルを使用してデバイスをブートします。インストールブートモードでは、ブートローダが `packages.conf` ファイルを使用してデバイスをブートします。

スイッチでは、次のソフトウェア インストール機能がサポートされています。

- スタンドアロン スイッチでのソフトウェア バンドルのインストール。
- 以前にインストールしたパッケージセットへのソフトウェア ロールバック。
- 有効なインストール済みパッケージがブート フラッシュに存在しない場合の緊急インストール。

ソフトウェアのブートモード

deviceでは、ソフトウェア パッケージを起動するための次の2種類のモードがサポートされています。

- インストール モード
- バンドル モード

インストールモードでのブート

以下のフラッシュ内のソフトウェア パッケージのプロビジョニング ファイルを起動して、インストール モードでdeviceを起動できます:

```
Switch: boot flash:packages.conf
```



-
- (注) 特定リリース用の `packages.conf` ファイルが「ソフトウェア パッケージのインストール」という項で説明するインストール ワークフローで作成されています。
-

プロビジョニング ファイルには、起動、マウント、実行するソフトウェア パッケージのリストが含まれます。インストールされている各パッケージの ISO ファイル システムは、フラッシュからルート ファイル システムに直接マウントされます。



-
- (注) インストール モードで起動するために使用するパッケージとプロビジョニング ファイルは、フラッシュに保存する必要があります。usbflash0 または tftp: からインストール モードで起動することはサポートされていません。
-

バンドルモードでのブート

バンドル (.bin) ファイルを使用して、device をバンドル モードでブートできます:

```
switch: boot flash:cat9k_iosxe.16.05.01a.SPA.bin
```

バンドルに含まれるプロビジョニングファイルは、どのパッケージを起動、マウント、および実行するかを判断するために使用されます。パッケージはバンドルから取得され、RAM にコピーされます。各パッケージの ISO ファイル システムは、ルート ファイル システムにマウントされます。

インストール モードでの起動とは異なり、バンドル モードでの起動では、バンドルのサイズに対応するサイズの追加メモリが使用されます。

ブートモードの変更

バンドルブートモードで実行中のデバイスをインストールモードに変更するには、ブート変数を `flash:packages.conf` に設定して **install add file flash:cat9k_2.bin activate commit** コマンドを実行します。コマンドの実行後、デバイスはインストールブートモードでリブートします。

ソフトウェアパッケージのインストール

デバイスにソフトウェアパッケージをインストールするには、**install add**、**install activate**、および **install commit** コマンドを特権 EXEC モードで使用します。

install add コマンドは、ソフトウェアパッケージをローカルまたはリモートの場所からデバイスにコピーします。FTP、HTTP、HTTPs、またはTFTPを使用できます。このコマンドは、`.bin` ファイルの個々のコンポーネントをサブパッケージと `packages.conf` ファイルに抽出します。またファイルを検証して、イメージファイルがプラットフォームに固有であることを確認します。

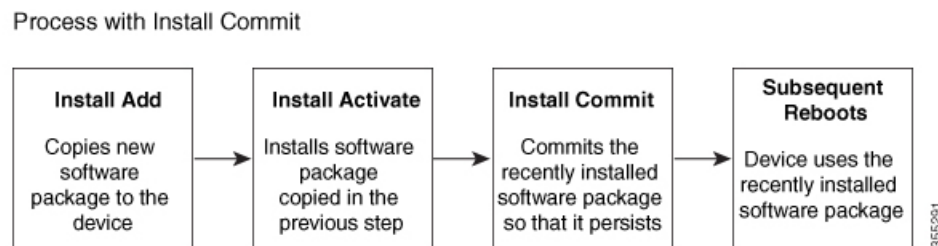
install activate コマンドを動作させるには、パッケージをデバイスのブートフラッシュで使用可能にする必要があります。このコマンドを設定すると、`.bin` ファイルから以前に追加したパッケージがアクティブ化され、システムがリロードします。

install commit コマンドを有効化して、更新プログラムをリロード全体にわたって確定します。

更新をインストールすると、以前にインストールしたソフトウェアイメージが置換されます。デバイスには常に1つのイメージのみがインストールされます。

次のフローチャートで、ソフトウェアのインストールの動作を説明します。

図 3: ソフトウェアパッケージのコミット



(注) **install activate** コマンドは、新しいイメージを使用してデバイスをリロードします。

ソフトウェアインストールの終了

ソフトウェアイメージのアクティブ化は次の方法で終了できます。

- **install activate auto-abort-timer** コマンドを使用します。新しいイメージをアクティブ化した後にデバイスをリロードすると、`auto-abort-timer` がトリガーされます。**install commit** コマンドを発行する前にタイマーが期限切れになった場合、インストールプロセスが終了し

ます。デバイスは再度リロードし、前のバージョンのソフトウェアイメージで起動します。

このタイマーを停止するには、**install auto-abort-timer stop** コマンドを使用します。

- **install abort** コマンドを使用します。このコマンドは、新しいソフトウェアのインストール前に実行していたバージョンにロールバックします。このコマンドは、**install commit** コマンドを発行する前に使用します。

Devices 情報の割り当て

IP 情報を割り当てるには、**device** のセットアッププログラムを使用する方法、DHCP サーバーを使用する方法、または手動で実行する方法があります。

特定の IP 情報の設定が必要な場合、**device** のセットアッププログラムを使用してください。このプログラムを使用すると、ホスト名とイネーブルシークレットパスワードを設定することもできます。

また、任意で、Telnet パスワードを割り当てたり（リモート管理中のセキュリティ確保のため）、スイッチをクラスタのコマンドまたはメンバスイッチとして、あるいはスタンドアロンスイッチとして設定したりできます。

サーバーの設定後は DHCP サーバーを使用して、IP 情報の集中管理と自動割り当てを行います。



- (注) DHCP を使用している場合は、**device** が動的に割り当てられた IP アドレスを受信してコンフィギュレーションファイルを読み込むまでは、セットアッププログラムからの質問に回答しないでください。

device の設定手順を熟知している経験豊富なユーザーの場合は、**device** を手動で設定してください。それ以外のユーザは、「ブートプロセス」で説明したセットアッププログラムを使用してください。

デフォルトのスイッチ情報

表 4: デフォルトのスイッチ情報

機能	デフォルト設定
IP アドレスおよびサブネットマスク	IP アドレスまたはサブネット マスクは定義されていません。
デフォルト ゲートウェイ	デフォルト ゲートウェイは定義されていません。
イネーブルシークレットパスワード	パスワードは定義されていません。

機能	デフォルト設定
ホスト名	出荷時に割り当てられるデフォルトのホスト名はデバイスです。
Telnet パスワード	パスワードは定義されていません。
クラスタ コマンド スイッチ機能	ディセーブル
クラスタ名	クラスタ名は定義されません。

DHCP ベースの自動設定の概要

DHCPは、インターネットホストおよびインターネットワーキングデバイスに設定情報を提供します。このプロトコルには、2つのコンポーネントがあります。1つはDHCPサーバからデバイスにコンフィギュレーションパラメータを提供するコンポーネント、もう1つはデバイスにネットワークアドレスを割り当てるコンポーネントです。DHCPはクライアント/サーバモデルに基づいています。指定されたDHCPサーバが、動的に設定されるデバイスに対して、ネットワークアドレスを割り当て、コンフィギュレーションパラメータを提供します。deviceは、DHCPクライアントおよびDHCPサーバとして機能できます。

DHCPベースの自動設定では、device（DHCPクライアント）は起動時に、IPアドレス情報およびコンフィギュレーションファイルを使用して自動的に設定されます。

DHCPベースの自動設定を使用すると、device上でDHCPクライアント側の設定を行う必要はありません。ただし、DHCPサーバで、IPアドレスに関連した各種リースオプションを設定する必要があります。

DHCPを使用してネットワーク上のコンフィギュレーションファイルの場所をリレーする場合は、TFTPサーバおよびドメインネームシステム（DNS）サーバの設定が必要になることがあります。



- (注) スイッチスタックとDHCP、DNS、TFTPサーバとの間では冗長接続を確立することを推奨します。接続されているスタックメンバーがスイッチスタックから削除された場合でも、これらのサーバがアクセス可能なまま維持されるように保証するうえで役立ちます。

deviceのDHCPサーバは、スイッチと同じLAN上に配置することも、そのdeviceとは別のLAN上に配置することもできます。DHCPサーバが異なるLAN上で動作している場合、deviceとDHCPサーバ間に、DHCPのリレーデバイスを設定する必要があります。リレーデバイスは、直接接続されている2つのLAN間でブロードキャストトラフィックを転送しません。ルータはブロードキャストパケットを転送しませんが、受信したパケットの宛先IPアドレスに基づいてパケットを転送します。

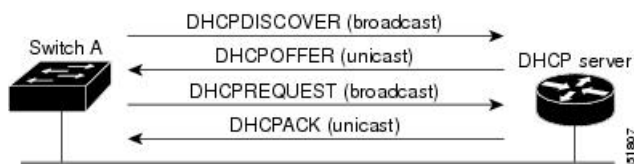
DHCPベースの自動設定は、deviceのBOOTPクライアント機能に代わるものです。

DHCP クライアントの要求プロセス

deviceを起動したときに、deviceにコンフィギュレーションファイルがない場合、DHCP クライアントが呼び出され、DHCPクライアントがDHCPサーバーに設定情報を要求します。コンフィギュレーションファイルが存在し、その設定に特定のルーテッドインターフェイスの **ip address dhcp** インターフェイス コンフィギュレーション コマンドが含まれる場合、DHCP クライアントが呼び出され、DHCPクライアントがインターフェイスにIPアドレス情報を要求します。

次は、DHCPクライアントとDHCPサーバの間で交換される一連のメッセージです。

図 4: DHCPクライアント/サーバ間のメッセージ交換



クライアントであるデバイス A は、DHCPサーバーの場所を特定するために、DHCPDISCOVER メッセージをブロードキャストします。DHCPサーバは、DHCPOFFER ユニキャストメッセージによって、使用可能なコンフィギュレーションパラメータ (IP アドレス、サブネットマスク、ゲートウェイ IP アドレス、DNS IP アドレス、IP アドレス用のリースなど) をクライアントに提示します。

DHCPREQUEST ブロードキャストメッセージでは、クライアントは、提示された設定情報に対して、DHCPサーバに正式な要求を戻します。この正式な要求はブロードキャストされるため、クライアントから DHCPDISCOVER ブロードキャストメッセージを受信した他のすべての DHCP サーバは、クライアントに提示した IP アドレスを再利用できます。

DHCP サーバは、DHCPACK ユニキャストメッセージをクライアントに戻すことで、IP アドレスがクライアントに割り当てられたことを確認します。このメッセージによって、クライアントとサーバはバウンドされ、クライアントはサーバから受信した設定情報を使用します。deviceの受信する情報量は、DHCP サーバの設定方法によって異なります。

DHCPOFFER ユニキャストメッセージによって送信されたコンフィギュレーションパラメータが無効である (コンフィギュレーションエラーがある) 場合、クライアントは DHCP サーバに、DHCPDECLINE ブロードキャストメッセージを戻します。

DHCP サーバはクライアントに、提示されたコンフィギュレーションパラメータが割り当てられていない、パラメータのネゴシエーション中にエラーが発生した、またはDHCPOFFERメッセージに対するクライアントの応答が遅れている (DHCPサーバがパラメータを別のクライアントに割り当てた) という意味のDHCPNAK拒否ブロードキャストメッセージを送信します。

DHCPクライアントは、複数のDHCPサーバまたはBOOTPサーバから提示を受け取り、そのうちの任意の1つを受け入れることができますが、通常は最初に受け取った提示を受け入れません。DHCPサーバから提示されたIPアドレスが必ずしもクライアントに割り当てられるわけではありません。ただし、サーバは通常、クライアントが正式にアドレスを要求するまではアドレスを確保しておきます。deviceがBOOTPサーバからの応答を受け入れ、自身を設定する場合、deviceはdeviceコンフィギュレーションファイルを取得するために、TFTP要求をユニキャストするのではなくブロードキャストします。

DHCP ホスト名オプションにより、`devices`のグループはホスト名および標準コンフィギュレーションを集中管理型 DHCP サーバーから取得できます。クライアント (`device`) は DHCPDISCOVER メッセージ内に、DHCP サーバーからのホスト名および他のコンフィギュレーションパラメータの要求に使用される **Option 12** フィールドを加えます。すべてのクライアントのコンフィギュレーションファイルは、DHCP から取得したホスト名を除き、まったく同じです。

DHCP ベースの自動設定およびイメージアップデート

DHCP イメージアップグレード機能を使用すると、ネットワーク内の1つ以上の`devices`に新しいイメージファイルおよび新しいコンフィギュレーションファイルをダウンロードするように DHCP サーバーを設定できます。ネットワーク内のすべてのスイッチでのイメージおよびコンフィギュレーションの同時アップグレードによって、ネットワークに加えられたそれぞれの新しい`device`が、同じイメージとコンフィギュレーションを確実に受信するようになります。

DHCP イメージアップグレードには、自動設定およびイメージアップデートの2つのタイプがあります。

DHCP ベースの自動設定の制約事項

- ネットワーク内に割り当てられた IP アドレスがなく、1つ以上のレイヤ3 インターフェイスが起動していない場合は、設定プロセスが保存された DHCP ベースの自動設定は停止します。
- タイムアウトを設定しない限り、設定機能を備えている DHCP ベースの自動設定は IP アドレスのダウンロードを無期限に繰り返します。
- コンフィギュレーションファイルをダウンロードできないか破損している場合は、自動インストールプロセスが停止します。
- TFTP からダウンロードされたコンフィギュレーションファイルは、実行コンフィギュレーション内の既存コンフィギュレーションとマージされますが、**write memory** または **copy running-configuration startup-configuration** 特権 EXEC コマンドを入力しない限り、NVRAM に保存されません。ダウンロードされたコンフィギュレーションがスタートアップコンフィギュレーションに保存された場合、後続のシステム再起動中にこの機能はトリガーされません。

DHCP 自動設定

DHCP 自動設定は、コンフィギュレーションファイルを DHCP サーバーからネットワーク内の1つ以上の`devices`にダウンロードします。ダウンロードされたコンフィギュレーションファイルは、`device`の実行コンフィギュレーションファイルになります。このファイルは、`device`がリロードされるまで、フラッシュメモリに保存されたブートアップコンフィギュレーションを上書きしません。

DHCP 自動イメージアップグレード

DHCP 自動設定とともに DHCP 自動イメージアップグレードを使用すると、コンフィギュレーションおよび新しいイメージをネットワーク内の 1 つ以上の devices にダウンロードできます。新しいコンフィギュレーションおよび新しいイメージをダウンロードしている 1 つの device スイッチ（または複数の devices）は、ブランク（つまり、出荷時のデフォルト設定がロードされている状態）にできます。

コンフィギュレーションをすでに持っているスイッチに新しいコンフィギュレーションをダウンロードすると、ダウンロードされたコンフィギュレーションは、スイッチに保存されているコンフィギュレーション ファイルに追加されます（どの既存のコンフィギュレーション ファイルも、ダウンロードされたファイルに上書きされません）。

device の DHCP 自動イメージアップグレードをイネーブルにするには、イメージファイルおよびコンフィギュレーション ファイルがある TFTP サーバーを、正しいオプション 67（コンフィギュレーション ファイル名）、オプション 66（DHCP サーバーホスト名）、オプション 150（TFTP サーバーアドレス）、およびオプション 125（Cisco IOS イメージファイルの説明）の設定で設定する必要があります。

device をネットワークに設置すると、自動イメージアップグレード機能が開始します。ダウンロードされたコンフィギュレーション ファイルは device の実行コンフィギュレーションに保存され、新しいイメージがダウンロードされて device にインストールされます。device を再起動すると、このコンフィギュレーションが device のコンフィギュレーションに保存されます。

DHCP サーバ設定時の注意事項

デバイスを DHCP サーバとして設定する場合、次の注意事項に従ってください。

- DHCP サーバには、device のハードウェア アドレスによって各 device と結び付けられている予約済みのリースを設定する必要があります。
- device に IP アドレス情報を受信させるには、DHCP サーバに次のリース オプションを設定する必要があります。
 - クライアントの IP アドレス（必須）
 - クライアントのサブネット マスク（必須）
 - DNS サーバの IP アドレス（任意）
 - ルータの IP アドレス（device で使用するデフォルト ゲートウェイ アドレス）（必須）
- device に TFTP サーバからコンフィギュレーション ファイルを受信させる場合は、DHCP サーバに次のリース オプションを設定する必要があります。
 - TFTP サーバ名（必須）
 - ブートファイル名（クライアントが必要とするコンフィギュレーション ファイル名）（推奨）
 - ホスト名（任意）

- DHCP サーバーの設定によっては、**device**は IP アドレス情報またはコンフィギュレーションファイル、あるいはその両方を受信できます。
- 前述のリースオプションを設定しなかった場合、DHCP サーバーは、設定されたパラメータのみを使用してクライアントの要求に応答します。IP アドレスおよびサブネットマスクが応答に含まれていないと、**device**は設定されません。ルータの IP アドレスまたは TFTP サーバー名が見つからなかった場合、**device**は TFTP 要求をユニキャストしないでブロードキャストする場合があります。その他のリースオプションは、使用できなくても自動設定には影響しません。
- **device**は DHCP サーバーとして動作することができます。デフォルトでは、Cisco IOS DHCP サーバーおよび DHCP リレー エージェント機能は **device**上でイネーブルにされていますが、設定されていません。（これらの機能は動作しません）

TFTP サーバの目的

DHCP サーバーの設定に基づいて、**device**は TFTP サーバーから 1 つまたは複数のコンフィギュレーションファイルをダウンロードしようとします。TFTP サーバーへの IP 接続に必要なすべてのオプションについて **device**に応答するよう DHCP を設定している場合で、なおかつ、TFTP サーバー名、アドレス、およびコンフィギュレーションファイル名を指定して DHCP サーバーを設定している場合、**device**は指定された TFTP サーバーから指定されたコンフィギュレーションファイルをダウンロードしようとします。

コンフィギュレーションファイル名、および TFTP サーバーを指定しなかった場合、またはコンフィギュレーションファイルをダウンロードできなかった場合は、**device**はファイル名と TFTP サーバーアドレスをさまざまに組み合わせてコンフィギュレーションファイルをダウンロードしようとします。ファイルには、特定のコンフィギュレーションファイル名（存在する場合）と次のファイルが指定されています。**network-config**、**cisconet.cfg**、**hostname.config**、または **hostname.cfg** です。この場合、**hostname**は **device**の現在のホスト名です。使用される TFTP サーバーアドレスには、（存在する場合）指定された TFTP サーバーのアドレス、およびブロードキャストアドレス（255.255.255.255）が含まれています。

deviceが正常にコンフィギュレーションファイルをダウンロードするには、TFTP サーバーのベース ディレクトリに 1 つまたは複数のコンフィギュレーションファイルが含まれていなければなりません。含めることのできるファイルは、次のとおりです。

- DHCP 応答で指定されているコンフィギュレーションファイル（実際の **device** コンフィギュレーションファイル）。
- **network-config** または **cisconet.cfg** ファイル（デフォルトのコンフィギュレーションファイル）
- **router-config** または **ciscotr.cfg** ファイル（これらのファイルには、すべての **devices**に共通のコマンドが含まれています。通常、DHCP および TFTP サーバが適切に設定されていれば、これらのファイルはアクセスされません）

DHCP サーバリース データベースに TFTP サーバ名を指定する場合は、DNS サーバのデータベースに TFTP サーバ名と IP アドレスのマッピングを設定することも必要です。

使用する TFTP サーバーが、**device**とは異なる LAN 上にある場合、または**device**がブロードキャスト アドレスを使用してアクセスした場合（前述のすべての必須情報が DHCP サーバーの応答に含まれていない場合に発生）は、リレーを設定して TFTP サーバーに TFTP パケットを転送する必要があります。適切な解決方法は、必要なすべての情報を使用して DHCP サーバを設定することです。

DNS サーバの目的

DHCP サーバは、DNS サーバを使用して TFTP サーバ名を IP アドレスに変換します。DNS サーバ上で、TFTP サーバ名から IP アドレスへのマッピングを設定する必要があります。TFTP サーバには、**device**のコンフィギュレーションファイルが存在します。

DHCP の応答時に IP アドレスを取得する DHCP サーバのリース データベースに、DNS サーバの IP アドレスを設定できます。リース データベースには、DNS サーバの IP アドレスを 2 つまで入力できます。

DNS サーバは、**device**と同じ LAN 上に配置することも、別の LAN 上に配置することもできます。DNS サーバが別の LAN 上に存在する場合、**device**はルータを介して DNS サーバにアクセスできなければなりません。

コンフィギュレーションファイルの入手方法

IP アドレスおよびコンフィギュレーションファイル名が DHCP で専用のリースとして取得できるかどうかに応じて、**device**は次の方法で設定情報を入手します。

- IP アドレスおよびコンフィギュレーションファイル名が、**device**用に予約され、DHCP 応答（1 ファイル読み込み方式）で提供されている場合

deviceは DHCP サーバから、IP アドレス、サブネットマスク、TFTP サーバアドレス、およびコンフィギュレーションファイル名を受信します。**device**は、TFTP サーバにユニキャストメッセージを送信し、指定されたコンフィギュレーションファイルをサーバのベース ディレクトリから取得して、ブートアッププロセスを完了します。

- **device**の IP アドレスおよびコンフィギュレーションファイル名が予約されているが、DHCP 応答に TFTP サーバアドレスが含まれていない場合（1 ファイル読み込み方式）。

deviceは DHCP サーバから、IP アドレス、サブネットマスク、およびコンフィギュレーションファイル名を受信します。**device**は、TFTP サーバにブロードキャストメッセージを送信し、指定されたコンフィギュレーションファイルをサーバのベースディレクトリから取得して、ブートアッププロセスを完了します。

- IP アドレスだけが**device**用に予約され、DHCP 応答で提供されており、コンフィギュレーションファイル名は提供されない場合（2 ファイル読み込み方式）

deviceは DHCP サーバから、IP アドレス、サブネットマスク、および TFTP サーバアドレスを受信します。**device**は、TFTP サーバにユニキャストメッセージを送信し、`network-config` または `cisconet.cfg` のデフォルト コンフィギュレーションファイルを取得します（`network-config` ファイルが読み込めない場合、**device**は `cisconet.cfg` ファイルを読み込みます）。

デフォルト コンフィギュレーション ファイルには、**device**のホスト名から IP アドレスへのマッピングが含まれています。**device**は、ファイルの情報をホスト テーブルに書き込み、ホスト名を入手します。ファイルにホスト名がない場合、**device**は DHCP 応答で指定されたホスト名を使用します。DHCP 応答でホスト名が指定されていない場合、**device**はデフォルトの *Switch* ホスト名として使用します。

デフォルトのコンフィギュレーション ファイルまたは DHCP 応答からホスト名を入手した後、**device**はホスト名と同じ名前のコンフィギュレーションファイル (*network-config* または *cisconet.cfg*) のどちらが先に読み込まれたかに応じて、*hostname-config* または *hostname.cf* を TFTP サーバーから読み込みます。*cisconet.cfg* ファイルが読み込まれている場合は、ホストのファイル名は 8 文字に切り捨てられます。

network-config、*cisconet.cfg*、またはホスト名と同じ名前のファイルを読み込むことができない場合、**device**は *router-config* ファイルを読み込みます。*router-config* ファイルを読み込むことができない場合、**device**は *ciscortt.cfg* ファイルを読み込みます。



- (注) DHCP 応答から TFTP サーバーを入手できなかった場合、ユニキャスト伝送によるコンフィギュレーション ファイルの読み込みにすべて失敗した場合、または TFTP サーバー名を IP アドレスに変換できない場合には、**device**は TFTP サーバー要求をブロードキャストします。

環境変数の制御方法

通常動作**device**では、9600 bps に設定されているコンソール接続のみを通じてブート ロード モードを開始します。電源コードを再接続中に**device**電源コードを取り外し、[Mode] ボタンを押します。システム LED がグリーン の点滅から点灯したままになったら、[Mode] ボタンを放してもかまいません。ブートローダの**device**プロンプトが表示されます。

deviceのブートローダソフトウェアは不揮発性の環境変数をサポートするため、これらの環境変数を使用して、ブートローダまたはシステムで稼働する他のソフトウェアの動作を制御できます。ブートローダの環境変数は、UNIX または DOS システムで設定できる環境変数と類似しています。

値を持つ環境変数は、フラッシュ ファイル システムの外にあるフラッシュ メモリに保存されます。

ファイルの各行には、環境変数名と等号に続いて、その変数の値が指定されます。変数が存在しない場合は、変数の値はありません。値がヌルストリングと表示された場合は、変数に値が設定されています。ヌルストリング (たとえば "") が設定されている変数は、値が設定された変数です。多くの環境変数は事前に定義されており、デフォルト値が設定されています。

環境変数の設定を変更するには、ブートローダにアクセスするか、Cisco IOS コマンドを使用します。通常的环境では、環境変数の設定を変更する必要はありません。

TFTP の環境変数

イーサネット管理ポートを通してスイッチに PC を接続していると、TFTP でブートローダに対してコンフィギュレーションファイルのアップロードまたはダウンロードができます。このテーブルの環境変数が設定されていることを確認します。

表 5: TFTP の環境変数

変数	説明
MAC_ADDR	<p>スイッチの MAC アドレスを指定します。</p> <p>(注) 変数は変更しないことを推奨します。</p> <p>ただし、ブートローダを稼働した後に変数を変更した場合、またはこの変数が保存されている値と異なる場合は、TFTP を使用する前にこのコマンドを入力します。新しい値を有効にするためにリセットする必要があります。</p>
IP_ADDRESS	<p>スイッチの関連付けられた IP サブネットに IP アドレスおよびサブネットマスクを指定します。</p>
DEFAULT_GATEWAY	<p>デフォルトゲートウェイに IP アドレスおよびサブネットマスクを指定します。</p>

ソフトウェアイメージのリロードのスケジューリング

device 上でソフトウェアイメージのリロードを後で（深夜、週末など device をあまり使用しないときに）行うように、スケジュールを設定できます。または（ネットワーク内のすべての devices でソフトウェアをアップグレードする場合など）ネットワーク全体でリロードを同時に行うことができます。



(注) リロードのスケジュールは、約 24 日以内に設定する必要があります。

リロードオプションには以下のものがあります。

- 指定した分数、または時間および分数が経過したときに、ソフトウェアがリロードされます。リロードは、約 24 時間以内に実行する必要があります。最大 255 文字で、リロードの理由を指定できます。
- ソフトウェアのリロードが（24時間制で）指定された時間に有効になります。月日を指定すると、指定された日時にリロードが行われるようにスケジュールが設定されます。月日を指定しなかった場合、リロードは当日の指定時刻に行われます（指定時刻が現時刻より後の場合）。または翌日の指定時刻に行われます（指定時刻が現在時刻よりも前の場合）。00:00 を指定すると、深夜 0 時のリロードが設定されます。

reload コマンドはシステムを停止させます。手動で起動することが設定されていない限り、システムは自動的に再起動します。

手動で起動するように **device** が設定されている場合、仮想端末からリロードを実行しないでください。これは **device** がブートローダモードになることでリモートユーザーが制御を失う事態を防止するための制約です。

コンフィギュレーションファイルを変更すると、リロードの前にコンフィギュレーションを保存するように指示するプロンプトが **device** により表示されます。保存操作時に、**CONFIG_FILE** 環境変数がすでに存在しないスタートアップ コンフィギュレーション ファイルを示していた場合、保存を続行するかどうかという問い合わせがシステムから出されます。その状況のまま続けると、リロード時にセットアップ モードが開始されます。

スケジュールがすでに設定されたリロードを取り消すには、**reload cancel** 特権 EXEC コマンドを使用します。

デバイスセットアップ設定の実行方法

DHCP を使用して **device** に新しいイメージおよび新しいコンフィギュレーションをダウンロードするには、少なくとも 2 つの **devices** を設定する必要があります。1 つ目の **device** は DHCP サーバーおよび TFTP サーバーと同じように機能し、2 つ目の **device** (クライアント) は新しいコンフィギュレーション ファイル、または新しいコンフィギュレーション ファイルおよび新しいイメージ ファイルをダウンロードするように設定されています。

DHCP 自動設定 (コンフィギュレーション ファイルだけ) の設定

このタスクでは、新しい **device** の自動設定をサポートできるように、ネットワーク内の既存の **device** で TFTP や DHCP の設定の DHCP 自動設定を行う方法を示します。

手順の概要

1. **configure terminal**
2. **ip dhcp pool poolname**
3. **boot filename**
4. **network network-number mask prefix-length**
5. **default-router address**
6. **option 150 address**
7. **exit**
8. **tftp-server flash:filename.text**
9. **interface interface-id**
10. **no switchport**
11. **ip address address mask**
12. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : デバイス# 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	ip dhcp pool poolname 例 : デバイス (config)# ip dhcp pool pool	DHCP サーバアドレス プールの名前を作成し、DHCP プール 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	boot filename 例 : デバイス (dhcp-config)# boot config-boot.text	ブートイメージとして使用されるコンフィギュレーション ファイルの名前を指定します。
ステップ 4	network network-number mask prefix-length 例 : デバイス (dhcp-config)# network 10.10.10.0 255.255.255.0	DHCP アドレス プールのサブネット ネットワーク番号およびマスクを指定します。 (注) プレフィックス長は、アドレス プレフィックスを構成するビット数を指定します。プレフィックスは、クライアントのネットワーク マスクを指定する二者択一の方法です。プレフィックス長は、スラッシュ (/) で開始する必要があります。
ステップ 5	default-router address 例 : デバイス (dhcp-config)# default-router 10.10.10.1	DHCP クライアントのデフォルト ルータの IP アドレスを指定します。
ステップ 6	option 150 address 例 : デバイス (dhcp-config)# option 150 10.10.10.1	TFTP サーバの IP アドレスを指定します。
ステップ 7	exit 例 :	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
	デバイス (dhcp-config) # exit	
ステップ 8	tftp-server flash:filename.text 例： デバイス (config) # tftp-server flash:config-boot.text	TFTP サーバ上のコンフィギュレーションファイルを指定します。
ステップ 9	interface interface-id 例： デバイス (config) # interface gigabitethernet1/0/4	コンフィギュレーションファイルを受信するクライアントのアドレスを指定します。
ステップ 10	no switchport 例： デバイス (config-if) # no switchport	インターフェイスをレイヤ 3 モードにします。
ステップ 11	ip address address mask 例： デバイス (config-if) # ip address 10.10.10.1 255.255.255.0	IP アドレスとインターフェイスのマスクを指定します。
ステップ 12	end 例： デバイス (config-if) # end	特権 EXEC モードに戻ります。

DHCP 自動イメージアップデート（コンフィギュレーションファイルおよびイメージ）の設定

このタスクでは、新しいスイッチのインストールをサポートするように既存の device で TFTP および DHCP を設定する DHCP 自動設定について説明します。

始める前に

最初にdevice1にアップロードするテキスト ファイル（たとえば、autoinstall_dhcp）を作成します。このテキストファイル内に、ダウンロードするイメージの名前を含めます（たとえば、cat9k_iosxe.16.xx.xx.SPA.bin）。

手順の概要

1. **configure terminal**
2. **ip dhcp pool *poolname***
3. **boot *filename***
4. **network *network-number mask prefix-length***
5. **default-router *address***
6. **option 150 *address***
7. **option 125 *hex***
8. **copy tftp flash *filename.txt***
9. **copy tftp flash *imagename.bin***
10. **exit**
11. **tftp-server flash: *config.txt***
12. **tftp-server flash: *imagename.bin***
13. **tftp-server flash: *filename.txt***
14. **interface *interface-id***
15. **no switchport**
16. **ip address *address mask***
17. **end**
18. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# 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	ip dhcp pool <i>poolname</i> 例： デバイス (config)# ip dhcp pool pool1	DHCP サーバアドレス プールの名前を作成し、DHCP プール 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	boot <i>filename</i> 例： デバイス (dhcp-config)# boot config-boot.text	ブート イメージとして使用されるファイルの名前を指定します。

	コマンドまたはアクション	目的
ステップ 4	network <i>network-number mask prefix-length</i> 例： デバイス (dhcp-config) # network 10.10.10.0 255.255.255.0	DHCP アドレス プールのサブネット ネットワーク 番号およびマスクを指定します。 (注) プレフィックス長は、アドレス プレフィックスを構成するビット数を指定します。プレフィックスは、クライアントのネットワーク マスクを指定する二者択一の方法です。プレフィックス長は、スラッシュ (/) で開始する必要があります。
ステップ 5	default-router <i>address</i> 例： デバイス (dhcp-config) # default-router 10.10.10.1	DHCP クライアントのデフォルト ルータの IP アドレスを指定します。
ステップ 6	option 150 <i>address</i> 例： デバイス (dhcp-config) # option 150 10.10.10.1	TFTP サーバの IP アドレスを指定します。
ステップ 7	option 125 <i>hex</i> 例： デバイス (dhcp-config) # option 125 hex 0000.0009.0a05.08661.7574.6f69.6e73.7461.6c6c.5f64.686370	イメージ ファイルのパスを記述したテキスト ファイルのパスを指定します。
ステップ 8	copy tftp flash <i>filename.txt</i> 例： デバイス (config) # copy tftp flash image.bin	device に、テキスト ファイルをアップロードします。
ステップ 9	copy tftp flash <i>imagename.bin</i> 例： デバイス (config) # copy tftp flash image.bin	device に、新しいイメージの tar ファイルをアップロードします。
ステップ 10	exit 例：	グローバル コンフィギュレーション モードに戻ります。

	コマンドまたはアクション	目的
	デバイス (dhcp-config) # exit	
ステップ 11	tftp-server flash: config.text 例 : デバイス (config) # tftp-server flash:config-boot.text	TFTP サーバ上の Cisco IOS コンフィギュレーション ファイルを指定します。
ステップ 12	tftp-server flash: imagename.bin 例 : デバイス (config) # tftp-server flash:image.bin	TFTP サーバ上のイメージ名を指定します。
ステップ 13	tftp-server flash: filename.txt 例 : デバイス (config) # tftp-server flash:boot-config.text	ダウンロードするイメージファイルの名前を記述したテキスト ファイルを指定します。
ステップ 14	interface interface-id 例 : デバイス (config) # interface gigabitEthernet1/0/4	コンフィギュレーション ファイルを受信するクライアントのアドレスを指定します。
ステップ 15	no switchport 例 : デバイス (config-if) # no switchport	インターフェイスをレイヤ 3 モードにします。
ステップ 16	ip address address mask 例 : デバイス (config-if) # ip address 10.10.10.1 255.255.255.0	IP アドレスとインターフェイスのマスクを指定します。
ステップ 17	end 例 : デバイス (config-if) # end	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 18	copy running-config startup-config 例 : デバイス (config-if) # end	(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。

DHCP サーバからファイルをダウンロードするクライアントの設定



(注) レイヤ3インターフェイスだけを設定してイネーブルにする必要があります。保存されているコンフィギュレーションのDHCPベースの自動設定にIPアドレスを割り当てないでください。

手順の概要

1. **configure terminal**
2. **boot host dhcp**
3. **boot host retry timeout *timeout-value***
4. **banner config-save ^C *warning-message* ^C**
5. **end**
6. **show boot**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : デバイス # 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	boot host dhcp 例 : デバイス (conf) # boot host dhcp	保存されているコンフィギュレーションで自動設定をイネーブルにします。
ステップ 3	boot host retry timeout <i>timeout-value</i> 例 : デバイス (conf) # boot host retry timeout 300	(任意) システムがコンフィギュレーション ファイルをダウンロードしようとする時間を設定します。 (注) タイムアウトを設定しないと、システムは無期限に DHCP サーバから IP アドレスを取得しようとします。

	コマンドまたはアクション	目的
ステップ 4	banner config-save ^C warning-message ^C 例 : デバイス (conf) # banner config-save ^C Caution - Saving Configuration File to NVRAM May Cause You to No longer Automatically Download Configuration Files at Reboot^C	(任意) コンフィギュレーション ファイルを NVRAM に保存しようとするときに表示される警告メッセージを作成します。
ステップ 5	end 例 : デバイス (config-if) # end	特権 EXEC モードに戻ります。
ステップ 6	show boot 例 : デバイス # show boot	設定を確認します。

複数の SVI への IP 情報の手動割り当て

このタスクでは、複数のスイッチ仮想インターフェイス (SVI) に IP 情報を手動で割り当てる方法について説明します。

手順の概要

1. **configure terminal**
2. **interface vlan *vlan-id***
3. **ip address *ip-address subnet-mask***
4. **exit**
5. **ip default-gateway *ip-address***
6. **end**
7. **show interfaces vlan *vlan-id***
8. **show ip redirects**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : デバイス # 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。

	コマンドまたはアクション	目的
ステップ 2	interface vlan <i>vlan-id</i> 例 : デバイス (config) # interface vlan 99	インターフェイス コンフィギュレーション モードを開始して、IP 情報が割り当てられている VLAN を指定します。指定できる範囲は 1 ~ 4094 です。
ステップ 3	ip address <i>ip-address subnet-mask</i> 例 : デバイス (config-vlan) # ip address 10.10.10.2 255.255.255.0	IP アドレスとサブネット マスクを入力します。
ステップ 4	exit 例 : デバイス (config-vlan) # exit	グローバル コンフィギュレーション モードに戻ります。
ステップ 5	ip default-gateway <i>ip-address</i> 例 : デバイス (config) # ip default-gateway 10.10.10.1	<p>device に直接接続しているネクスト ホップのルータ インターフェイスの IP アドレスを入力します。このスイッチにはデフォルトゲートウェイが設定されています。デフォルトゲートウェイは、device スイッチから宛先 IP アドレスを取得していない IP パケットを受信します。</p> <p>デフォルトゲートウェイが設定されると、device は、ホストが接続する必要のあるリモートネットワークに接続できます。</p> <p>(注) IP でルーティングするように device を設定した場合、デフォルトゲートウェイの設定は不要です。</p> <p>(注) デフォルトゲートウェイの構成に基づいて、device の CAPWAP は中継を行い、ルーティングされたアクセスポイントと device の接続をサポートします。</p>
ステップ 6	end 例 : デバイス (config) # end	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
ステップ 7	show interfaces vlan <i>vlan-id</i> 例： デバイス# <code>show interfaces vlan 99</code>	設定された IP アドレスを確認します。
ステップ 8	show ip redirects 例： デバイス# <code>show ip redirects</code>	設定されたデフォルトゲートウェイを確認します。

デバイスのスタートアップコンフィギュレーションの変更

システムコンフィギュレーションを読み書きするためのファイル名の指定

Cisco IOS ソフトウェアは、デフォルトで `config.text` ファイルを使用して、システムコンフィギュレーションの不揮発性コピーを読み書きします。別のファイル名を指定することもできます。次回の起動時には、その名前のファイルが読み込まれます。

始める前に

このタスクではスタンドアロンの device を使用します。

手順の概要

1. `configure terminal`
2. `boot flash:/file-url`
3. `end`
4. `show boot`
5. `copy running-config startup-config`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Switch# <code>configure terminal</code>	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	boot flash:/file-url 例：	次回の起動時に読み込むコンフィギュレーションファイルを指定します。

スイッチの手動による起動

	コマンドまたはアクション	目的
	Switch(config)# boot flash:config.text	<i>file-url</i> : パス (ディレクトリ) およびコンフィギュレーションファイル名。 ファイル名およびディレクトリ名は、大文字と小文字を区別します。
ステップ 3	end 例 : Switch(config)# end	特権 EXEC モードに戻ります。
ステップ 4	show boot 例 : Switch# show boot	入力を確認します。 boot グローバル コンフィギュレーション コマンドによって、CONFIG_FILE 環境変数の設定が変更されます。
ステップ 5	copy running-config startup-config 例 : Switch# 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。

スイッチの手動による起動

スイッチはデフォルトで自動的に起動しますが、手動で起動するように設定することもできます。

始める前に

このタスクのスタンドアロンスイッチを使用します。

手順の概要

1. **configure terminal**
2. **boot manual**
3. **end**
4. **show boot**
5. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 :	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。

	コマンドまたはアクション	目的
	デバイス# <code>configure terminal</code>	
ステップ 2	<p>boot manual</p> <p>例 :</p> <p>デバイス(config)# <code>boot manual</code></p>	<p>次の起動時に、スイッチを手動で起動できるようにします。</p>
ステップ 3	<p>end</p> <p>例 :</p> <p>デバイス(config)# <code>end</code></p>	<p>特権 EXEC モードに戻ります。</p>
ステップ 4	<p>show boot</p> <p>例 :</p> <p>デバイス# <code>show boot</code></p>	<p>入力を確認します。</p> <p>boot manual グローバルコマンドは、MANUAL_BOOT 環境変数の設定を変更します。</p> <p>次回、システムを再起動した際には、スイッチはブートローダモードになり、ブートローダモードであることが <i>switch:</i> プロンプトによって示されます。システムを起動するには、boot filesystem:/file-url ブートローダコマンドを使用します。</p> <ul style="list-style-type: none"> • <i>filesystem</i> : システムボードのフラッシュデバイスに <i>flash:</i> を使用します。 <p>Switch: boot flash:</p> <ul style="list-style-type: none"> • <i>file-url</i> : パス (ディレクトリ) および起動可能なイメージの名前を指定します。 <p>ファイル名およびディレクトリ名は、大文字と小文字を区別します。</p>
ステップ 5	<p>copy running-config startup-config</p> <p>例 :</p> <p>デバイス# <code>copy running-config startup-config</code></p>	<p>(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。</p>

インストールモードでのデバイスのブート

ソフトウェアパッケージのインストール

単一のコマンドまたは個別のコマンドを使用してソフトウェアパッケージをインストールして、アクティブ化し、コミットできます。このタスクでは、ソフトウェアパッケージをインストールするための **install add file activate commit** コマンドの使用方法を示します。

手順の概要

1. **enable**
2. **install add file tftp: filename [activate commit]**
3. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	install add file tftp: filename [activate commit] 例： Device# install add file tftp://172.16.0.1/tftpboot/folder1/ cat9k_iosxe.16.06.01.SPA.bin activate commit	ソフトウェアインストールパッケージをリモートロケーションから（FTP、HTTP、HTTPS、TFTPを介して）デバイスにコピーし、プラットフォームおよびイメージバージョンの互換性チェックを実行し、ソフトウェアパッケージをアクティブ化し、そのパッケージを複数回リロードしても維持されるようにします。 • このコマンドは、.bin ファイルの個別のコンポーネントをサブパッケージと packages.conf ファイルに抽出します。 • このコマンドの実行後にデバイスはリロードします。
ステップ 3	exit 例： Device# exit	特権 EXEC モードを終了し、ユーザー EXEC モードに戻ります。

更新プログラムパッケージの管理

手順の概要

1. **enable**
2. **install add file tftp: filename**
3. **install activate [auto-abort-timer]**

4. **install abort**
5. **install commit**
6. **install rollback to committed**
7. **install remove {file filesystem: filename | inactive}**
8. **show install summary**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>Device> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>install add file tftp: filename</p> <p>例 :</p> <pre>Device# install add file tftp://172.16.0.1//tftpboot/folder1/ cat9k_iosxe.16.06.01.SPA.bin</pre>	<p>リモートロケーションから (FTP、HTTP、HTTPs、TFTP を介して) デバイスにソフトウェアインストールパッケージをコピーし、プラットフォームとイメージのバージョンの互換性チェックを実行します。</p> <ul style="list-style-type: none"> • このコマンドは、.bin ファイルの個別のコンポーネントをサブパッケージと packages.conf ファイルに抽出します。
ステップ 3	<p>install activate [auto-abort-timer]</p> <p>例 :</p> <pre>Device# install activate</pre>	<p>追加のソフトウェア インストール パッケージをアクティブ化し、デバイスをリロードします。</p> <ul style="list-style-type: none"> • ソフトウェアの完全インストールを実行する場合は、パッケージファイル名を指定しないでください。 • auto-abort-timer キーワードがソフトウェアイメージのアクティブ化を自動的にロールバックします。 <p>新しいイメージがアクティブになった後で自動タイマーがトリガーされます。 install commit コマンドを発行する前にタイマーの期限が切れた場合、インストールプロセスは自動的に終了します。デバイスがリロードし、以前のバージョンのソフトウェアイメージで起動します。</p>
ステップ 4	<p>install abort</p> <p>例 :</p> <pre>Device# install abort</pre>	<p>(任意) ソフトウェアインストールのアクティブ化を終了し、現在のインストール手順の前に実行していたバージョンにロールバックします。</p>

デバイスをバンドルモードで起動する場合

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> このコマンドは、イメージがアクティブ化されている状態でのみ使用できます。イメージがコミットされた状態の場合は使用できません。
ステップ 5	install commit 例： Device# install commit	リロードが繰り返されても持続する変更を行います。 <ul style="list-style-type: none"> install commit コマンドで、新しいイメージのインストールを完了します。自動アポートタイマーが期限切れになるまで、複数回のリロード後も変更は維持されます。
ステップ 6	install rollback to committed 例： Device# install rollback to committed	(任意) 最後にコミットしたバージョンに更新をロールバックします。
ステップ 7	install remove {file filesystem: filename inactive} 例： Device# install remove inactive	(任意) 未使用および非アクティブ状態のソフトウェア インストール ファイルを削除します。
ステップ 8	show install summary 例： Device# show install summary	アクティブパッケージに関する情報を表示します。 <ul style="list-style-type: none"> このコマンドの出力は、設定されている install コマンドに応じて変化します。

デバイスをバンドルモードで起動する場合

deviceを起動するには、いくつかの方法があります。1つは、TFTP サーバから bin ファイルをコピーしてdeviceを起動する方法です。または、**boot flash:<image.bin>** コマンドか、**boot usbflash0:<image.bin>** コマンドを使用して、deviceをフラッシュまたはUSBフラッシュから直接起動することもできます。

以下の手順は、バンドルモードで TFTP サーバから device を起動する方法を示します。

手順の概要

1. **switch:BOOT=<source path of .bin file>**
2. **boot**
3. **show version**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	switch:BOOT=<source path of .bin file> 例： switch:BOOT=tftp://10.0.0.2/cat9k_iosxe.16.05.01a.SPA.bin	ブートパラメータを設定します。
ステップ 2	boot 例： switch: boot	deviceをブートします。
ステップ 3	show version	deviceがバンドルモードであることを確認します。

ソフトウェアイメージのリロードのスケジュール設定

このタスクでは、ソフトウェアイメージを後でリロードするようにdeviceを設定する方法について説明します。

手順の概要

1. **configure terminal**
2. **copy running-config startup-config**
3. **reload in [hh:]mm [text]**
4. **reload at hh: mm [month day | day month] [text]**
5. **reload cancel**
6. **show reload**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# <code>configure terminal</code>	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	copy running-config startup-config 例： デバイス# <code>copy running-config startup-config</code>	reload コマンドを使用する前に、device の設定情報をスタートアップコンフィギュレーションに保存します。
ステップ 3	reload in [hh:]mm [text] 例： デバイス# <code>reload in 12</code> System configuration has been modified. Save?	指定した分数、または時間および分数が経過したときに、ソフトウェアがリロードされるようにスケジュールを設定します。リロードは、約 24 日以内に行う必要があります。最大 255 文字で、リロードの理由を指定できます。

	コマンドまたはアクション	目的
	[yes/no]: y	
ステップ 4	reload at hh: mm [month day day month] [text] 例： デバイス(config)# reload at 14:00	リロードを実行する時間を、時間数と分数で指定します。 (注) at キーワードを使用するのは、 device システムクロックが（ネットワークタイムプロトコル、ハードウェアカレンダー、または手動で）設定されている場合だけです。時刻は、 device に設定されたタイムゾーンに基づきます。リロードが複数の devices で同時に行われるようにスケジューリングするには、各 device の時間がNTPと同期している必要があります。
ステップ 5	reload cancel 例： デバイス(config)# reload cancel	以前にスケジューリングされたリロードをキャンセルします。
ステップ 6	show reload 例： show reload	以前 device にスケジューリングされたリロードに関する情報、またはリロードがスケジューリングされているかを表示します。

デバイスのセットアップ設定のモニターリング

例: インストールモードでのソフトウェアブートアップディスプレイ

この例では、インストールモードでのソフトウェアブートアップの表示を示します。

```
switch: boot flash:packages.conf
Attempting to boot from [flash:packages.conf]
Located packages.conf
#

validate_package: SHA-1 hash:
    expected 340D5091:2872A0DD:03E9068C:3FDBECAB:69786462
    calculated 340D5091:2872A0DD:03E9068C:3FDBECAB:69786462
Image parsed from conf file is cat9k-rpboot.16.05.01a.SPA.pkg
#####

Waiting for 120 seconds for other switches to boot
#####
Switch number is 1

Restricted Rights Legend
```

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.5.1a, RELEASE SOFTWARE (fc2)
Technical Support: <http://www.cisco.com/techsupport>
Copyright (c) 1986-2017 by Cisco Systems, Inc.
Compiled Tue 30-May-17 00:36 by mcpre

Cisco IOS-XE software, Copyright (c) 2005-2017 by cisco Systems, Inc. All rights reserved. Certain components of Cisco IOS-XE software are licensed under the GNU General Public License ("GPL") Version 2.0. The software code licensed under GPL Version 2.0 is free software that comes with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such GPL code under the terms of GPL Version 2.0. For more details, see the documentation or "License Notice" file accompanying the IOS-XE software, or the applicable URL provided on the flyer accompanying the IOS-XE software.

FIPS: Flash Key Check : Begin
FIPS: Flash Key Check : End, Not Found, FIPS Mode Not Enabled

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at: <http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to export@cisco.com.

cisco C9300-48P (X86) processor with 818597K/6147K bytes of memory.
Processor board ID FCW2049G03S
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
0K bytes of WebUI ODM Files at webui:.

Base Ethernet MAC Address : 04:6c:9d:01:3b:80
Motherboard Assembly Number : 73-17956-04
Motherboard Serial Number : FOC20465ABU

例: インストールモードでのソフトウェアブートアップディスプレイ

```

Model Revision Number      : P4B
Motherboard Revision Number : 04
Model Number                : C9300-48P
System Serial Number       : FCW2049G03S

```

```
%INIT: waited 0 seconds for NVRAM to be available
```

```
Defaulting CPP : Policer rate for all classes will be set to their defaults
```

```
Press RETURN to get started!
```

この例では、バンドルモードでのソフトウェアブートアップの表示を示します。

```
switch: boot flash:cat9k_iosxe.16.05.01a.SPA.bin
```

```
Attempting to boot from [flash:cat9k_iosxe.16.05.01a.SPA.bin]
```

```
Located cat9k_iosxe.16.05.01a.SPA.bin
```

```
#####
Warning: ignoring ROMMON var "BOOT_PARAM"
```

```
Waiting for 120 seconds for other switches to boot
```

```
#####
Switch number is 3
```

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

```

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

```

```

Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.5.1a,
RELEASE SOFTWARE (fc2)

```

```
Technical Support: http://www.cisco.com/techsupport
```

```
Copyright (c) 1986-2017 by Cisco Systems, Inc.
```

```
Compiled Tue 30-May-17 00:36 by mcpre
```

```

Cisco IOS-XE software, Copyright (c) 2005-2017 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are
licensed under the GNU General Public License ("GPL") Version 2.0. The
software code licensed under GPL Version 2.0 is free software that comes
with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such
GPL code under the terms of GPL Version 2.0. For more details, see the
documentation or "License Notice" file accompanying the IOS-XE software,
or the applicable URL provided on the flyer accompanying the IOS-XE
software.

```

```
FIPS: Flash Key Check : Begin
```

```
FIPS: Flash Key Check : End, Not Found, FIPS Mode Not Enabled

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wvl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

cisco C9300-24U (X86) processor with 818597K/6147K bytes of memory.
Processor board ID FCW2111G00X
2048K bytes of non-volatile configuration memory.
8388608K bytes of physical memory.
1638400K bytes of Crash Files at crashinfo:.
11264000K bytes of Flash at flash:.
15633392K bytes of USB Flash at usbflash0:.
0K bytes of WebUI ODM Files at webui:.

Base Ethernet MAC Address      : 04:6c:9d:1e:2a:80
Motherboard Assembly Number   : 73-17954-05
Motherboard Serial Number     : FOC21094MWL
Model Revision Number         : PP
Motherboard Revision Number   : 05
Model Number                   : C9300-24U
System Serial Number          : FCW2111G00X

%INIT: waited 0 seconds for NVRAM to be available

Defaulting CPP : Policer rate for all classes will be set to their defaults

Press RETURN to get started!
```

例：緊急インストール

以下に、**emergency-install** コマンドが開始された場合の出力サンプルの例を示します。

```
switch: emergency-install
tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin
WARNING: The system partition (bootflash:) will be erased during the system recovery
install process.
Are you sure you want to proceed? [y] y/n [n]: y
Starting system recovery
(tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin) ...
Attempting to boot from [sda9:cat9k-recovery.SSA.bin]
Located cat9k-recovery.SSA.bin
#####
Warning: ignoring ROMMON var "BOOT_PARAM"
```

```

PLATFORM_TYPE C9300 speed 9600

Booting Recovery Image 16.5.1a

Initiating Emergency Installation of bundle
tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin

Downloading bundle
tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
curl_vrf=2
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
 100 485M  100 485M    0     0 5143k      0  0:01:36  0:01:36  --:--:-- 5256k
 100 485M  100 485M    0     0 5143k      0  0:01:36  0:01:36  --:--:-- 5143k

Validating bundle tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Installing bundle
tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Verifying bundle tftp://223.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Package cat9k-cc_srdriver.16.05.01a.SPA.pkg
/temp//stage/cat9k-cc_srdriver.16.05.01a.SPA.pkg is Digitally Signed
Package cat9k-espbase.16.05.01a.SPA.pkg /temp//stage/cat9k-espbase.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-guestshell.16.05.01a.SPA.pkg /temp//stage/cat9k-guestshell.16.05.01a.SPA.pkg
is Digitally Signed
Package cat9k-rpbase.16.05.01a.SPA.pkg /temp//stage/cat9k-rpbase.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-sipbase.16.05.01a.SPA.pkg /temp//stage/cat9k-sipbase.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-sipspa.16.05.01a.SPA.pkg /temp//stage/cat9k-sipspa.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-srdriver.16.05.01a.SPA.pkg /temp//stage/cat9k-srdriver.16.05.01a.SPA.pkg
is Digitally Signed
Package cat9k-webui.16.05.01a.SPA.pkg /temp//stage/cat9k-webui.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-wlc.16.05.01a.SPA.pkg /temp//stage/cat9k-wlc.16.05.01a.SPA.pkg is Digitally
Signed
Package /cat9k-rpboot.16.05.01a.SPA.pkg /temp//rpboot/cat9k-rpboot.16.05.01a.SPA.pkg is
Digitally Signed
Preparing flash...
Flash filesystem unmounted successfully /dev/sdb3
Syncing device...
Emergency Install successful... Rebooting
Will reboot now

Initializing Hardware...

System Bootstrap, Version 16.5.2r, RELEASE SOFTWARE (P)
Compiled Wed 05/31/2017 15:58:35.22 by rel

Current image running:
Primary Rommon Image

Last reset cause: SoftwareReload
C9300-24U platform with 8388608 Kbytes of main memory

```


デバイスのセットアップを実行する場合の設定例

例：更新プログラムパッケージの管理

次に、ソフトウェア パッケージ ファイルを追加する例を示します。

```
Device# install add file flash:cat9k_iosxe.16.06.02.SPA.bin activate commit

install_add_activate_commit: START Mon Oct 30 19:54:51 UTC 2017

System configuration has been modified.
Press Yes(y) to save the configuration and proceed.
Press No(n) for proceeding without saving the configuration.
Press Quit(q) to exit, you may save configuration and re-enter the command. [y/n/q]y
Building configuration...

[OK]Modified configuration has been saved

*Oct 30 19:54:55.633: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 19:54:55 install_engine.sh:

%INSTALL-5-INSTALL_START_INFO: Started install one-shot
flash:cat9k_iosxe.16.06.02.SPA.bininstall_add_activate_commit: Adding PACKAGE

This operation requires a reload of the system. Do you want to proceed?
Please confirm you have changed boot config to flash:packages.conf [y/n]y

--- Starting initial file syncing ---
Info: Finished copying flash:cat9k_iosxe.16.06.02.SPA.bin to the selected switch(es)
Finished initial file syncing

--- Starting Add ---
Performing Add on all members
  [1] Add package(s) on switch 1
  [1] Finished Add on switch 1
Checking status of Add on [1]
Add: Passed on [1]
Finished Add

install_add_activate_commit: Activating PACKAGE
Following packages shall be activated:
/flash/cat9k-wlc.16.06.02.SPA.pkg
/flash/cat9k-webui.16.06.02.SPA.pkg
/flash/cat9k-srdriver.16.06.02.SPA.pkg
/flash/cat9k-sipspa.16.06.02.SPA.pkg
/flash/cat9k-sipbase.16.06.02.SPA.pkg
/flash/cat9k-rpboot.16.06.02.SPA.pkg
/flash/cat9k-rpbase.16.06.02.SPA.pkg
/flash/cat9k-guestshell.16.06.02.SPA.pkg
/flash/cat9k-espbase.16.06.02.SPA.pkg
/flash/cat9k-cc_srdriver.16.06.02.SPA.pkg

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on all members
  [1] Activate package(s) on switch 1
  [1] Finished Activate on switch 1
Checking status of Activate on [1]
Activate: Passed on [1]
Finished Activate
```

```

--- Starting Commit ---
Performing Commit on all members

*Oct 30 19:57:41.145: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 19:57:41 rollback_timer.sh:

%INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Install auto abort timer will expire in
7200 seconds [1]
Commit package(s) on switch 1
  [1] Finished Commit on switch 1
Checking status of Commit on [1]
Commit: Passed on [1]
Finished Commit

Install will reload the system now!
SUCCESS: install_add_activate_commit Mon Oct 30 19:57:48 UTC 2017

Device#
*Oct 30 19:57:48.384: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 19:57:48 install_engine.sh:

%INSTALL-5-INSTALL_COMPLETED_INFO: Completed install one-shot PACKAGE
flash:cat9k_iosxe.16.06.02.SPA.bin

Chassis 1 reloading, reason - Reload command

```

次に、ソフトウェアパッケージファイルをデバイスに追加した後の **show install summary** コマンドの出力例を示します。

```

Device# show install summary

[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
             C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St   Filename/Version
-----
IMG   I    16.6.1.0
IMG   C    16.6.2.0

```

次に、追加したソフトウェアパッケージファイルをアクティブ化する例を示します。

```

Device# install activate

install_activate: START Mon Oct 30 20:14:20 UTC 2017
install_activate: Activating PACKAGE

*Oct 30 20:14:21.379: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:14:21 install_engine.sh:

%INSTALL-5-INSTALL_START_INFO: Started install activateFollowing packages shall be
activated:
/flash/cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-sipspace.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-espspace.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
/flash/cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg

```

```

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on all members
[1] Activate package(s) on switch 1
--- Starting list of software package changes ---
Old files list:
  Removed cat9k-cc_srdriver.16.06.02.SPA.pkg
  Removed cat9k-espbase.16.06.02.SPA.pkg
  Removed cat9k-guestshell.16.06.02.SPA.pkg
  Removed cat9k-rpbase.16.06.02.SPA.pkg
  Removed cat9k-rpboot.16.06.02.SPA.pkg
  Removed cat9k-sipbase.16.06.02.SPA.pkg
  Removed cat9k-sipspa.16.06.02.SPA.pkg
  Removed cat9k-srdriver.16.06.02.SPA.pkg
  Removed cat9k-webui.16.06.02.SPA.pkg
  Removed cat9k-wlc.16.06.02.SPA.pkg
New files list:
  Added cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-espbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
  Added cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Finished list of software package changes
[1] Finished Activate on switch 1
Checking status of Activate on [1]
Activate: Passed on [1]
Finished Activate

*Oct 30 20:15:56.572: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:15:56 rollback_timer.sh:

%INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Install auto abort timer will expire in
7200 seconds
Install will reload the system now!
SUCCESS: install_activate Mon Oct 30 20:16:01 UTC 2017

Device#
*Oct 30 20:16:01.935: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:16:01
install_engine.sh: %INSTALL-5-INSTALL_COMPLETED_INFO: Completed install activate PACKAGE

Chassis 1 reloading, reason - Reload command

```

次に示すのは、**show install summary** コマンドがソフトウェアパッケージのステータスをアクティブでありコミット未完了と表示する場合の出力例です。

```

Device# show install summary

[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
             C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   I   16.6.2.0
IMG   U   16.6.1.0
Device#

```

次の例では、**install commit** コマンドの実行方法を示しています。

```
Device# install commit
install_commit: START Fri Jun 23 21:24:45 IST 2017
install_commit: Committing PACKAGE

--- Starting Commit ---
Performing Commit on Active/Standby
  [R0] Commit package(s) on R0
  [R0] Finished Commit on R0
Checking status of Commit on [R0]
Commit: Passed on [R0]
Finished Commit

SUCCESS: install_commit  Fri Jun 23 21:24:48 IST 2017

Device#
```

次の例は、更新プログラムパッケージを基本パッケージにロールバックする方法を示しています。

```
Device# install rollback to committed

install_rollback: START Mon Oct 30 20:53:33 UTC 2017

This operation requires a reload of the system. Do you want to proceed? [y/n]

*Oct 30 20:53:34.713: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:53:34
install_engine.sh: %INSTALL-5-INSTALL_START_INFO: Started install rollback

--- Starting Rollback ---
Performing Rollback on all members
  [1] Rollback package(s) on switch 1
    --- Starting rollback impact ---
    Changes that are part of this rollback
    Current   : rp 0 0   rp_boot   cat9k-rpboot.16.06.02.prd9.SPA.pkg
    Current   : rp 1 0   rp_boot   cat9k-rpboot.16.06.02.prd9.SPA.pkg
    Replacement: rp 0 0   rp_boot   cat9k-rpboot.16.06.02.SPA.pkg
    Replacement: rp 1 0   rp_boot   cat9k-rpboot.16.06.02.SPA.pkg
    Current   : cc 0 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 0 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 0 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 1 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 1 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 1 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 10 0  cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 10 0  cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 10 0  cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 2 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 2 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 2 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 3 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 3 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 3 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 4 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 4 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 4 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 5 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 5 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 5 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
    Current   : cc 6 0   cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
    Current   : cc 6 0   cc         cat9k-sipbase.16.06.02.prd9.SPA.pkg
    Current   : cc 6 0   cc_spa     cat9k-sipspa.16.06.02.prd9.SPA.pkg
```

```

Current      : cc 7 0 cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
Current      : cc 7 0 cc cat9k-sipbase.16.06.02.prd9.SPA.pkg
Current      : cc 7 0 cc_spa cat9k-sipspa.16.06.02.prd9.SPA.pkg
Current      : cc 8 0 cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
Current      : cc 8 0 cc cat9k-sipbase.16.06.02.prd9.SPA.pkg
Current      : cc 8 0 cc_spa cat9k-sipspa.16.06.02.prd9.SPA.pkg
Current      : cc 9 0 cc_srdriver cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
Current      : cc 9 0 cc cat9k-sipbase.16.06.02.prd9.SPA.pkg
Current      : cc 9 0 cc_spa cat9k-sipspa.16.06.02.prd9.SPA.pkg
Current      : fp 0 0 fp cat9k-espbase.16.06.02.prd9.SPA.pkg
Current      : fp 1 0 fp cat9k-espbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 guestshell cat9k-guestshell.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_base cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_daemons cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_iosd cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_security cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_webui cat9k-webui.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 rp_wlc cat9k-wlc.16.06.02.prd9.SPA.pkg
Current      : rp 0 0 srdriver cat9k-srdriver.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 guestshell cat9k-guestshell.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_base cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_daemons cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_iosd cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_security cat9k-rpbase.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_webui cat9k-webui.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 rp_wlc cat9k-wlc.16.06.02.prd9.SPA.pkg
Current      : rp 1 0 srdriver cat9k-srdriver.16.06.02.prd9.SPA.pkg
Replacement: cc 0 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 0 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 0 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 1 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 1 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 1 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 10 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 10 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 10 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 2 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 2 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 2 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 3 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 3 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 3 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 4 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 4 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 4 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 5 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 5 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 5 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 6 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 6 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 6 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 7 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 7 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 7 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 8 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 8 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 8 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 9 0 cc_srdriver cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 9 0 cc cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 9 0 cc_spa cat9k-sipspa.16.06.02.SPA.pkg
Replacement: fp 0 0 fp cat9k-espbase.16.06.02.SPA.pkg
Replacement: fp 1 0 fp cat9k-espbase.16.06.02.SPA.pkg
Replacement: rp 0 0 guestshell cat9k-guestshell.16.06.02.SPA.pkg
Replacement: rp 0 0 rp_base cat9k-rpbase.16.06.02.SPA.pkg

```

```

Replacement:  rp 0 0  rp_daemons  cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_iosd      cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_security  cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_webui    cat9k-webui.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_wlc      cat9k-wlc.16.06.02.SPA.pkg
Replacement:  rp 0 0  srdriver     cat9k-srdriver.16.06.02.SPA.pkg
Replacement:  rp 1 0  guestshell   cat9k-guestshell.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_base      cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_daemons  cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_iosd      cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_security  cat9k-rpbase.16.06.02.SPA.pkg

```

Chassis 1 reloading, reason - Reload command

```

Replacement:  rp 1 0  rp_webui    cat9k-webui.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_wlc      cat9k-wlc.16.06.02.SPA.pkg
Replacement:  rp 1 0  srdriver     cat9k-srdriver.16.06.02.SPA.pkg

```

Finished rollback impact

```

[1] Finished Rollback on switch 1
Checking status of Rollback on [1]
Rollback: Passed on [1]
Finished Rollback

```

```

Install will reload the system now!
SUCCESS: install_rollback  Mon Oct 30 20:54:23 UTC 2017

```

```

Device#
*Oct 30 20:54:23.576: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:54:23
  install_engine.sh: %INSTALL-5-INSTALL_COMPLETED_INFO: Completed install rollback PACKAGE
*Oct 30 20:54:25.416: %STACKMGR-1-RELOAD: Switch 1 R0/0: stack_mgr:
  Reloading due to reason Reload command Oct 30 20:54:31.615 FP0/0: %PMAN-5-EXITACTION:
  Process manager is exiting: reload fp action requested
Oct 30 20:54

```

次に、**install remove inactive** コマンドの出力例を示します。

```
Device# install remove inactive
```

```

install_remove: START Mon Oct 30 19:51:48 UTC 2017
Cleaning up unnecessary package files
  Scanning boot directory for packages ... done.
  Preparing packages list to delete ...
  done.

```

```

The following files will be deleted:
[switch 1]:
/flash/cat9k-cc_srdriver.16.06.02.SPA.pkg
/flash/cat9k-espbase.16.06.02.SPA.pkg
/flash/cat9k-guestshell.16.06.02.SPA.pkg
/flash/cat9k-rpbase.16.06.02.SPA.pkg
/flash/cat9k-rpboot.16.06.02.SPA.pkg
/flash/cat9k-sipbase.16.06.02.SPA.pkg
/flash/cat9k-sipspa.16.06.02.SPA.pkg
/flash/cat9k-srdriver.16.06.02.SPA.pkg
/flash/cat9k-webui.16.06.02.SPA.pkg
/flash/cat9k-wlc.16.06.02.SPA.pkg
/flash/packages.conf

```

```

Do you want to remove the above files? [y/n]y
[switch 1]:
Deleting file flash:cat9k-cc_srdriver.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-espbase.16.06.02.SPA.pkg ... done.

```

```

Deleting file flash:cat9k-guestshell.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-rpbase.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-rpboot.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-sipbase.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-sipspa.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-srdriver.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-webui.16.06.02.SPA.pkg ... done.
Deleting file flash:cat9k-wlc.16.06.02.SPA.pkg ... done.
Deleting file flash:packages.conf ... done.
SUCCESS: Files deleted.
--- Starting Post_Remove_Cleanup ---
Performing Post_Remove_Cleanup on all members
  [1] Post_Remove_Cleanup package(s) on switch 1
  [1] Finished Post_Remove_Cleanup on switch 1
Checking status of Post_Remove_Cleanup on [1]
Post_Remove_Cleanup: Passed on [1]
Finished Post_Remove_Cleanup

SUCCESS: install_remove  Mon Oct 30 19:52:25 UTC 2017
Device#

```

次に、**install abort** コマンドの出力例を示します。

```

Device# install abort

/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
install_abort: START Mon Oct 30 20:27:32 UTC 2017
install_abort: Abort type PACKAGE subtype NONE smutype NONE

This install abort would require a reload. Do you want to proceed? [y/n]

*Oct 30 20:27:33.189: %INSTALL-5-INSTALL_START_INFO: Switch 1 R0/0: install_engine:
Started install abort
--- Starting Abort ---
Performing Abort on all members
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
  [1] Abort package(s) on switch 1
    --- Starting rollback impact ---
    Changes that are part of this rollback
    Current      : rp 0 0  rp_boot
cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : rp 1 0  rp_boot
cat9k-rpboot.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Replacement: rp 0 0  rp_boot          cat9k-rpboot.16.06.02.SPA.pkg
    Replacement: rp 1 0  rp_boot          cat9k-rpboot.16.06.02.SPA.pkg
    Current      : cc 0 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : cc 0 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : cc 0 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : cc 1 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
    Current      : cc 1 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg

```

例：更新プログラムパッケージの管理

```

Current      : cc 1 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 10 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 10 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 10 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 2 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 2 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 2 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 3 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 3 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 3 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 4 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 4 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 4 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 5 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 5 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 5 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 6 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 6 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 6 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 7 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 7 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 7 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 8 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 8 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 8 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 9 0  cc_srdriver
cat9k-cc_srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 9 0  cc
cat9k-sipbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : cc 9 0  cc_spa
cat9k-sipspa.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : fp 0 0  fp
cat9k-espbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : fp 1 0  fp
cat9k-espbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0  guestshell
cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0  rp_base
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg

```



```

Current      : rp 0 0 rp_daemons
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 rp_iosd
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 rp_security
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 rp_webui
cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 rp_wlc
cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 0 0 srdriver
cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 guestshell
cat9k-guestshell.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_base
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_daemons
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_iosd
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_security
cat9k-rpbase.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_webui
cat9k-webui.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 rp_wlc
cat9k-wlc.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Current      : rp 1 0 srdriver
cat9k-srdriver.BLD_POLARIS_DEV_LATEST_20171029_082249.SSA.pkg
Replacement: cc 0 0 cc_srdriver      cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 0 0 cc                cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 0 0 cc_spa            cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 1 0 cc_srdriver      cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 1 0 cc                cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 1 0 cc_spa            cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 10 0 cc              cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 10 0 cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 10 0 cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 2 0 cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 2 0 cc              cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 2 0 cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 3 0 cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 3 0 cc              cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 3 0 cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 4 0 cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 4 0 cc              cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 4 0 cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 5 0 cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 5 0 cc              cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 5 0 cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 6 0 cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 6 0 cc              cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 6 0 cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 7 0 cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 7 0 cc              cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 7 0 cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 8 0 cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 8 0 cc              cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 8 0 cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
Replacement: cc 9 0 cc_srdriver     cat9k-cc_srdriver.16.06.02.SPA.pkg
Replacement: cc 9 0 cc              cat9k-sipbase.16.06.02.SPA.pkg
Replacement: cc 9 0 cc_spa          cat9k-sipspa.16.06.02.SPA.pkg
Replacement: fp 0 0 fp              cat9k-espbase.16.06.02.SPA.pkg
Replacement: fp 1 0 fp              cat9k-espbase.16.06.02.SPA.pkg
Replacement: rp 0 0 guestshell      cat9k-guestshell.16.06.02.SPA.pkg
    
```

```

Replacement:  rp 0 0  rp_base          cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_daemons       cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_iosd           cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_security       cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_webui         cat9k-webui.16.06.02.SPA.pkg
Replacement:  rp 0 0  rp_wlc           cat9k-wlc.16.06.02.SPA.pkg
Replacement:  rp 0 0  srdriver         cat9k-srdriver.16.06.02.SPA.pkg
Replacement:  rp 1 0  guestshell       cat9k-guestshell.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_base          cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_daemons       cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_iosd           cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_security       cat9k-rpbase.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_webui         cat9k-webui.16.06.02.SPA.pkg
Replacement:  rp 1 0  rp_wlc           cat9k-wlc.16.06.02.SPA.pkg
Replacement:  rp 1 0  srdriver         cat9k-srdriver.16.06.02.SPA.pkg
Finished rollback impact
[1] Finished Abort on switch 1
Checking status of Abort on [1]
Abort: Passed on [1]
Finished Abort

```

```

/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function
[1]: Performing MCU Upgrade Service
/usr/binos/conf/provfunc.sh: line 8792: $!_log_file: ambiguous redirect
SUCCESS: MCU Upgrade Service finished
Install will reload the system now!
SUCCESS: install_abort Mon Oct 30 20:28:21 UTC 2017
/usr/binos/conf/chasutils.sh: line 428: chasfs_is_dominica: readonly function

```

次に、**install activate auto-abort-timer** コマンドの出力例を示します。

```

Device# install activate auto-abort-timer 30

install_activate: START Mon Oct 30 20:42:28 UTC 2017
install_activate: Activating PACKAGE

*Oct 30 20:42:29.149: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:42:29 install_engine.sh:

%INSTALL-5-INSTALL_START_INFO: Started install activateFollowing packages shall be
activated:
/flash/cat9k-wlc.16.06.02.pr9.SPA.pkg
/flash/cat9k-webui.16.06.02.pr9.SPA.pkg
/flash/cat9k-srdriver.16.06.02.pr9.SPA.pkg
/flash/cat9k-sipspace.16.06.02.pr9.SPA.pkg
/flash/cat9k-sipbase.16.06.02.pr9.SPA.pkg
/flash/cat9k-rpboot.16.06.02.pr9.SPA.pkg
/flash/cat9k-rpbase.16.06.02.pr9.SPA.pkg
/flash/cat9k-guestshell.16.06.02.pr9.SPA.pkg
/flash/cat9k-espbase.16.06.02.pr9.SPA.pkg
/flash/cat9k-cc_srdriver.16.06.02.pr9.SPA.pkg

This operation requires a reload of the system. Do you want to proceed? [y/n]y
--- Starting Activate ---
Performing Activate on all members
[1] Activate package(s) on switch 1
--- Starting list of software package changes ---
Old files list:
  Removed cat9k-cc_srdriver.16.06.02.SPA.pkg
  Removed cat9k-espbase.16.06.02.SPA.pkg
  Removed cat9k-guestshell.16.06.02.SPA.pkg
  Removed cat9k-rpbase.16.06.02.SPA.pkg
  Removed cat9k-rpboot.16.06.02.SPA.pkg
  Removed cat9k-sipbase.16.06.02.SPA.pkg

```

```
Removed cat9k-sipspace.16.06.02.SPA.pkg
Removed cat9k-srdriver.16.06.02.SPA.pkg
Removed cat9k-webui.16.06.02.SPA.pkg
Removed cat9k-wlc.16.06.02.SPA.pkg
New files list:
Added cat9k-cc_srdriver.16.06.02.prd9.SPA.pkg
Added cat9k-espbase.16.06.02.prd9.SPA.pkg
Added cat9k-guestshell.16.06.02.prd9.SPA.pkg
Added cat9k-rpbase.16.06.02.prd9.SPA.pkg
Added cat9k-rpboot.16.06.02.prd9.SPA.pkg
Added cat9k-sipbase.16.06.02.prd9.SPA.pkg
Added cat9k-sipspace.16.06.02.prd9.SPA.pkg
Added cat9k-srdriver.16.06.02.prd9.SPA.pkg
Added cat9k-webui.16.06.02.prd9.SPA.pkg
Added cat9k-wlc.16.06.02.prd9.SPA.pkg
Finished list of software package changes
[1] Finished Activate on switch 1
Checking status of Activate on [1]
Activate: Passed on [1]
Finished Activate

*Oct 30 20:43:39.249: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:43:39 rollback_timer.sh:

%INSTALL-5-INSTALL_AUTO_ABORT_TIMER_PROGRESS: Install auto abort timer will expire in
1800 seconds
Install will reload the system now!
SUCCESS: install_activate Mon Oct 30 20:43:44 UTC 2017

Device#
*Oct 30 20:43:44.615: %IOSXE-5-PLATFORM: Switch 1 R0/0: Oct 30 20:43:44 install_engine.sh:

%INSTALL-5-INSTALL_COMPLETED_INFO: Completed install activate PACKAGE
Chassis 1 reloading, reason - Reload command
```

ソフトウェアインストールの確認

手順の概要

1. **enable**
2. **show install log**
3. **show install summary**
4. **show install package *filesystem: filename***
5. **show install active**
6. **show install inactive**
7. **show install committed**
8. **show install uncommitted**

手順の詳細

ステップ 1 enable

例 :

```
Device> enable
```

特権 EXEC モードを有効にします。

- パスワードを入力します（要求された場合）。

ステップ2 show install log

例：

```
Device# show install log
```

デバイスの起動以降に実行されたすべてのソフトウェアインストール動作に関する情報を表示します。

```
Device# show install log
```

```
[0|install_op_boot]: START Sun Jun 11 15:01:37 Universal 2017
[0|install_op_boot]: END SUCCESS Sun Jun 11 15:01:44 Universal 2017
[1|install_commit]: START Mon Jun 12 07:27:31 UTC 2017
[1|install_commit(INFO, )]: Releasing transaction lock...
[1|install_commit(CONSOLE, )]: Committing PACKAGE
[remote|install_commit]: START Mon Jun 12 07:28:08 UTC 2017
[remote|install_commit(INFO, )]: Releasing transaction lock...
[remote|install_commit]: END SUCCESS Mon Jun 12 07:28:41 UTC 2017
[1|install_commit(INFO, )]: [1 2 3]: Performing Commit
SUCCESS: Commit finished
[1|install_commit(INFO, )]: install_commit: START Mon Jun 12 07:28:08 UTC 2017
SUCCESS: install_commit Mon Jun 12 07:28:41 UTC 2017
[1|install_commit(INFO, )]: Remote output from switch 2
[1|install_commit(INFO, )]: install_commit: START Mon Jun 12 07:28:12 UTC 2017
SUCCESS: install_commit Mon Jun 12 07:28:44 UTC 2017
[1|install_commit(INFO, )]: install_commit: START Mon Jun 12 07:28:12 UTC 2017
SUCCESS: install_commit Mon Jun 12 07:28:45 UTC 2017
[1|install_commit]: END SUCCESS Mon Jun 12 07:28:47 UTC 2017
```

ステップ3 show install summary

例：

```
Device# show install summary
```

すべてのメンバ/現場交換可能ユニット（FRU）のイメージのバージョンとそれらに対応するインストール状態に関する情報を表示します。

- このコマンドの出力は、実行した **install** コマンドによって異なります。

```
Device# show install summary
```

```
[ R0 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
```

```
-----
Type  St  Filename/Version
-----
```

```
IMG   I   16.6.2.0
IMG   C   16.6.1.0
```

```
Device#
```

ステップ 4 show install package filesystem: filename

例：

```
Device# show install package flash:cat9k_iosxe.16.06.01.SPA.bin
```

指定したソフトウェア インストール パッケージ ファイルに関する情報を表示します。

```
Device# show install package flash:cat9k_iosxe.16.06.01.SPA.bin
```

```
Package: cat9k_iosxe.16.06.01.SPA.bin
Size: 333806196
Timestamp: Sun Jun 11 14:47:23 2017 UTC
Canonical path: /flash/cat9k_iosxe.16.06.01.SPA.bin

Raw disk-file SHA1sum:
 5e9ef6ed1f7472b35eddd61df300e44b14b65ec4
Header size:      1000 bytes
Package type:     10002
Package flags:    0
Header version:   3

Internal package information:
Name: cc_srdriver
BuildTime:
ReleaseDate: Sun-27-Aug-17-09:05
BootArchitecture: none
RouteProcessor: cat9k
Platform: CAT9K
User: mcpre
PackageName: cc_srdriver
Build: BLD_V166_THROTTLE_LATEST_20170827_090555
CardTypes:

Package is not bootable.
Device#
```

ステップ 5 show install active

例：

```
Device# show install active
```

アクティブなソフトウェア インストール パッケージに関する情報を表示します。

```
Device# show install active
[ R0 ] Active Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
             C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St   Filename/Version
-----
IMG   C    16.6.2.0
```

ステップ 6 show install inactive

例：

```
Device# show install inactive
```

非アクティブなパッケージに関する情報を表示します。

例：DHCP サーバーとしてのデバイスの設定

```
Device# show install inactive

[ R0 ] Inactive Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   I   16.7.1.0
Device#
```

ステップ7 show install committed

例：

```
Device# show install committed
```

コミット済みのパッケージに関する情報を表示します。

```
Device# show install committed

[ R0 ] Committed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   C   16.6.1.0
Device#
```

ステップ8 show install uncommitted

例：

```
Device# show install uncommitted
```

コミットされていないパッケージに関する情報を表示します。

```
Device# show install uncommitted

[ R0 ] Uncommitted Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
           C - Activated & Committed, D - Deactivated & Uncommitted
-----
Type  St  Filename/Version
-----
IMG   U   16.6.2.0
Device#
```

例：DHCP サーバーとしてのデバイスの設定

```
デバイス# configure terminal
デバイス(config)# ip dhcp pool pool1
デバイス(dhcp-config)# network 10.10.10.0 255.255.255.0
デバイス(dhcp-config)# boot config-boot.text
デバイス(dhcp-config)# default-router 10.10.10.1
デバイス(dhcp-config)# option 150 10.10.10.1
```

```
デバイス (dhcp-config) # exit
デバイス (config) # tftp-server flash:config-boot.text
デバイス (config) # interface gigabitethernet1/0/4
デバイス (config-if) # no switchport
デバイス (config-if) # ip address 10.10.10.1 255.255.255.0
デバイス (config-if) # end
```

例：DHCP 自動イメージアップデートの設定

```
デバイス # configure terminal
デバイス (config) # ip dhcp pool pool1
デバイス (dhcp-config) # network 10.10.10.0 255.255.255.0
デバイス (dhcp-config) # boot config-boot.text
デバイス (dhcp-config) # default-router 10.10.10.1
デバイス (dhcp-config) # option 150 10.10.10.1
デバイス (dhcp-config) # option 125 hex
0000.0009.0a05.08661.7574.6f69.6e73.7461.6c6c.5f64.686370
デバイス (dhcp-config) # exit
デバイス (config) # tftp-server flash:config-boot.text
デバイス (config) # tftp-server flash:image_name
デバイス (config) # tftp-server flash:boot-config.text
デバイス (config) # tftp-server flash:autoinstall_dhcp
デバイス (config) # interface gigabitethernet1/0/4
デバイス (config-if) # no switchport
デバイス (config-if) # ip address 10.10.10.1 255.255.255.0
デバイス (config-if) # end
```

例：DHCP サーバーから設定をダウンロードするためのデバイスの設定

次に、VLAN 99 上のレイヤ 3 SVI インターフェイスを使用し、保存されているコンフィギュレーションで DHCP ベースの自動設定をイネーブルにする例を示します。

```
デバイス # configure terminal
デバイス (config) # boot host dhcp
デバイス (config) # boot host retry timeout 300
デバイス (config) # banner config-save ^C Caution - Saving Configuration File to NVRAM May
Cause You to No longer Automatically Download Configuration Files at Reboot^C
デバイス (config) # vlan 99
デバイス (config-vlan) # interface vlan 99
デバイス (config-if) # no shutdown
デバイス (config-if) # end
デバイス # show boot
BOOT path-list:
Config file:          flash:/config.text
Private Config file: flash:/private-config.text
Enable Break:        no
Manual Boot:         no
```

例：ソフトウェアイメージのリロードのスケジューリング

```
HELPER path-list:
NVRAM/Config file
  buffer size:      32768
Timeout for Config
  Download:         300 seconds
Config Download
  via DHCP:         enabled (next boot: enabled)
デバイス#
```

例：ソフトウェアイメージのリロードのスケジューリング

次に、当日の午後 7 時 30 分に、ソフトウェアをdeviceにリロードする例を示します。

```
デバイス# reload at 19:30
Reload scheduled for 19:30:00 UTC Wed Jun 5 2013 (in 2 hours and 25 minutes)
Proceed with reload? [confirm]
```

次に、未来の日時を指定して、ソフトウェアをdeviceにリロードする例を示します。

```
デバイス# reload at 02:00 jun 20
Reload scheduled for 02:00:00 UTC Thu Jun 20 2013 (in 344 hours and 53 minutes)
Proceed with reload? [confirm]
```

デバイスセットアップの実行に関する追加情報

関連資料

関連項目	マニュアルタイトル
デバイスセットアップ コマンド ブート ローダ コマンド	<i>Command Reference (Catalyst 9300 Series Switches)</i>
ハードウェアの設置	<i>Cisco Catalyst 9300 シリーズ スイッチ ハードウェア 設置ガイド</i> .

デバイスセットアップ設定の実行に関する機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	デバイスのセットアップ設定	IP アドレス割り当てと DHCP の自動設定を含むデバイスセットアップ設定を実行できます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 4 章

スマート ライセンスの設定

- スマートライセンシングの設定の前提条件 (107 ページ)
- スマートライセンシングの概要 (107 ページ)
- CSSM への接続 (109 ページ)
- CSSM への既存のライセンスのリンク (111 ページ)
- CSSM への接続の設定とライセンスレベルの設定 (111 ページ)
- CSSM でのデバイスの登録 (124 ページ)
- スマート ライセンスの設定のモニターリング (130 ページ)
- スマートライセンシングの設定例 (131 ページ)
- その他の参考資料 (138 ページ)
- スマートライセンシングに関する機能情報 (138 ページ)

スマートライセンシングの設定の前提条件

CSSM に以下が必要です。

- Cisco スマート アカウント
- 1 つ以上のバーチャルアカウント
- 適切なアクセス権を持つユーザーロール
- デバイスを登録するには、CSSM のスマート ソフトウェア ライセンシング契約に同意する必要があります。
- <https://tools.cisco.com> へのネットワーク到達可能性

スマートライセンシングの概要

シスコ スマート ライセンシングは、シスコ ポートフォリオ全体および組織全体でソフトウェアをより簡単かつ迅速に一貫して購入および管理できる柔軟なライセンスモデルです。また、

これは安全です。ユーザーがアクセスできるものを制御できます。スマートライセンスを使用すると、次のことが可能になります。

- 簡単なアクティベーション：スマートライセンスは、組織全体で使用できるソフトウェアライセンスのプールを確立します。PAK（製品アクティベーションキー）は不要です。
- 管理の統合：My Cisco Entitlements（MCE）は、使いやすいポータルですべてのシスコ製品とサービスの完全なビューを提供します。
- ライセンスの柔軟性：ソフトウェアはハードウェアにノードロックされていないため、必要に応じてライセンスを簡単に使用および転送できます。

スマートライセンスを使用するには、まず Cisco Software Central でスマートアカウントを設定する必要があります（software.cisco.com）。

シスコライセンスの詳細については、cisco.com/go/licensingguide [英語] を参照してください。



- (注) Cisco IOS XE Fuji 16.9.1 以降、ライセンスはスマートライセンスとして管理されます。使用権ライセンスは Cisco IOS XE Fuji 16.9.1 以降廃止されました。

CSSM の概要

CSSM を使用すると、一元化されたポータルからすべてのシスコ スマート ソフトウェア ライセンスを管理できます。CSSM を使用して、バーチャルアカウントと呼ばれるグループ（ライセンスと製品インスタンスの集合体）でライセンスを整理および表示します。

[License] タブの [Smart Software Licensing] リンクをクリックすると、<https://software.cisco.com/#> から CSSM にアクセスできます。



- (注) CSSM にアクセスするには、Chrome 32.0、Firefox 25.0、または Safari 6.0.5 の Web ブラウザを使用します。また、Javascript 1.5 以降のバージョンをブラウザで有効にする必要があります。

CSSM を使用して次のタスクを実行できます。

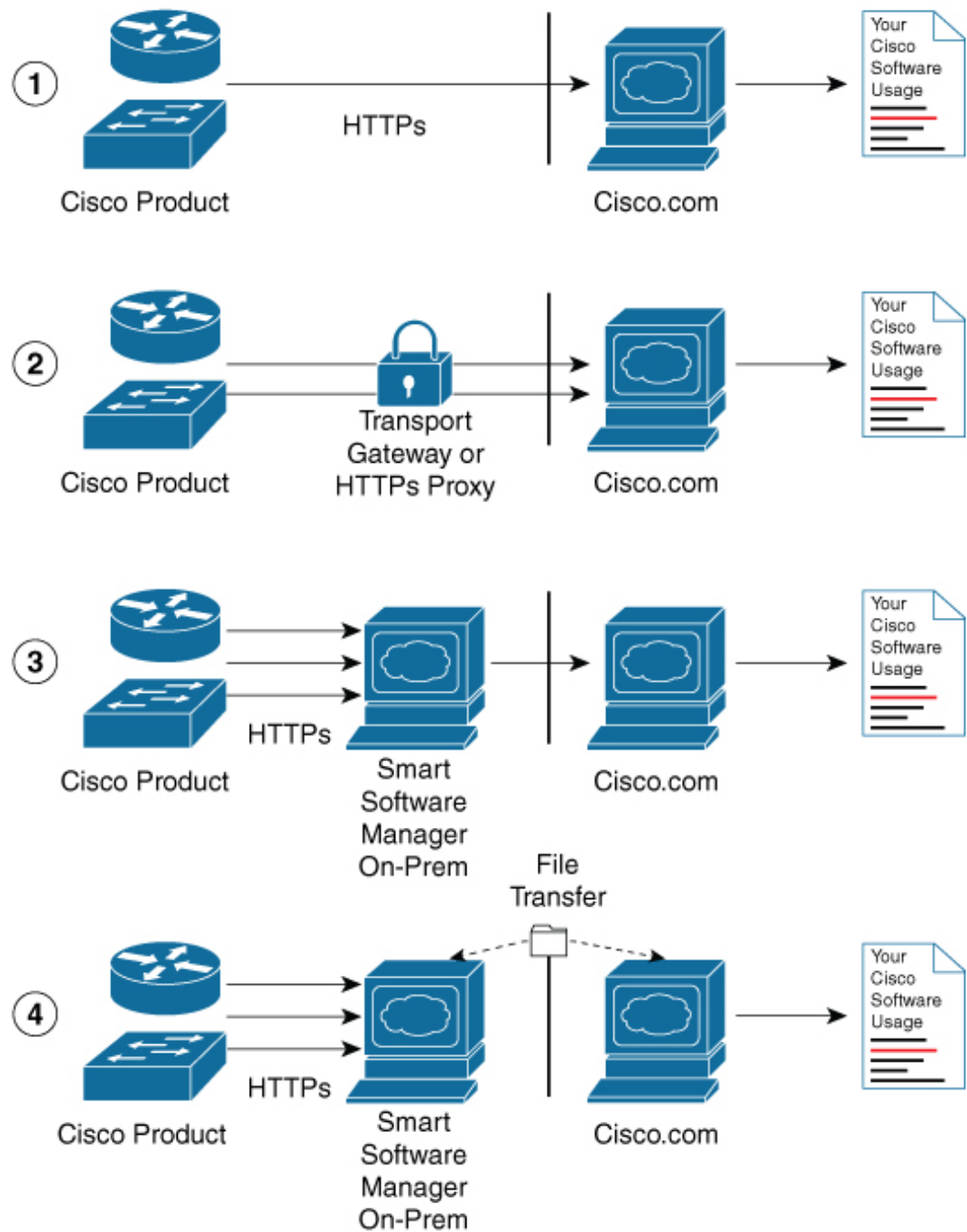
- バーチャルアカウントを作成、管理、または表示する。
- 製品インスタンスの登録トークンを作成および管理する。
- バーチャルアカウント間または表示ライセンス間でライセンスを転送する。
- 製品インスタンスを転送、削除、または表示する。
- バーチャルアカウントに関するレポートを実行する。
- 電子メール通知の設定を変更する。
- 仮想アカウント情報を表示する。

CSSM のヘルプでは、これらのタスクを実行する手順について説明しています。

CSSM への接続

次の図は、CSSM への接続に使用できるさまざまなオプションを示しています。

図 5: [Connection] のオプション



1. **ダイレクトクラウドアクセス：**この方法では、シスコ製品からインターネット経由で Cisco.com に使用状況情報を直接送信します。接続に追加のコンポーネントは必要ありません。

356271

2. HTTPS プロキシを介したダイレクトクラウドアクセス：この方法では、シスコ製から、インターネット経由でプロキシサーバー（Call Home トランスポートゲートウェイまたは市販のプロキシ（Apache など）のいずれか）を介して Cisco.com に使用状況情報を送信します。
3. 接続状態のオンプレミスコレクタを介した間接アクセス：この方法では、シスコ製品から、現地のライセンス認証局として機能するローカルに接続されたコレクタに使用状況情報を送信します。データベースの同期を保つため、周期的にこの情報が交換されます。
4. 非接続状態のオンプレミスコレクタを介した間接アクセス：この方法では、シスコ製品から、現地のライセンス認証局として機能するローカルの接続が解除されたコレクタに使用状況情報を送信します。データベースの同期を保つため、不定期に（月に 1 回など）人による読み取りが可能な情報の交換が実施されます。

オプション 1 と 2 には簡単な接続オプションを、オプション 3 と 4 にはセキュアな環境接続オプションを提供します。Cisco Smart Software Manager オンプレミス（旧称 Cisco Smart Software Manager サテライト）は、オプション 3 と 4 のサポートを提供します。

CSSM への既存のライセンスのリンク

次のセクションは、Cisco スマートアカウントを使用しないで購入したライセンスに必要です。これらのライセンスは、Cisco IOS XE Fuji 16.9.1 へのアップグレード後に CSSM で使用できなくなります。次の電子メールテンプレートを使用して、Cisco Global Licensing Operations（GLO）チームにお問い合わせください。テンプレートに適切な情報を入力し、既存のライセンスを CSSM の Cisco スマートアカウントにリンクするように要求します。

電子メールテンプレート：

宛先：licensing@cisco.com

件名：Request for Linking Existing Licenses to Cisco Smart Account

電子メールの本文：

Cisco.com ID: #####

Smart virtual account name: #####

Smart account domain ID (domain in the form of "xyz.com"): #####

List of UDIs:

List of licenses with count:

Proof of purchase（このメールと一緒に購入証明書を添付してください）

CSSM への接続の設定とライセンスレベルの設定

ここでは、CSSM への接続の設定方法とライセンスレベルの設定方法について説明します。

CSSM への接続の設定

次の手順では、CSSM へのレイヤ 3 接続を設定してネットワーク到達可能性を確認する方法を説明します。CSSM へのレイヤ 3 接続がすでに確立されている場合は、このセクションをスキップしてください。

手順の概要

1. **enable**
2. **configure terminal**
3. **{ip | ipv6} name-server server-address 1 [server-address 2] [server-address 3] [server-address 4] [server-address 5] [server-address 6]**
4. **ip name-server vrf Mgmt-vrf server-address 1 [server-address 2] [server-address 3] [server-address 4] [server-address 5] [server-address 6]**
5. **ip domain lookup source-interface interface-type interface-number**
6. **ip domain name example.com**
7. **ip host tools.cisco.com ip-address**
8. **interface vlan_id**
9. **ntp server ip-address [version number] [key key-id] [prefer]**
10. **switchport access vlan vlan_id**
11. **ip route ip-address ip-mask subnet mask**
12. **license smart transport callhome**
13. **ip http client source-interface interface-type interface-number**
14. **exit**
15. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	configure terminal 例： Device# 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	{ip ipv6} name-server server-address 1 [server-address 2] [server-address 3] [server-address 4] [server-address 5] [server-address 6] 例： Device(config)# ip name-server 209.165.201.1 209.165.200.225 209.165.201.14 209.165.200.230	ドメインネームシステム (DNS) の設定

	コマンドまたはアクション	目的
ステップ 4	<p>ip name-server vrf Mgmt-vrf server-address 1 [server-address 2] [server-address 3] [server-address 4] [server-address 5] [server-address 6]</p> <p>例 :</p> <pre>Device(config)# ip name-server vrf Mgmt-vrf 209.165.201.1 209.165.200.225 209.165.201.14 209.165.200.230</pre>	<p>(任意) VRF インターフェイスで DNS を設定します。</p> <p>(注) このコマンドは、ip name-server コマンドの代わりに設定する必要があります。</p>
ステップ 5	<p>ip domain lookup source-interface interface-type interface-number</p> <p>例 :</p> <pre>Device(config)# ip domain lookup source-interface Vlan100</pre>	(任意) DNS ドメインルックアップ用のソースインターフェイスを設定します。
ステップ 6	<p>ip domain name example.com</p> <p>例 :</p> <pre>Device(config)# ip domain name example.com</pre>	ドメイン名を設定します。
ステップ 7	<p>ip host tools.cisco.com ip-address</p> <p>例 :</p> <pre>Device(config)# ip host tools.cisco.com 209.165.201.30</pre>	(任意) 自動 DNS マッピングが使用できない場合は、DNS ホスト名キャッシュ内のホスト名/アドレス静的マッピングを設定します。
ステップ 8	<p>interface vlan_id</p> <p>例 :</p> <pre>Device(config)# interface Vlan100 Device(config-if)# ip address 192.0.2.10 255.255.255.0 Device(config-if)# exit</pre>	レイヤ 3 インターフェイスを設定します。
ステップ 9	<p>ntp server ip-address [version number] [key key-id] [prefer]</p> <p>例 :</p> <pre>Device(config)# ntp server 198.51.100.100 version 2 prefer</pre>	<p>指定したシステムとのサーバーアソシエーションを形成します。</p> <p>(注) ntp server コマンドは、デバイスの時刻が CSSM と同期されるようにするために必須です。</p>
ステップ 10	<p>switchport access vlan vlan_id</p> <p>例 :</p> <pre>Device(config)# interface GigabitEthernet1/0/1 Device(config-if)# switchport access vlan 100 Device(config-if)# switchport mode access Device(config-if)# exit Device(config)#</pre>	<p>(任意) このアクセスポートがトラフィックを伝送する VLAN を有効にし、非ランキングで非タグ付きのシングル VLAN イーサネットインターフェイスとしてインターフェイスを設定します。</p> <p>(注) このステップは、スイッチポートアクセスモードが必要な場合にのみ設定します。</p>

	コマンドまたはアクション	目的
ステップ 11	ip route <i>ip-address ip-mask subnet mask</i> 例： Device(config)# ip route 192.0.2.0 255.255.255.255 192.0.2.1	デバイスにルートを設定します。 (注) スタティックルートまたはダイナミックルートのいずれかを設定できます。
ステップ 12	license smart transport callhome 例： Device(config)# license smart transport callhome	転送モードを Call Home として有効にします。 (注) license smart transport callhome コマンドは必須です。
ステップ 13	ip http client source-interface <i>interface-type interface-number</i> 例： Device(config)# ip http client source-interface vlan100	HTTP クライアントのソースインターフェイスを設定します。 (注) ip http client source-interface interface-type interface-number コマンドは必須です。
ステップ 14	exit 例： Device(config)# exit	(任意) グローバル コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 15	copy running-config startup-config 例： Device# 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。

ダイレクトクラウドアクセス用の Call Home サービスの設定



(注) デフォルトでは、CiscoTAC-1 プロファイルはすでにデバイスに設定されています。プロファイルのステータスを確認するには、**show call-home profile all** コマンドを使用します。

Call Home サービスは、CSSM に対してクリティカルなシステムイベントを電子メールおよび Web 上で通知します。

Call Home サービスを設定して有効にするには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **call-home**
4. **no http secure server-identity-check**
5. **contact-email-address email-address**

6. **profile CiscoTAC-1**
7. **destination transport-method http**
8. **destination address http url**
9. **active**
10. **no destination transport-method email**
11. **exit**
12. **exit**
13. **service call-home**
14. **exit**
15. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	configure terminal 例： Device# 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	call-home 例： Device(config)# call-home	CallHome コンフィギュレーションモードを開始します。
ステップ 4	no http secure server-identity-check 例： Device(config-call-home)# no http secure server-identity-check	HTTP 接続の確立時のサーバー ID チェックを無効にします。
ステップ 5	contact-email-address email-address 例： Device(config-call-home)# contact-email-addr username@example.com	顧客の電子メールアドレスを割り当てます。電子メールアドレスフォーマットには、スペースなしで最大 200 文字まで入力できます。
ステップ 6	profile CiscoTAC-1 例： Device(config-call-home)# profile CiscoTAC-1	デフォルトでは、CiscoTAC-1 プロファイルは非アクティブです。このプロファイルを CallHome サービスで使用するには、プロファイルを有効にする必要があります。
ステップ 7	destination transport-method http 例： Device(config-call-home-profile)# destination transport-method http	HTTP 経由の Call Home サービスを有効にします。

	コマンドまたはアクション	目的
ステップ 8	destination address http url 例： Device(config-call-home-profile)# destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService	CSSM に接続します。
ステップ 9	active 例： Device(config-call-home-profile)# active	宛先プロファイルをイネーブルにします。
ステップ 10	no destination transport-method email 例： Device(config-call-home-profile)# no destination transport-method email	電子メールによる Call Home サービスを無効にします。
ステップ 11	exit 例： Device(config-call-home-profile)# exit	Call Home 宛先プロファイルコンフィギュレーションモードを終了して、Call Home コンフィギュレーションモードに戻ります。
ステップ 12	exit 例： Device(config-call-home)# exit	Call Home コンフィギュレーションモードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 13	service call-home 例： Device(config)# service call-home	Call Home 機能をイネーブルにします。
ステップ 14	exit 例： Device(config)# exit	グローバルコンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。
ステップ 15	copy running-config startup-config 例： Device# 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。

HTTPS プロキシサーバを介したダイレクトクラウドアクセス用の Call Home サービスの設定

Call Home サービスは、HTTPS プロキシサーバを介して設定できます。この設定では、CSSM への接続にユーザ認証は必要ありません。



(注) 認証された HTTPS プロキシ設定はサポートされていません。

HTTPS プロキシを介して CallHome サービスを設定して有効にするには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **call-home**
4. **contact-email-address** *email-address*
5. **http-proxy** *proxy-address* **proxy-port** *port-number*
6. **profile** **CiscoTAC-1**
7. **destination transport-method** **http**
8. **no destination transport-method** **email**
9. **profile** *name*
10. **reporting smart-licensing-data**
11. **destination transport-method** **http**
12. **destination address** **http** *url*
13. **active**
14. **exit**
15. **exit**
16. **service call-home**
17. **ip http client proxy-server** *proxy-address* **proxy-port** *port-number*
18. **exit**
19. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	configure terminal 例： Device# 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	call-home 例： Device(config)# call-home	CallHome コンフィギュレーションモードを開始します。

HTTPS プロキシサーバを介したダイレクトクラウドアクセス用の Call Home サービスの設定

	コマンドまたはアクション	目的
ステップ 4	contact-email-address <i>email-address</i> 例： Device(config-call-home)# contact-email-address sch-smart-licensing@cisco.com	デフォルトの電子メールアドレスを sch-smart-licensing@cisco.com として設定します。
ステップ 5	http-proxy <i>proxy-address</i> proxy-port <i>port-number</i> 例： Device(config-call-home)# http-proxy 198.51.100.10 port 3128	Call Home サービスへのプロキシサーバ情報を設定します。
ステップ 6	profile CiscoTAC-1 例： Device(config-call-home)# profile CiscoTAC-1	デフォルトでは、CiscoTAC-1 プロファイルは非アクティブです。このプロファイルを Call Home サービスで使用するには、プロファイルを有効にする必要があります。
ステップ 7	destination transport-method http 例： Device(config-call-home-profile)# destination transport-method http	HTTP 経由の Call Home サービスを有効にします。
ステップ 8	no destination transport-method email 例： Device(config-call-home-profile)# no destination transport-method email	電子メールによる Call Home サービスを無効にします。
ステップ 9	profile <i>name</i> 例： Device(config-call-home)# profile test1	指定された宛先プロファイル名の Call Home 宛先プロファイル コンフィギュレーション モードを開始します。指定された宛先プロファイルが存在しない場合、作成されます。
ステップ 10	reporting smart-licensing-data 例： Device(config-call-home-profile)# reporting smart-licensing-data	HTTP 経由の Call Home サービスとのデータ共有を有効にします。
ステップ 11	destination transport-method http 例： Device(config-call-home-profile)# destination transport-method http	HTTP メッセージの転送方法をイネーブルにします。
ステップ 12	destination address http <i>url</i> 例： Device(config-call-home-profile)# destination address http https://tools.cisco.com/its/service/oddce/services/DDCEService	CSSM に接続します。

	コマンドまたはアクション	目的
ステップ 13	active 例： Device(config-call-home-profile)# active	宛先プロファイルをイネーブルにします。
ステップ 14	exit 例： Device(config-call-home-profile)# exit	Call Home 宛先プロファイルコンフィギュレーションモードを終了して、Call Home コンフィギュレーションモードに戻ります。
ステップ 15	exit 例： Device(config-call-home)# exit	Call Home コンフィギュレーションモードを終了し、グローバルコンフィギュレーションモードに戻ります。
ステップ 16	service call-home 例： Device(config)# service call-home	Call Home 機能をイネーブルにします。
ステップ 17	ip http client proxy-server proxy-address proxy-port port-number 例： Device(config)# ip http client proxy-server 198.51.100.10 port 3128	Call Home 機能をイネーブルにします。
ステップ 18	exit 例： Device(config)# exit	グローバルコンフィギュレーションモードを終了し、特権 EXEC モードに戻ります。
ステップ 19	copy running-config startup-config 例： Device# 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。

Cisco Smart Software Manager オンプレミス用の Call Home サービスの設定

Smart Software Manager オンプレミス (旧称 Cisco Smart Software Manager サテライト) については、<https://www.cisco.com/c/en/us/buy/smart-accounts/software-manager-satellite.html> を参照してください。

Cisco Smart Software Manager オンプレミス (旧称 Cisco Smart Software Manager サテライト) 用の Call Home サービスを設定するには、次の手順を実行します。

手順の概要

1. **enable**

2. **configure terminal**
3. **call-home**
4. **no http secure server-identity-check**
5. **profile *name***
6. **reporting smart-licensing-data**
7. **destination transport-method http**
8. **destination address http *url***
9. **destination preferred-msg-format {*long-text* | *short-text* | *xml*}**
10. **active**
11. **exit**
12. **exit**
13. **ip http client source-interface *interface-type interface-number***
14. **crypto pki trustpoint *name***
15. **revocation-check none**
16. **end**
17. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# 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	call-home 例： Device(config)# call-home	Call Home コンフィギュレーションモードを開始します。
ステップ 4	no http secure server-identity-check 例： Device(config-call-home)# no http secure server-identity-check	HTTP 接続の確立時のサーバー ID チェックを無効にします。
ステップ 5	profile <i>name</i> 例： Device(config-call-home)# profile test1	指定された宛先プロファイル名の Call Home 宛先プロファイル コンフィギュレーション モードを開始します。指定された宛先プロファイルが存在しない場合、作成されます。
ステップ 6	reporting smart-licensing-data 例：	HTTP 経由の Call Home サービスとのデータ共有を有効にします。

	コマンドまたはアクション	目的
	Device (config-call-home-profile) # reporting smart-licensing-data	
ステップ 7	destination transport-method http 例： Device (config-call-home-profile) # destination transport-method http	HTTP メッセージの転送方法をイネーブルにします。
ステップ 8	destination address http url 例： Device (config-call-home-profile) # destination address http https://209.165.201.15:443/Transportgateway/services/DeviceRequestHandler または Device (config-call-home-profile) # destination address http http://209.165.201.15:80/Transportgateway/services/DeviceRequestHandler	Call Home メッセージが送信される宛先 URL (CSSM) を設定します。 注 宛先 URL の IP アドレスまたは完全修飾ドメイン名 (FQDN) が、Cisco Smart Software Manager オンプレミスの [Satellite Name] に設定されている IP アドレスまたは FQDN と一致することを確認します。
ステップ 9	destination preferred-msg-format {long-text short-text xml} 例： Device (config-call-home-profile) # destination preferred-msg-format xml	(任意) 使用するメッセージ形式を設定します。デフォルトは XML です。
ステップ 10	active 例： Device (config-call-home-profile) # active	宛先プロファイルをイネーブルにします。デフォルトでは、プロファイルは作成時にイネーブルになります。
ステップ 11	exit 例： Device (config-call-home-profile) # exit	Call Home 宛先プロファイルコンフィギュレーションモードを終了して、Call Home コンフィギュレーションモードに戻ります。
ステップ 12	exit 例： Device (config-call-home) # exit	Call Home コンフィギュレーションモードを終了し、グローバル コンフィギュレーションモードに戻ります。
ステップ 13	ip http client source-interface interface-type interface-number 例： Device (config) # ip http client source-interface Vlan100	HTTP クライアントのソースインターフェイスを設定します。 (注) ip http client source-interface interface-type interface-number コマンドは、vrf インターフェイスでは必須です。

	コマンドまたはアクション	目的
ステップ 14	crypto pki trustpoint name 例： Device(config)# crypto pki trustpoint SLA-TrustPoint	(任意) トラストポイントおよび設定された名前を宣言して、CA トラストポイント コンフィギュレーション モードを開始します。
ステップ 15	revocation-check none 例： Device(ca-trustpoint)# revocation-check none	(任意) 証明書の確認が無視されることを指定します。
ステップ 16	end 例： Device(ca-trustpoint)# end	(任意) CA トラストポイント コンフィギュレーション モードを終了し、特権 EXEC モードに戻ります。
ステップ 17	copy running-config startup-config 例： Device# 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。

ライセンスレベルの設定

この手順は任意です。次の手順を使用すると、以下のことができます。

- ライセンスのダウングレードとアップグレード
- 評価ライセンスと拡張ライセンスの有効化と無効化
- アップグレードライセンスのクリア

登録する前に、必要なライセンスレベルをデバイスで設定する必要があります。Cisco Catalyst 9000 シリーズ スイッチで使用できるライセンスレベルは次のとおりです。

基本ライセンス

- Network Essentials
- Network Advantage (Network Essentials を含む)

アドオンライセンス：3年、5年、または7年の固定期間にわたって次のライセンスをサブスクリプションでできます。

- Digital Networking Architecture (DNA) Essentials
- DNA Advantage (DNA Essentials を含む)

ライセンスレベルを設定するには、次の手順に従います。

手順の概要

1. enable

2. **configure terminal**
3. **license boot level** *license_level*
4. **exit**
5. **write memory**
6. **show version**
7. **reload**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	configure terminal 例： デバイス# 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	license boot level <i>license_level</i> 例： デバイス(config)# license boot level network-essentials	スイッチのライセンスをアクティブ化します。
ステップ 4	exit 例： デバイス(config)# exit	特権 EXEC モードに戻ります。
ステップ 5	write memory 例： デバイス# write memory	ライセンス情報をスイッチに保存します。
ステップ 6	show version 例： デバイス# show version Technology-package Current Type Technology-package Next reboot network-essentials Smart License network-essentials None Subscription Smart License None	ライセンスレベルの情報を表示します。
ステップ 7	reload 例：	デバイスがリロードされます。

	コマンドまたはアクション	目的
	デバイス# <code>reload</code>	

CSSM でのデバイスの登録

CSSM でデバイスを登録するには、次のタスクを実行する必要があります。

1. CSSM から一意のトークンを生成します。
2. 生成されたトークンを使用してデバイスを登録します。

登録が成功すると、デバイスで ID 証明書を受信します。この証明書はデバイスに保存され、それ以降のシスコとのすべての通信で自動的に使用されます。CSSM は 30 日ごとに登録情報の更新を試みます。

また、ライセンスの使用状況データが収集され、毎月レポートが送信されます。必要に応じて、機密情報（ホスト名、ユーザー名、パスワードなど）が使用状況レポートから除外されるように Call Home 設定を構成できます。



- (注) Cisco IOS XE Fuji 16.9.1 から以前のリリースにデバイスをダウングレードすると、スマートライセンスは従来のライセンスに移行されます。デバイス上のすべてのスマートライセンス情報が削除されます。デバイスを Cisco IOS XE Fuji 16.9.1 に再びアップグレードする必要がある場合、デバイスが CSSM に再登録されるまで、ライセンスステータスは評価モードのままです。

CSSM からの新しいトークンの生成

新しい製品インスタンスをバーチャルアカウントに登録するために、トークンが生成されます。

- ステップ 1 <https://software.cisco.com/#> から CSSM にログインします。
シスコから提供されたユーザー名とパスワードを使用してポータルにログインする必要があります。
- ステップ 2 [Inventory] タブをクリックします。
- ステップ 3 [Virtual Account] ドロップダウンリストから、必要なバーチャルアカウントを選択します。
- ステップ 4 [General] タブをクリックします。
- ステップ 5 [New Token] をクリックします。

Cisco Software Central > Smart Software Licensing

English [Change] Hello, Smart Account Name

Smart Software Licensing

Alerts | Inventory | License Conversion | Reports | Preferences | Satellites | Activity

Virtual Account: **Virtual Account 1**

23 Major | 9 Minor | Hide Alerts

General | Licenses | Product Instances | Event Log

Virtual Account

Description: Account 1
Default Virtual Account: No

Product Instance Registration Tokens

The registration tokens below can be used to register new product instances to this virtual account.

New Token...

Token	Expiration Date	Description	Export-Controlled	Created By	Actions
ZjgxNzdjYjctOWRhMCO0M2I0L...	Expired	Token 1	Allowed	User 1	Actions
ZTg2MjBjMzUIN2U0Ni00NDdkL...	Expired		Allowed	User 1	Actions

[Create Registration Token] ウィンドウが表示されます。

ステップ 6 [Description] フィールドに、トークンの説明を入力します。

ステップ 7 [Expire After] フィールドに、トークンをアクティブにする必要がある日数を入力します。

ステップ 8 (任意) [Max. Number of Uses] フィールドに、トークンの有効期限が切れるまでの最大使用回数を入力します。

Create Registration Token

This will create a token that is used to register product instances, so that they can use licenses from this virtual account. Once it's created, go to the Smart Licensing configuration for your products and enter the token, to register them with this virtual account.

Virtual Account: Virtual Account 1

Description: Token 2

* Expire After: 30 Days
Between 1 - 365, 30 days recommended

Max. Number of Uses:

The token will be expired when either the expiration or the maximum uses is reached

Allow export-controlled functionality on the products registered with this token

Create Token Cancel

ステップ 9 [Allow export-controlled functionality on the products registered with this token] チェックボックスをオンにします。

このチェックボックスをオンにすると、シスコは米国および各国固有のエクスポートポリシーおよびガイドラインに準拠するようになります。詳細については、<https://www.cisco.com/c/en/us/about/legal/global-export-trade.html>を参照してください。

ステップ 10 [Create Token] をクリックしてトークンを作成します。

ステップ 11 トークンを作成したら、[Copy] をクリックし、新しく作成したトークンをコピーします。

Create Registration Token

This will create a token that is used to register product instances, so that they can use licenses from this virtual account. Once it's created, go to the Smart Licensing configuration for your products and enter the token, to register them with this virtual account.

Virtual Account: Virtual Account 1

Description:

* Expire After: Days
Between 1 - 365, 30 days recommended

Max. Number of Uses:

The token will be expired when either the expiration or the maximum uses is reached

Allow export-controlled functionality on the products registered with this token ⓘ

新しいトークンを使用するデバイスの登録

新しいトークンを使用してデバイスを登録するには、次の手順を実行します。

手順の概要

1. **enable**
2. **license smart register idtoken token_ID**
3. **write memory**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	license smart register idtoken token_ID 例：	CSSM から生成されたトークンを使用して、デバイスをバックエンドサーバーに登録します。

	コマンドまたはアクション	目的
	デバイス# <code>license smart register idtoken</code> <code>\$T14UytrcNXBzbEslck8veUtWag5abnZJOFdDa1FwbVRa%0Ab1FMbz0%3D%0A</code>	
ステップ 3	write memory 例 : デバイス# <code>write memory</code>	ライセンス情報をデバイスに保存します。

登録後のライセンスステータスの確認

登録後にライセンスのステータスを確認するには、**show license all** コマンドを使用します。

```
Device> enable
Device# show license all
Smart Licensing Status
=====

Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Smart Account: Smart Account Name
  Virtual Account: Virtual Account 1
  Export-Controlled Functionality: Allowed
  Initial Registration: SUCCEEDED on Jul 13 09:30:40 2018 EDT
  Last Renewal Attempt: None
  Next Renewal Attempt: Jan 09 09:30:40 2019 EDT
  Registration Expires: Jul 13 09:25:31 2019 EDT

License Authorization:
  Status: AUTHORIZED on Jul 13 09:30:45 2018 EDT
  Last Communication Attempt: SUCCEEDED on Jul 13 09:30:45 2018 EDT
  Next Communication Attempt: Aug 12 09:30:45 2018 EDT
  Communication Deadline: Oct 11 09:25:40 2018 EDT

Utility:
  Status: DISABLED

Data Privacy:
  Sending Hostname: yes
  Callhome hostname privacy: DISABLED
  Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED

Transport:
  Type: Callhome

License Usage
=====

C9300 DNA Advantage (C9300-24 DNA Advantage):
  Description: C9300-24P DNA Advantage
  Count: 3
  Version: 1.0
  Status: AUTHORIZED

C9300 Network Advantage (C9300-24 Network Advantage):
```

```

Description: C9300-24P Network Advantage
Count: 3
Version: 1.0
Status: AUTHORIZED

Product Information
=====
UDI: PID:C9300-24U,SN:FCW2125L046

HA UDI List:
  Active:PID:C9300-24U,SN:FCW2125L046
  Standby:PID:C9300-24U,SN:FCW2125L03U
  Member:PID:C9300-24U,SN:FCW2125G01T

Agent Version
=====
Smart Agent for Licensing: 4.4.13_rel/116
Component Versions: SA:(1_3_dev)1.0.15, SI:(dev22)1.2.1, CH:(rel15)1.0.3, PK:(dev18)1.0.3

Reservation Info
=====
License reservation: DISABLED

```

CSSM でのデバイスの登録キャンセル

デバイスがインベントリから移された場合、再導入のために別の場所に出荷された場合、または返品許可（RMA）プロセスを使用して交換のためにシスコに返送された場合は、**deregister** コマンドを使用してデバイスの登録をキャンセルできます。

デバイス登録をキャンセルするには、次の手順に従います。

始める前に

デバイスを正常に登録解除するには、CSSM へのレイヤ 3 接続が使用可能である必要があります。

手順の概要

1. **enable**
2. **license smart deregister**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 プロンプトが表示されたらパスワードを入力します。
ステップ 2	license smart deregister 例： Device# license smart deregister	デバイスの登録をキャンセルし、デバイスを評価モードに送信します。対応するプラットフォームのすべてのスマートライセンス資格と証明書が削除さ

	コマンドまたはアクション	目的
		れます。CSSM に保存されているデバイス製品インスタンスも削除されます。

スマートライセンスの設定のモニターリング

スマートライセンスの設定をモニターするには、特権 EXEC モードで次のコマンドを使用します。

表 6: スマートライセンスの設定をモニターリングするコマンド

コマンド	目的
<p>show license status</p>	<p>スマートライセンスのコンプライアンスステータスを表示します。以下は、表示される可能性があるステータスのリストです。</p> <ul style="list-style-type: none"> • Enabled : スマートライセンスが有効になっていることを示します。 • Waiting : デバイスがライセンス権限付与要求を行った後の初期状態を示します。デバイスはシスコとの通信を確立し、CSSM に正常に登録されます。 • Registered : デバイスが CSSM と通信できること、およびライセンス権限付与の要求を開始する権限を持っていることを示します。 • Authorized : デバイスがコンプライアンスステータスであり、要求されたライセンスのタイプおよび数を使用する権限があることを示します。承認ステータスのライフタイムは90日です。承認を更新するため、デバイスは30日後にCSSM に新しい権限承認要求を送信します。 • Out Of Compliance : 1つ以上のライセンスがコンプライアンス違反になっていることを示します。追加ライセンスを購入する必要があります。 • Eval Mode : (デバイスの使用後) 90日以内にCSSM にデバイスを登録する必要があります。登録しない場合、デバイスの評価期間が終了します。 • Evaluation Period Expired : デバイスが登録されていない場合、デバイスは90日後に評価期間終了モードになります。

コマンド	目的
show license all	使用中のすべての権限を表示します。さらに、関連付けられているライセンス証明書、コンプライアンスステータス、UDI、およびその他の詳細が表示されます。
show tech-support license	詳細なデバッグ出力を表示します。
show license usage	ライセンスの使用情報を表示します。
show license summary	すべてのアクティブなライセンスの要約を表示します。

スマートライセンシングの設定例

ここでは、さまざまなスマートライセンスの設定例を示します。

例：Call Home プロファイルの表示

例

Call Home プロファイルを表示するには、**show call-home profile all** コマンドを使用します。

```
Device> enable
Device# show call-home profile all
Profile Name: CiscoTAC-1
  Profile status: ACTIVE
  Profile mode: Full Reporting
  Reporting Data: Smart Call Home, Smart Licensing
  Preferred Message Format: xml
  Message Size Limit: 3145728 Bytes
  Transport Method: http
  HTTP address(es): https://tools.cisco.com/its/service/oddce/services/DDCEService
  Other address(es): default

  Periodic configuration info message is scheduled every 1 day of the month at 09:15

  Periodic inventory info message is scheduled every 1 day of the month at 09:00

Alert-group          Severity
-----
crash                debug
diagnostic           minor
environment          warning
inventory            normal

Syslog-Pattern      Severity
-----
```

例：登録前のライセンス情報の表示

```

APF-.-WLC_.*          warning
.*                    major

```

例：登録前のライセンス情報の表示

例

ライセンスの付与資格を表示するには、**show license all** コマンドを使用します。

```

Device> enable
Device# show license all
Load for five secs: 1%/0%; one minute: 0%; five minutes: 0%
No time source, 09:28:07.210 EDT Fri Jul 13 2018

Smart Licensing Status
=====

Smart Licensing is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 68 days, 0 hours, 30 minutes, 5 seconds

Utility:
  Status: DISABLED

Data Privacy:
  Sending Hostname: yes
  Callhome hostname privacy: DISABLED
  Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED

Transport:
  Type: Callhome

License Usage
=====

(C9300-24 DNA Advantage):
  Description:
  Count: 3
  Version: 1.0
  Status: EVAL MODE

(C9300-24 Network Advantage):
  Description:
  Count: 3
  Version: 1.0
  Status: EVAL MODE

Product Information
=====
UDI: PID:C9300-24U,SN:FCW2125L046

HA UDI List:

```

```
Active:PID:C9300-24U,SN:FCW2125L046
Standby:PID:C9300-24U,SN:FCW2125L03U
Member:PID:C9300-24U,SN:FCW2125G01T

Agent Version
=====
Smart Agent for Licensing: 4.4.13_rel/116
Component Versions: SA:(1_3_dev)1.0.15, SI:(dev22)1.2.1, CH:(rel5)1.0.3, PK:(dev18)1.0.3

Reservation Info
=====
License reservation: DISABLED
```

例

ライセンス使用情報を表示するには、**show license usage** コマンドを使用します。

```
Device> enable
Device# show license usage
Load for five secs: 1%/0%; one minute: 0%; five minutes: 0%
No time source, 09:28:34.123 EDT Fri Jul 13 2018

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 68 days, 0 hours, 29 minutes, 38 seconds

(C9300-24 DNA Advantage):
  Description:
  Count: 3
  Version: 1.0
  Status: EVAL MODE

(C9300-24 Network Advantage):
  Description:
  Count: 3
  Version: 1.0
  Status: EVAL MODE
```

例

すべてのライセンスの概要を表示するには、**show license summary** コマンドを使用します。

```
Device> enable
Device# show license summary
Load for five secs: 1%/0%; one minute: 0%; five minutes: 0%
No time source, 09:28:39.986 EDT Fri Jul 13 2018

Smart Licensing is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 68 days, 0 hours, 29 minutes, 33 seconds
```

例：デバイスの登録

```
License Usage:
License                Entitlement tag                Count Status
-----
(C9300-24 DNA Advantage)    3 EVAL MODE
(C9300-24 Network Advan...) 3 EVAL MODE
```

例

ライセンスのステータス情報を表示するには、**show license status** コマンドを使用します。

```
Device> enable
Device# show license status
Load for five secs: 1%/0%; one minute: 0%; five minutes: 0%
No time source, 09:28:37.683 EDT Fri Jul 13 2018

Smart Licensing is ENABLED

Utility:
  Status: DISABLED

Data Privacy:
  Sending Hostname: yes
  Callhome hostname privacy: DISABLED
  Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED

Transport:
  Type: Callhome

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 68 days, 0 hours, 29 minutes, 35 seconds
```

例：デバイスの登録

例

デバイスを登録するには、**license smart register idtoken** コマンドを使用します。

```
Device> enable
Device# license smart register idtoken
T14UytrNXBzbEslck8veUtWaG5abnZJOFdDalFwbVRa%0Ab1RMBz0%3D%0A
Device# write memory
```

例：登録後のライセンスステータスの表示

例

ライセンスの付与資格を表示するには、**show license all** コマンドを使用します。

```
Device> enable
Device# show license all
Load for five secs: 0%/0%; one minute: 2%; five minutes: 1%
No time source, 09:31:16.387 EDT Fri Jul 13 2018

Smart Licensing Status
=====

Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Smart Account: Smart Account Name
  Virtual Account: Virtual Account 1
  Export-Controlled Functionality: Allowed
  Initial Registration: SUCCEEDED on Jul 13 09:30:40 2018 EDT
  Last Renewal Attempt: None
  Next Renewal Attempt: Jan 09 09:30:40 2019 EDT
  Registration Expires: Jul 13 09:25:31 2019 EDT

License Authorization:
  Status: AUTHORIZED on Jul 13 09:30:45 2018 EDT
  Last Communication Attempt: SUCCEEDED on Jul 13 09:30:45 2018 EDT
  Next Communication Attempt: Aug 12 09:30:45 2018 EDT
  Communication Deadline: Oct 11 09:25:40 2018 EDT

Utility:
  Status: DISABLED

Data Privacy:
  Sending Hostname: yes
  Callhome hostname privacy: DISABLED
  Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED

Transport:
  Type: Callhome

License Usage
=====

C9300 DNA Advantage (C9300-24 DNA Advantage):
  Description: C9300-24P DNA Advantage
  Count: 3
  Version: 1.0
  Status: AUTHORIZED

C9300 Network Advantage (C9300-24 Network Advantage):
  Description: C9300-24P Network Advantage
  Count: 3
  Version: 1.0
  Status: AUTHORIZED

Product Information
```

例：登録後のライセンスステータスの表示

```

=====
UDI: PID:C9300-24U,SN:FCW2125L046

HA UDI List:
  Active:PID:C9300-24U,SN:FCW2125L046
  Standby:PID:C9300-24U,SN:FCW2125L03U
  Member:PID:C9300-24U,SN:FCW2125G01T

Agent Version
=====
Smart Agent for Licensing: 4.4.13_rel/116
Component Versions: SA:(1_3_dev)1.0.15, SI:(dev22)1.2.1, CH:(rel5)1.0.3, PK:(dev18)1.0.3

Reservation Info
=====
License reservation: DISABLED

```

例

ライセンス使用情報を表示するには、**show license usage** コマンドを使用します。

```

Device> enable
Device# show license usage
License Authorization:
  Status: AUTHORIZED on Jul 17 09:47:28 2018 EDT

C9300 DNA Advantage (C9300-24 DNA Advantage):
  Description: C9300-24P DNA Advantage
  Count: 3
  Version: 1.0
  Status: AUTHORIZED

C9300 Network Advantage (C9300-24 Network Advantage):
  Description: C9300-24P Network Advantage
  Count: 3
  Version: 1.0
  Status: AUTHORIZED

```

例

すべてのライセンスの概要を表示するには、**show license summary** コマンドを使用します。

```

Device> enable
Device# show license summary
Load for five secs: 1%/0%; one minute: 1%; five minutes: 1%
No time source, 09:32:13.746 EDT Fri Jul 13 2018

Smart Licensing is ENABLED

Registration:
  Status: REGISTERED
  Smart Account: Smart Account Name
  Virtual Account: Virtual Account 1
  Export-Controlled Functionality: Allowed
  Last Renewal Attempt: None
  Next Renewal Attempt: Jan 09 09:30:40 2019 EDT

```



```
License Authorization:
  Status: AUTHORIZED
  Last Communication Attempt: SUCCEEDED
  Next Communication Attempt: Aug 12 09:30:44 2018 EDT
```

```
License Usage:
  License                               Entitlement tag                               Count Status
  -----
  C9300 DNA Advantage                   (C9300-24 DNA Advantage)                   3 AUTHORIZED
  C9300 Network Advantage (C9300-24 Network Advan... 3 AUTHORIZED
```

例

ライセンスのステータス情報を表示するには、**show license status** コマンドを使用します。

```
Device> enable
Device# show license status
Load for five secs: 0%/0%; one minute: 1%; five minutes: 1%
No time source, 09:32:00.191 EDT Fri Jul 13 2018

Smart Licensing is ENABLED

Utility:
  Status: DISABLED

Data Privacy:
  Sending Hostname: yes
  Callhome hostname privacy: DISABLED
  Smart Licensing hostname privacy: DISABLED
  Version privacy: DISABLED

Transport:
  Type: Callhome

Registration:
  Status: REGISTERED
  Smart Account: Smart Account Name
  Virtual Account: Virtual Account 1
  Export-Controlled Functionality: Allowed
  Initial Registration: SUCCEEDED on Jul 13 09:30:40 2018 EDT
  Last Renewal Attempt: None
  Next Renewal Attempt: Jan 09 09:30:40 2019 EDT
  Registration Expires: Jul 13 09:25:31 2019 EDT

License Authorization:
  Status: AUTHORIZED on Jul 13 09:30:45 2018 EDT
  Last Communication Attempt: SUCCEEDED on Jul 13 09:30:45 2018 EDT
  Next Communication Attempt: Aug 12 09:30:45 2018 EDT
  Communication Deadline: Oct 11 09:25:40 2018 EDT
```

その他の参考資料

関連資料

関連項目	マニュアル タイトル
Cisco Smart Software Manager のヘルプ	Smart Software Manager Help
Cisco Smart Software Manager オンプレミス	Cisco Smart Software Manager On-Prem

シスコのテクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	http://www.cisco.com/support

スマートライセンシングに関する機能情報

次の表に、このモジュールで説明した機能に関するリリース情報を示します。この表は、ソフトウェア リリース トレインで各機能のサポートが導入されたときのソフトウェア リリースだけを示しています。その機能は、特に断りが無い限り、それ以降の一連のソフトウェア リリースでもサポートされます。

プラットフォームのサポートおよびシスコ ソフトウェア イメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator にアクセスするには、www.cisco.com/go/cfn に移動します。Cisco.com のアカウントは必要ありません。

表 7: スマートライセンシングに関する機能情報

機能名	リリース	機能情報
スマートライセンス	Cisco IOS XE Fuji 16.9.1	スマート ライセンスは、クラウドベースのソフトウェアライセンス管理ソリューションであり、ライセンス、ハードウェア、およびソフトウェアの使用状況の傾向を管理および追跡できます。



第 5 章

有線ネットワークでの Application Visibility and Control の設定

- 機能情報の確認 (141 ページ)
- 有線ネットワークでの Application Visibility and Control について (142 ページ)
- サポートされる AVC クラス マップおよびポリシー マップのフォーマット (142 ページ)
- 有線 Application Visibility and Control の制限 (144 ページ)
- Application Visibility and Control の設定方法 (145 ページ)
- Application Visibility and Control のモニターリング (172 ページ)
- 例：Application Visibility and Control の設定 (173 ページ)
- 基本的なトラブルシューティング：質問と回答 (183 ページ)
- Application Visibility and Control に関する追加情報 (184 ページ)
- 有線ネットワークでの Application Visibility and Control の機能履歴 (185 ページ)

機能情報の確認

ご使用のソフトウェアリリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェア リリースの Bug Search Tool およびリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、Cisco Feature Navigator を使用します。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> からアクセスします。Cisco.com のアカウントは必要ありません。

有線ネットワークでの Application Visibility and Control について

Application Visibility and Control (AVC) は、アプリケーションへの適応力やアプリケーションへのインテリジェンス性に基づいて、厳密なパケットおよび接続からブランチおよびキャンパスソリューションを発展させるためのシスコの取り組みの重要な部分です。Application Visibility and Control (AVC) は、ネットワークベースのアプリケーション認識 (NBAR2) エンジンによるディープパケットインスペクション技術を使用してアプリケーションを分類します。AVC は、スタンドアロンスイッチおよびスイッチスタックの有線アクセスポート上に設定できます。NBAR2 は、プロトコル検出を有効にすることによって明示的に、または **match protocol** 分類子を含む QoS ポリシーを接続することによって暗黙的に、インターフェイス上でアクティブにできます。有線 AVC Flexible Netflow (FNF) をインターフェイス上に設定し、インターフェイスごとのクライアント、サーバー、アプリケーションの統計情報を提供できます。このレコードは、Easy Performance Monitor (Easy perf-mon または ezPM) の **application-statistics** および **application-performance** プロファイルで利用できる **application-client-server-stats** トラフィック監視と同様です。

サポートされる AVC クラス マップおよびポリシー マップのフォーマット

サポートされる AVC クラス マップのフォーマット

クラスマップのフォーマット	クラスマップの例	方向
<code>match protocol protocol name</code>	<code>class-map match-any NBAR-VOICE match protocol ms-lync-audio</code>	入力と出力の両方
組み合わせフィルタ	<code>class-map match-any NBAR-VOICE match protocol ms-lync-audio match dscp ef</code>	入力と出力の両方

サポートされる AVC ポリシーのフォーマット

ポリシーのフォーマット	QoS 処理
<code>match protocol</code> フィルタに基づく出力ポリシー	マークおよびポリシー
<code>match protocol</code> フィルタに基づく入力ポリシー	マークおよびポリシー

次の表で、AVC ポリシーの詳細なフォーマット、および例について説明します。

AVC ポリシーのフォーマット	AVC ポリシーの例	方向
ベーシック セット	<pre>policy-map MARKING-IN class NBAR-MM_CONFERENCING set dscp af41</pre>	入力および出力
ベーシック ポリシー	<pre>policy-map POLICING-IN class NBAR-MM_CONFERENCING police cir 600000 set dscp af41</pre>	入力および出力
ベーシック セットおよびポリシー	<pre>policy-map webex-policy class webex-class set dscp ef police 5000000</pre>	入力および出力
デフォルトを含む複数のセットおよびポリシー	<pre>policy-map webex-policy class webex-class set dscp af31 police 4000000 class class-webex-category set dscp ef police 6000000 class class-default set dscp <></pre>	入力および出力
階層型ポリシー	<pre>policy-map webex-policy class webex-class police 5000000 service-policy client-in-police-only policy-map client-in-police-only class webex-class police 100000 class class-webex-category set dscp ef police 200000</pre>	入力および出力
階層型セットおよびポリシー	<pre>policy-map webex-policy class class-default police 1500000 service policy client-up-child policy-map client-up-child class webex-class police 100000 set dscp ef class class-webex-category police 200000 set dscp af31</pre>	

有線 Application Visibility and Control の制限

- NBAR 対応 QoS ポリシー設定は有線物理ポートでのみ許可されます。ポリシー設定は、VLAN およびその他の論理インターフェイスなどの仮想インターフェイスではサポートされていません。
 - NBAR ベースの QoS ポリシー設定は、ポートチャネルメンバーポートおよび SVI やサブインターフェイスなどの仮想インターフェイスではサポートされません。
 - NBAR ベースの QoS ポリシー設定は、レイヤ 2 アクセスポートとトランクポート、およびレイヤ 3 ルーテッドポートでサポートされます。
 - NBAR と送信 (Tx) スイッチドポートアナライザ (SPAN) は、同じインターフェイスではサポートされません。
 - プロトコルベースまたは属性ベースのいずれかのポートに同時に接続できるのは、NBAR ベースの QoS メカニズムの 1 つだけです。次の 2 つの属性のみがサポートされます。
 - traffic-class
 - business-relevance
 - 従来の WDAVC QoS の制限事項は引き続き適用されます。
 - マーキングとポリシングのみがサポートされます。
 - 物理インターフェイスだけがサポートされます。
 - アプリケーション分類がオフラインで行われるため、QoS 分類には遅延があります (ただし、フローの最初のパケットは、正確な QoS 分類の前に転送されます)。
 - NBAR2 ベースの一致基準 **match protocol** は、マーキングアクションおよびポリシングアクションでのみ許可されます。NBAR2 一致基準は、キューイング機能が設定されているポリシーでは許可されません。
 - 「一致プロトコル」：すべてのポリシーで最大 255 の同時に異なるプロトコル (8 ビットの HW 制限)。
 - AVC は管理ポート (Gig 0/0) ではサポートされていません。
 - IPv6 パケットの分類はサポートされていません。
 - IPv4 ユニキャスト (TCP/UDP) のみがサポートされます。
 - Web UI : Web UI からアプリケーションの可視性を設定し、アプリケーションのモニタリングを実行できます。アプリケーション制御は、CLI を使用してのみ実行できます。Web UI ではサポートされていません。
- Web UI 上で有線 AVC のトラフィックを管理、またはチェックするには、最初に CLI を使用して **ip http authentication local** と **ip nbar http-service** コマンドを設定する必要があります。

- NBAR および ACL のロギングは、同一スイッチ上で一緒に設定することはできません。
- プロトコル検出、アプリケーションベースの QoS、および有線 AVC FNF は、非アプリケーションベース FNF がある同一インターフェイス上で同時に設定することはできません。ただし、これらの有線 AVC 機能は、相互に設定できます。たとえば、プロトコル検出、アプリケーションベースの QoS、および有線 AVC FNF は、同一インターフェイス上で同時に設定できます。
- AVC と暗号化トラフィック分析 (ETA) を同じインターフェイスで同時に設定することはできません。
- 接続は、物理 Layer2 (アクセス/トランク) および Layer3 ポートでのみ行う必要があります。アップリンクは、単一のアップリンクであり、ポートチャネルの一部でなければ接続できます。
- パフォーマンス：各スイッチ メンバーは、50% 未満の CPU 使用率で、1 秒あたり 500 の接続 (CPS) を処理できます。
- 拡張性：48 個のアクセス ポートごとに最大 10,000 の双方向フローと、24 個のアクセス ポートごとに 5000 の双方向フローを処理できます。(アクセス ポートごとに ~ 200 フロー)。
- 有線 AVC では、この章の手順にリストされている固定のフィールドセットのみを使用できます。その他の組み合わせは使用できません。通常の FNF フローモニターでは、他の組み合わせも使用できます (サポートされている FNF フィールドのリストについては、『*Network Management Configuration Guide*』の「Configuring Flexible NetFlow」の章を参照してください)。

Application Visibility and Control の設定方法

有線ネットワークでの Application Visibility and Control の設定

有線ポートで Application Visibility and Control を設定するには、次の手順を実行します。

可視性の設定

- インターフェイス コンフィギュレーション モードで **ip nbar protocol-discovery** コマンドを使用してインターフェイス上でプロトコル検出を有効にすることで、NBAR2 エンジン をアクティブ化します。 [インターフェイスでのアプリケーション認識の有効化 \(146 ページ\)](#) を参照してください。

制御設定：次の手順に従って、アプリケーションに基づいて QoS ポリシーを設定します。

1. AVC QoS ポリシーの作成。 [AVC QoS ポリシーの作成 \(147 ページ\)](#) を参照してください。
2. インターフェイスへの AVC QoS ポリシーの適用。 [スイッチポートへの QoS ポリシーの適用 \(150 ページ\)](#) を参照してください。

アプリケーションベースの Flexible Netflow の設定 :

- フローにキー フィールドおよび非キー フィールドを指定して、フロー レコードを作成します。 [フロー レコードの作成 \(150 ページ\)](#) を参照してください。
- フローエクスポートを作成してフローレコードをエクスポートします。 [フローエクスポートの作成 \(163 ページ\)](#) を参照してください。
- フロー レコードおよびフロー エクスポートに基づいて、フロー モニターを作成します。 [フロー モニターの作成 \(164 ページ\)](#) を参照してください。
- インターフェイスにフローモニターを接続します。 [インターフェイスへのフローモニターの関連付け \(166 ページ\)](#) を参照してください。

プロトコル検出、アプリケーションベースの QoS およびアプリケーションベースの FNF は、すべて独立した機能です。単独で設定することも、または同じインターフェイスで同時に設定することもできます。

インターフェイスでのアプリケーション認識の有効化

インターフェイス上でアプリケーション認識をイネーブルにするには、次の手順を実行します。

手順の概要

1. **configure terminal**
2. **interfaceinterface-id**
3. **ip nbar protocol-discovery**
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : デバイス# 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	interfaceinterface-id 例 : デバイス (config)# interface gigabitethernet 1/0/1	プロトコル検出をイネーブルにするインターフェイスを指定し、インターフェイス コンフィギュレーション モードを開始します。

	コマンドまたはアクション	目的
ステップ 3	ip nbar protocol-discovery 例： デバイス (config-if) # ip nbar protocol-discovery	NBAR2 エンジンを実アクティブ化することで、インターフェイスでアプリケーション認識を有効にします。
ステップ 4	end 例： デバイス (config-if) # end	特権 EXEC モードに戻ります。

AVC QoS ポリシーの作成

AVC QoS ポリシーを作成するには、次の一般的な手順を実行します。

1. match protocol フィルタでクラス マップを作成します。
2. ポリシー マップを作成します。
3. インターフェイスにポリシー マップを適用します。

クラス マップの作成

match protocol フィルタを設定する前に、クラス マップを作成する必要があります。マーキングやポリシングなどの QoS アクションをトラフィックに適用できます。AVC の match protocol フィルタは、有線アクセスポートに適用されます。サポートされているプロトコルの詳細については、http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/prot_lib/config_library/nbar-prot-pack-library.html を参照してください。

手順の概要

1. **configure terminal**
2. **class-map class-map-name**
3. **match protocol application-name**
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# 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。

	コマンドまたはアクション	目的
ステップ 2	class-map <i>class-map-name</i> 例： デバイス(config)# class-map webex-class	クラス マップを作成します。
ステップ 3	match protocol <i>application-name</i> 例： デバイス(config)# class-map webex-class デバイス(config-cmap)# match protocol webex-media	アプリケーション名との一致を指定します。
ステップ 4	end 例： Device(config)# end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバルコンフィギュレーションモードを終了できます。

ポリシー マップの作成

手順の概要

1. **configure terminal**
2. **policy-map** *policy-map-name*
3. **class** [*class-map-name* | **class-default**]
4. **police** *rate-bps burst-byte*
5. **set** {*dscp new-dscp* | **cos** *cos-value*}
6. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# 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	policy-map <i>policy-map-name</i> 例： デバイス(config)# policy-map webex-policy	<p>ポリシーマップ名を入力することによってポリシーマップを作成し、ポリシーマップコンフィギュレーションモードを開始します。</p> <p>デフォルトでは、ポリシーマップは定義されていません。</p> <p>ポリシーマップのデフォルトの動作では、パケットが IP パケットの場合は DSCP が 0 に、パケットがタグ付きの場合は CoS が 0 に設定されます。ポリシーマップは実行されません。</p>

	コマンドまたはアクション	目的
		<p>(注) 既存のポリシーマップを削除するには、no policy-map policy-map-name 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。</p>
<p>ステップ 3</p>	<p>class [<i>class-map-name</i> class-default]</p> <p>例 :</p> <p>デバイス (config-pmap) # class webex-class</p>	<p>トラフィックの分類を定義し、ポリシーマップクラス コンフィギュレーション モードを開始します。</p> <p>デフォルトでは、ポリシーマップおよびクラスマップは定義されていません。</p> <p>すでに class-map 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トラフィッククラスが定義されている場合は、このコマンドで <i>class-map-name</i> にその名前を指定します。</p> <p>class-default トラフィッククラスは定義済みで、どのポリシーにも追加できます。このトラフィッククラスは、常にポリシー マップの最後に配置されます。暗黙の match any が class-default クラスに含まれている場合、他のトラフィッククラスと一致しないパケットはすべて class-default と一致します。</p> <p>(注) 既存のクラスマップを削除するには、no class class-map-name ポリシーマップ コンフィギュレーション コマンドを使用します。</p>
<p>ステップ 4</p>	<p>police rate-bps burst-byte</p> <p>例 :</p> <p>デバイス (config-pmap-c) # police 100000 80000</p>	<p>分類したトラフィックにポリサーを定義します。</p> <p>デフォルトでは、ポリサーは定義されていません。</p> <ul style="list-style-type: none"> • <i>rate-bps</i> には、平均トラフィック レートをビット/秒 (bps) で指定します。指定できる範囲は 8000 ~ 10000000000 です • <i>burst-byte</i> には、標準バースト サイズをバイト数で指定します。有効範囲は、1000 ~ 512000000 です。
<p>ステップ 5</p>	<p>set {dscp new-dscp cos cos-value}</p> <p>例 :</p> <p>デバイス (config-pmap-c) # set dscp 45</p>	<p>パケットに新しい値を設定することによって、IP トラフィックを分類します。</p> <ul style="list-style-type: none"> • dscp new-dscp には、分類されたトラフィックに割り当てる新しい DSCP 値を入力します。指定できる範囲は 0 ~ 63 です。

■ スイッチポートへの QoS ポリシーの適用

	コマンドまたはアクション	目的
ステップ 6	end 例： Device(config)# end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバルコンフィギュレーションモードを終了できます。

スイッチポートへの QoS ポリシーの適用

手順の概要

1. **configure terminal**
2. **interface interface-id**
3. **service-policy input policymapname**
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# 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	interface interface-id 例： デバイス(config)# interface Gigabitethernet 1/0/1	インターフェイス 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	service-policy input policymapname 例： デバイス(config-if)# service-policy input MARKING_IN	インターフェイスにローカル ポリシーを適用します。
ステップ 4	end 例： Device(config)# end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバルコンフィギュレーションモードを終了できます。

有線 AVC Flexible Netflow の設定

フローレコードの作成

有線 AVC FNF は、従来の双方向フローレコードと方向性フローレコード（入力と出力）の 2 種類の定義済みフローレコードをサポートします。合計 4 つの異なる定義済みフローレコード（2 つの双方向フローレコードと 2 つの方向性フローレコード）を設定し、フローモニターに関連付けることができます。従来の双方向レコードはクライアント/サーバーアプリケーション

ン統計情報レコードであり、新しい方向性レコードは入出力のアプリケーション統計情報です。

- [双方向フローレコード \(151 ページ\)](#)
- [方向性フローレコード \(158 ページ\)](#)

双方向フローレコード

フローレコード 1 : 双方向フローレコード

手順の概要

1. **configure terminal**
2. **flow record** *flow_record_name*
3. **description** *description*
4. **match ipv4 version**
5. **match ipv4 protocol**
6. **match application name**
7. **match connection client ipv4 address**
8. **match connection server ipv4 address**
9. **match connection server transport port**
10. **match flow observation point**
11. **collect flow direction**
12. **collect connection initiator**
13. **collect connection new-connections**
14. **collect connection client counter packets long**
15. **collect connection client counter bytes network long**
16. **collect connection server counter packets long**
17. **collect connection server counter bytes network long**
18. **collect timestamp absolute first**
19. **collect timestamp absolute last**
20. **end**
21. **show flow record**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : Device# 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	flow record <i>flow_record_name</i> 例 : デバイス (config)# flow record fr-wdavic-1	フローレコードコンフィギュレーションモードを開始します。

	コマンドまたはアクション	目的
ステップ 3	description <i>description</i> 例 : デバイス (config-flow-record) # description fr-wdavic-1	(任意) フローレコードの説明を作成します。
ステップ 4	match ipv4 version 例 : デバイス (config-flow-record) # match ipv4 version	IPv4 ヘッダーからの IP バージョンとの一致を指定します。
ステップ 5	match ipv4 protocol 例 : デバイス (config-flow-record) # match ipv4 protocol	IPv4 プロトコルとの一致を指定します。
ステップ 6	match application name 例 : デバイス (config-flow-record) # match application name	アプリケーション名との一致を指定します。 (注) この操作は、AVC サポートでは必須です。フローがアプリケーションと一致することが可能になるためです。
ステップ 7	match connection client ipv4 address 例 : デバイス (config-flow-record) # match connection client ipv4 address	クライアント (フローイニシエータ) の IPv4 アドレスとの一致を指定します。
ステップ 8	match connection server ipv4 address 例 : デバイス (config-flow-record) # match connection server ipv4 address	サーバー (フローレスポンド) の IPv4 アドレスとの一致を指定します。
ステップ 9	match connection server transport port 例 : デバイス (config-flow-record) # match connection server transport port	サーバーのトランスポートポートとの一致を指定します。
ステップ 10	match flow observation point 例 : デバイス (config-flow-record) # match flow observation point	フロー観測メトリックの観測ポイント ID との一致を指定します。
ステップ 11	collect flow direction 例 : デバイス (config-flow-record) # collect flow direction	次の手順で collect connection initiator コマンドの initiator キーワードで指定される双方向フローの関連する側 (イニシエータまたはレスポンド) の方向 (入力または出力) を収集するように指定します。

	コマンドまたはアクション	目的
		<p>initiator キーワードで指定される値に応じて、flow direction キーワードは次の値をとります。</p> <ul style="list-style-type: none"> • 0x01 = 入力フロー • 0x02 = 出力フロー <p>initiator キーワードがイニシエータに設定されている場合、フローの方向はフローのイニシエータ側から指定されます。initiator キーワードがレスポндаに設定されている場合、フローの方向はフローのレスポнда側から指定されます。有線 AVC では、initiator キーワードは常にイニシエータに設定されています。</p>
ステップ 12	<p>collect connection initiator</p> <p>例 :</p> <pre>デバイス (config-flow-record)# collect connection initiator</pre>	<p>collect flow direction コマンドで指定されたフローの方向に関連するフローの側（イニシエータまたはレスポнда）を収集するように指定します。initiator キーワードは、フローの方向に関する次の情報を提供します。</p> <ul style="list-style-type: none"> • 0x01 = イニシエータ : フローの送信元は接続のイニシエータです <p>有線 AVC では、initiator キーワードは常にイニシエータに設定されています。</p>
ステップ 13	<p>collect connection new-connections</p> <p>例 :</p> <pre>デバイス (config-flow-record)# collect connection new-connections</pre>	<p>観測された接続開始の数を収集するように指定します。</p>
ステップ 14	<p>collect connection client counter packets long</p> <p>例 :</p> <pre>デバイス (config-flow-record)# collect connection client counter packets long</pre>	<p>クライアントが送信したパケット数を収集するように指定します。</p>
ステップ 15	<p>collect connection client counter bytes network long</p> <p>例 :</p> <pre>デバイス (config-flow-record)# collect connection client counter bytes network long</pre>	<p>クライアントが送信したバイト数の合計を収集するように指定します。</p>
ステップ 16	<p>collect connection server counter packets long</p> <p>例 :</p> <pre>デバイス (config-flow-record)# collect connection server counter packets long</pre>	<p>サーバーが送信したパケット数を収集するように指定します。</p>

フローレコード 2 : 双方向フローレコード

	コマンドまたはアクション	目的
ステップ 17	collect connection server counter bytes network long 例 : デバイス (config-flow-record)# collect connection server counter bytes network long	サーバーが送信したバイト数の合計を収集するように指定します。
ステップ 18	collect timestamp absolute first 例 : デバイス (config-flow-record)# collect timestamp absolute first	最初のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。
ステップ 19	collect timestamp absolute last 例 : デバイス (config-flow-record)# collect timestamp absolute last	最新のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。
ステップ 20	end 例 : Device(config)# end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。
ステップ 21	show flow record 例 : デバイス # show flow record	すべてのフローレコードに関する情報を表示します。

フローレコード 2 : 双方向フローレコード

手順の概要

1. **configure terminal**
2. **flow record** *flow_record_name*
3. **description** *description*
4. **match ipv4 version**
5. **match ipv4 protocol**
6. **match application name**
7. **match connection client ipv4 address**
8. **match connection client transport port**
9. **match connection server ipv4 address**
10. **match connection server transport port**
11. **match flow observation point**
12. **collect flow direction**
13. **collect connection initiator**
14. **collect connection new-connections**
15. **collect connection client counter packets long**
16. **collect connection client counter bytes network long**

17. **collect connection server counter packets long**
18. **collect connection server counter bytes network long**
19. **collect timestamp absolute first**
20. **collect timestamp absolute last**
21. **end**
22. **show flow record**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# 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	flow record flow_record_name 例： デバイス (config)# flow record fr-wdavic-1	フローレコード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	description description 例： デバイス (config-flow-record)# description fr-wdavic-1	(任意) フローレコードの説明を作成します。
ステップ 4	match ipv4 version 例： デバイス (config-flow-record)# match ipv4 version	IPv4 ヘッダーからの IP バージョンとの一致を指定します。
ステップ 5	match ipv4 protocol 例： デバイス (config-flow-record)# match ipv4 protocol	IPv4 プロトコルとの一致を指定します。
ステップ 6	match application name 例： デバイス (config-flow-record)# match application name	アプリケーション名との一致を指定します。 (注) この操作は、AVC サポートでは必須です。フローがアプリケーションと一致することが可能になるためです。
ステップ 7	match connection client ipv4 address 例： デバイス (config-flow-record)# match connection client ipv4 address	クライアント (フローイニシエータ) の IPv4 アドレスとの一致を指定します。
ステップ 8	match connection client transport port 例：	(任意) フローレコードのキーフィールドとして、クライアントの接続ポートとの一致を指定します。

	コマンドまたはアクション	目的
	デバイス (config-flow-record)# match connection client transport port	
ステップ 9	match connection server ipv4 address 例 : デバイス (config-flow-record)# match connection server ipv4 address	サーバー (フローレスポンド) の IPv4 アドレスとの一致を指定します。
ステップ 10	match connection server transport port 例 : デバイス (config-flow-record)# match connection server transport port	サーバーのトランスポート ポートとの一致を指定します。
ステップ 11	match flow observation point 例 : デバイス (config-flow-record)# match flow observation point	フロー観測メトリックの観測ポイント ID との一致を指定します。
ステップ 12	collect flow direction 例 : デバイス (config-flow-record)# collect flow direction	次の手順で collect connection initiator コマンドの initiator キーワードで指定される双方向フローの関連する側 (イニシエータまたはレスポンド) の方向 (入力または出力) を収集するように指定します。 initiator キーワードで指定される値に応じて、 flow direction キーワードは次の値をとります。 <ul style="list-style-type: none">• 0x01 = 入力フロー• 0x02 = 出力フロー initiator キーワードがイニシエータに設定されている場合、フローの方向はフローのイニシエータ側から指定されます。 initiator キーワードがレスポンドに設定されている場合、フローの方向はフローのレスポンド側から指定されます。有線 AVC では、 initiator キーワードは常にイニシエータに設定されています。
ステップ 13	collect connection initiator 例 : デバイス (config-flow-record)# collect connection initiator	collect flow direction コマンドで指定されたフローの方向に関連するフローの側 (イニシエータまたはレスポンド) を収集するように指定します。 initiator キーワードは、フローの方向に関する次の情報を提供します。 <ul style="list-style-type: none">• 0x01 = イニシエータ : フローの送信元は接続のイニシエータです

	コマンドまたはアクション	目的
		有線 AVC では、 initiator キーワードは常にイニシエータに設定されています。
ステップ 14	collect connection new-connections 例： デバイス (config-flow-record)# collect connection new-connections	観測された接続開始の数を収集するように指定します。
ステップ 15	collect connection client counter packets long 例： デバイス (config-flow-record)# collect connection client counter packets long	クライアントが送信したパケット数を収集するように指定します。
ステップ 16	collect connection client counter bytes network long 例： デバイス (config-flow-record)# collect connection client counter bytes network long	クライアントが送信したバイト数の合計を収集するように指定します。
ステップ 17	collect connection server counter packets long 例： デバイス (config-flow-record)# collect connection server counter packets long	サーバーが送信したパケット数を収集するように指定します。
ステップ 18	collect connection server counter bytes network long 例： デバイス (config-flow-record)# collect connection server counter bytes network long	サーバーが送信したバイト数の合計を収集するように指定します。
ステップ 19	collect timestamp absolute first 例： デバイス (config-flow-record)# collect timestamp absolute first	最初のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。
ステップ 20	collect timestamp absolute last 例： デバイス (config-flow-record)# collect timestamp absolute last	最新のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。
ステップ 21	end 例： Device (config)# end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。
ステップ 22	show flow record 例：	すべてのフロー レコードに関する情報を表示します。

	コマンドまたはアクション	目的
	デバイス # <code>show flow record</code>	

方向性フローレコード

フローレコード 3 : 方向性フローレコード : 入力

手順の概要

1. **configure terminal**
2. **flow record** *flow_record_name*
3. **description** *description*
4. **match ipv4 version**
5. **match ipv4 protocol**
6. **match ipv4 source address**
7. **match ipv4 destination address**
8. **match transport source-port**
9. **match transport destination-port**
10. **match interface input**
11. **match application name**
12. **collect interface output**
13. **collect counter bytes long**
14. **collect counter packets long**
15. **collect timestamp absolute first**
16. **collect timestamp absolute last**
17. **end**
18. **show flow record**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : Device# <code>configure terminal</code>	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	flow record <i>flow_record_name</i> 例 : デバイス (config) # <code>flow record fr-wdavic-3</code>	フローレコード 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	description <i>description</i> 例 : デバイス (config-flow-record) # <code>description flow-record-1</code>	(任意) フローレコードの説明を作成します。

	コマンドまたはアクション	目的
ステップ 4	match ipv4 version 例 : デバイス (config-flow-record)# match ipv4 version	IPv4 ヘッダーからの IP バージョンとの一致を指定します。
ステップ 5	match ipv4 protocol 例 : デバイス (config-flow-record)# match ipv4 protocol	IPv4 プロトコルとの一致を指定します。
ステップ 6	match ipv4 source address 例 : デバイス (config-flow-record)# match ipv4 source address	IPv4 送信元アドレスとの一致をキーフィールドとして指定します。
ステップ 7	match ipv4 destination address 例 : デバイス (config-flow-record)# match ipv4 destination address	IPv4 宛先アドレスとの一致をキーフィールドとして指定します。
ステップ 8	match transport source-port 例 : デバイス (config-flow-record)# match transport source-port	トランスポート発信元ポートとの一致をキーフィールドとして指定します。
ステップ 9	match transport destination-port 例 : デバイス (config-flow-record)# match transport destination-port	トランスポート宛先ポートとの一致をキーフィールドとして指定します。
ステップ 10	match interface input 例 : デバイス (config-flow-record)# match interface input	入力インターフェイスとの一致をキーフィールドとして指定します。
ステップ 11	match application name 例 : デバイス (config-flow-record)# match application name	アプリケーション名との一致を指定します。 (注) この操作は、AVC サポートでは必須です。フローがアプリケーションと一致することが可能になるためです。
ステップ 12	collect interface output 例 :	フローから出力インターフェイスを収集するように指定します。

フローレコード 4 : 方向性フローレコード : 出力

	コマンドまたはアクション	目的
	デバイス (config-flow-record)# collect interface output	
ステップ 13	collect counter bytes long 例 : デバイス (config-flow-record)# collect counter bytes long	フローのバイト数を収集するように指定します。
ステップ 14	collect counter packets long 例 : デバイス (config-flow-record)# collect counter packets long	フローのパケット数を収集するように指定します。
ステップ 15	collect timestamp absolute first 例 : デバイス (config-flow-record)# collect timestamp absolute first	最初のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。
ステップ 16	collect timestamp absolute last 例 : デバイス (config-flow-record)# collect timestamp absolute last	最新のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。
ステップ 17	end 例 : Device(config)# end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。
ステップ 18	show flow record 例 : デバイス # show flow record	すべてのフロー レコードに関する情報を表示します。

フローレコード 4 : 方向性フローレコード : 出力

手順の概要

1. **configure terminal**
2. **flow record** *flow_record_name*
3. **description** *description*
4. **match ipv4 version**
5. **match ipv4 protocol**
6. **match ipv4 source address**
7. **match ipv4 destination address**
8. **match transport source-port**
9. **match transport destination-port**

10. **match interface output**
11. **match application name**
12. **collect interface input**
13. **collect counter bytes long**
14. **collect counter packets long**
15. **collect timestamp absolute first**
16. **collect timestamp absolute last**
17. **end**
18. **show flow record**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : Device# 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	flow record flow_record_name 例 : デバイス (config)# flow record fr-wdavic-4	フローレコード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	description description 例 : デバイス (config-flow-record)# description flow-record-1	(任意) フローレコードの説明を作成します。
ステップ 4	match ipv4 version 例 : デバイス (config-flow-record)# match ipv4 version	IPv4 ヘッダーからの IP バージョンとの一致を指定します。
ステップ 5	match ipv4 protocol 例 : デバイス (config-flow-record)# match ipv4 protocol	IPv4 プロトコルとの一致を指定します。
ステップ 6	match ipv4 source address 例 : デバイス (config-flow-record)# match ipv4 source address	IPv4 送信元アドレスとの一致をキーフィールドとして指定します。
ステップ 7	match ipv4 destination address 例 : デバイス (config-flow-record)# match ipv4 destination address	IPv4 宛先アドレスとの一致をキーフィールドとして指定します。

	コマンドまたはアクション	目的
ステップ 8	match transport source-port 例 : デバイス (config-flow-record)# match transport source-port	トランスポート発信元ポートとの一致をキーフィールドとして指定します。
ステップ 9	match transport destination-port 例 : デバイス (config-flow-record)# match transport destination-port	トランスポート宛先ポートとの一致をキーフィールドとして指定します。
ステップ 10	match interface output 例 : デバイス (config-flow-record)# match interface output	出カインターフェイスとの一致をキーフィールドとして指定します。
ステップ 11	match application name 例 : デバイス (config-flow-record)# match application name	アプリケーション名との一致を指定します。 (注) この操作は、AVC サポートでは必須です。フローがアプリケーションと一致することが可能になるためです。
ステップ 12	collect interface input 例 : デバイス (config-flow-record)# collect interface input	フローから入カインターフェイスを収集するように指定します。
ステップ 13	collect counter bytes long 例 : デバイス (config-flow-record)# collect counter bytes long	フローのバイト数を収集するように指定します。
ステップ 14	collect counter packets long 例 : デバイス (config-flow-record)# collect counter packets long	フローのパケット数を収集するように指定します。
ステップ 15	collect timestamp absolute first 例 : デバイス (config-flow-record)# collect timestamp absolute first	最初のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。
ステップ 16	collect timestamp absolute last 例 :	最新のパケットがフローで確認されたときの時間をミリ秒単位で収集するように指定します。

	コマンドまたはアクション	目的
	デバイス (config-flow-record)# collect timestamp absolute last	
ステップ 17	end 例： Device (config)# end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。
ステップ 18	show flow record 例： デバイス # show flow record	すべてのフロー レコードに関する情報を表示します。

フロー エクスポートの作成

フロー エクスポートを作成すると、フローのエクスポート パラメータを定義できます。

手順の概要

1. **configure terminal**
2. **flow exporter** *flow_exporter_name*
3. **description** *description*
4. **destination** { *hostname* | *ipv4-address* | *ipv6-address* }
5. **option application-table** [*timeout seconds*]
6. **end**
7. **show flow exporter**
8. **show flow exporter statistics**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# 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	flow exporter <i>flow_exporter_name</i> 例： デバイス (config)# flow exporter flow-exporter-1	フロー エクスポート 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	description <i>description</i> 例： デバイス (config-flow-exporter)# description flow-exporter-1	(任意) フローエクスポートの説明を作成します。

フロー モニターの作成

	コマンドまたはアクション	目的
ステップ 4	destination { <i>hostname</i> <i>ipv4-address</i> <i>ipv6-address</i> } 例： デバイス (config-flow-exporter)# destination 10.10.1.1	エクスポートでデータを送信する宛先システムのホスト名、IPv4 または IPv6 アドレスを指定します。
ステップ 5	option application-table [timeout seconds] 例： デバイス (config-flow-exporter)# option application-table timeout 500	(任意) フロー エクスポートのアプリケーション テーブルのオプションを設定します。 timeout オプションを使用すると、フローエクスポートの再送信時間を秒単位で設定できます。有効な範囲は 1 ~ 86400 秒です。
ステップ 6	end 例： Device(config)# end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバルコンフィギュレーションモードを終了できます。
ステップ 7	show flow exporter 例： デバイス # show flow exporter	すべてのフローエクスポートに関する情報を表示します。
ステップ 8	show flow exporter statistics 例： デバイス # show flow exporter statistics	フロー エクスポートの統計情報を表示します。

フロー モニターの作成

フロー モニターを作成して、フロー レコードに関連付けることができます。

手順の概要

1. **configure terminal**
2. **flow monitor** *monitor-name*
3. **description** *description*
4. **record** *record-name*
5. **exporter** *exporter-name*
6. **cache** { **entries** *number-of-entries* | **timeout** { **active** | **inactive** } | **type normal** }
7. **end**
8. **show flow monitor**
9. **show flow monitor** *flow-monitor-name*
10. **show flow monitor** *flow-monitor-name* **statistics**
11. **clear flow monitor** *flow-monitor-name* **statistics**
12. **show flow monitor** *flow-monitor-name* **cache format table**
13. **show flow monitor** *flow-monitor-name* **cache format record**
14. **show flow monitor** *flow-monitor-name* **cache format csv**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# 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	flow monitor monitor-name 例： デバイス (config)# flow monitor flow-monitor-1	フロー モニターを作成し、フロー モニター 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	description description 例： デバイス (config-flow-monitor)# description flow-monitor-1	(任意) フロー モニターの説明を作成します。
ステップ 4	record record-name 例： デバイス (config-flow-monitor)# record flow-record-1	事前に作成されたレコードの名前を指定します。
ステップ 5	exporter exporter-name 例： デバイス (config-flow-monitor)# exporter flow-exporter-1	事前に作成されたエクスポートの名前を指定します。
ステップ 6	cache { entries number-of-entries timeout { active inactive } type normal } 例： デバイス (config-flow-monitor)# cache timeout active 1800 例： デバイス (config-flow-monitor)# cache timeout inactive 200 例： デバイス (config-flow-monitor)# cache type normal	(任意) フロー キャッシュ パラメータを設定するように指定します。 • entries number-of-entries : フローキャッシュ内のフローエントリの最大数を 16 ~ 65536 の範囲で指定します。 (注) 標準のキャッシュタイプのみがサポートされます。
ステップ 7	end 例： Device(config)# end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。
ステップ 8	show flow monitor 例：	すべてのフロー モニターに関する情報を表示します。

インターフェイスへのフロー モニターの関連付け

	コマンドまたはアクション	目的
	デバイス # <code>show flow monitor</code>	
ステップ 9	<code>show flow monitor flow-monitor-name</code> 例： デバイス # <code>show flow monitor flow-monitor-1</code>	指定した有線 AVC フロー モニターに関する情報を表示します。
ステップ 10	<code>show flow monitor flow-monitor-name statistics</code> 例： デバイス# <code>show flow monitor flow-monitor-1 statistics</code>	有線 AVC フロー モニターの統計情報を表示します。
ステップ 11	<code>clear flow monitor flow-monitor-name statistics</code> 例： デバイス# <code>clear flow monitor flow-monitor-1 statistics</code>	指定したフロー モニターの統計情報をクリアします。 clear flow monitor flow-monitor-1 statistics を使用した後に show flow monitor flow-monitor-1 statistics コマンドを使用して、すべての統計情報がリセットされたことを確認します。
ステップ 12	<code>show flow monitor flow-monitor-name cache format table</code> 例： デバイス# <code>show flow monitor flow-monitor-1 cache format table</code>	表形式でフロー キャッシュの内容を表示します。
ステップ 13	<code>show flow monitor flow-monitor-name cache format record</code> 例： デバイス# <code>show flow monitor flow-monitor-1 cache format record</code>	フロー レコードと同様の形式でフロー キャッシュの内容を表示します。
ステップ 14	<code>show flow monitor flow-monitor-name cache format csv</code> 例： デバイス# <code>show flow monitor flow-monitor-1 cache format csv</code>	CSV 形式でフロー キャッシュの内容を表示します。

インターフェイスへのフロー モニターの関連付け

異なる事前定義済みレコードを持つ 2 つの異なる有線 AVC モニターをインターフェイスに同時に接続できます。

手順の概要

1. **configure terminal**
2. **interface***interface-id*
3. **ip flow monitor** *monitor-name* { **input** | **output** }

4. end

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： Device# 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	interface interface-id 例： デバイス (config) # interface GigabitEthernet 1/0/1	インターフェイス 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	ip flow monitor monitor-name { input output } 例： デバイス (config-if) # ip flow monitor flow-monitor-1 input	入力パケットと出力パケットの両方またはいずれか用のインターフェイスにフローモニターを関連付けます。
ステップ 4	end 例： Device (config) # end	特権 EXEC モードに戻ります。また、Ctrl+Z キーを押しても、グローバル コンフィギュレーション モードを終了できます。

NBAR2 カスタム アプリケーション

NBAR2 では、カスタム プロトコルを使用してカスタム アプリケーションを識別できます。カスタム プロトコルは、プロトコルとアプリケーションをサポートしますが、現在のところ、NBAR2 はサポートしていません。

すべての展開において、シスコが提供する NBAR2 プロトコルパックの対象外であるローカル アプリケーションおよび特定のアプリケーションがあります。ローカル アプリケーションは主に次のように分類されます。

- 組織への特定のアプリケーション
- 地域特有のアプリケーション

NBAR2 では、このようなローカル アプリケーションを手動でカスタマイズする方法を提供しています。グローバル コンフィギュレーション モードで **ip nbar custom myappname** コマンドを使用して、手動でアプリケーションをカスタマイズできます。カスタム アプリケーションは、組み込みプロトコルより優先されます。それぞれのカスタム プロトコルでは、ユーザーは、レポート目的に使用できるセレクト ID を定義できます。

さまざまなタイプのアプリケーション カスタマイズがあります。

一般的なプロトコルのカスタマイズ

- HTTP
- SSL
- DNS

コンポジット：複数の基本的なプロトコルに基づくカスタマイズ：**server-name**

レイヤ 3/レイヤ 4 のカスタマイズ

- IPv4 アドレス
- DSCP 値
- TCP/UDP ポート
- フロー送信元または宛先の方向

バイトオフセット：ペイロードの特定のバイト値に基づくカスタマイズ

HTTP のカスタマイズ

HTTP のカスタマイズは、次の HTTP フィールドの組み合わせに基づいて実行できます。

- **cookie** : HTTP クッキー
- **host** : リソースを含む元のサーバーのホスト名
- **method** : HTTP メソッド
- **referrer** : リソース リクエストの取得元のアドレス
- **url** : Uniform Resource Locator のパス
- **user-agent** : 要求を送信するエージェントによって使用されているソフトウェア
- **version** : HTTP バージョン
- **via** : HTTP 経由フィールド

HTTP のカスタマイズ

セレクタ ID 10 が付いた HTTP ホスト「*mydomain.com」を使用する MYHTTP と呼ばれるカスタム アプリケーション。

```
Device# configure terminal
Device(config)# ip nbar custom MYHTTP http host *mydomain.com id 10
```

SSL のカスタマイズ

SSL サーバー名指定 (SNI) または共通名 (CN) から抽出した情報を使用して、SSL 暗号化トラフィックでカスタマイズを行うことができます。

SSL のカスタマイズ

セレクト ID 11 が付いた SSL 固有名「mydomain.com」を使用する MYSSL と呼ばれるカスタム アプリケーション。

```
Device# configure terminal
Device(config)#ip nbar custom MYSSL ssl unique-name *mydomain.com id 11
```

DNS のカスタマイズ

NBAR2 は、DNS 要求および応答トラフィックを確認し、アプリケーションへの DNS 応答に関連付けることができます。DNS 応答から戻された IP アドレスはキャッシュされ、その特定のアプリケーションに関連付けられているその後のパケットフローに使用されます。

ip nbar custom application-name dns domain-name id application-id コマンドは、DNS のカスタマイズに使用されます。既存のアプリケーションを拡張するには、**ip nbar custom application-name dns domain-name domain-name extends existing-application** コマンドを使用します。

DNS ベースのカスタマイズの詳細については、http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/qos_nbar/configuration/xs/asr1000/qos-nbar-xe-3s-asr-1000-book/nbar-custapp-dns-xe.html を参照してください。

DNS のカスタマイズ

セレクト ID 12 が付いた DNS ドメイン名「mydomain.com」を使用する MYDNS と呼ばれるカスタム アプリケーション。

```
Device# configure terminal
Device(config)# ip nbar custom MYDNS dns domain-name *mydomain.com id 12
```

複合カスタマイズ

NBAR2 では、HTTP、SSL または DNS に現れるドメイン名に基づいてアプリケーションをカスタマイズする方法が提供されます。

複合カスタマイズ

セレクト ID 13 が付いた HTTP、SSL または DNS ドメイン名「mydomain.com」を使用する MYDOMAIN と呼ばれるカスタム アプリケーション。

```
Device# configure terminal
Device(config)# ip nbar custom MYDOMAIN composite server-name *mydomain.com id 13
```

L3/L4 のカスタマイズ

レイヤ3/レイヤ4のカスタマイズは、パケットタプルに基づいており、フローの最初のパケットで常に一致します。

L3/L4 のカスタマイズ

IP アドレス 10.56.1.10 および 10.56.1.11、セクタ ID 14 が付いた TCP および DSCP ef に一致する LAYER4CUSTOM と呼ばれるカスタムアプリケーション。

```
Device# configure terminal
Device(config)# ip nbar custom LAYER4CUSTOM transport tcp id 14
Device(config-custom)# ip address 10.56.1.10 10.56.1.11
Device(config-custom)# dscp ef
```

例：カスタムアプリケーションのモニターリング

カスタムアプリケーションのモニターリングのための show コマンド

show ip nbar protocol-id | inc Custom

```
デバイス# show ip nbar protocol-id | inc Custom
LAYER4CUSTOM          14          Custom
MYDNS                 12          Custom
MYDOMAIN              13          Custom
MYHTTP                10          Custom
MYSSL                 11          Custom
```

show ip nbar protocol-discovery protocol CUSTOM_APP

```
WSW-157# show ip nbar protocol-id MYSSL
Protocol Name          id          type
-----
MYSSL                  11          Custom
```

NBAR2 ダイナミック ヒットレス プロトコルパックのアップグレード

プロトコルパックは、デバイスのシスコソフトウェアを置き換えることなく、デバイスの NBAR2 プロトコルサポートを更新するソフトウェアパッケージです。プロトコルパックには、NBAR2 によって正式にサポートされている、コンパイル済みでパック済みのアプリケーションに関する情報が含まれています。各アプリケーションについて、プロトコルパックには、アプリケーション署名とアプリケーション属性の情報が含まれています。各ソフトウェアリリースには、組み込みのプロトコルパックがバンドルされています。

プロトコルパックには次の特長があります。

- ロードが容易で高速。
- 高いバージョンのプロトコルパックにアップグレードしたり、低いバージョンのプロトコルパックに戻したりするのが容易。
- スイッチのリロードを必要としない。

NBAR2 プロトコルパックは、次の URL から Cisco Software Center でダウンロードできます：
<https://software.cisco.com/download/navigator.html>

NBAR2 プロトコルパックの前提条件

新しいプロトコルパックをロードする前に、すべてのスイッチ メンバー上でプロトコルパックをフラッシュにコピーする必要があります。

プロトコルパックをロードするには、[NBAR2 プロトコルパックのロード \(171 ページ\)](#) を参照してください。

NBAR2 プロトコルパックのロード

手順の概要

1. **enable**
2. **configure terminal**
3. **ip nbar protocol-pack** *protocol-pack* [**force**]
4. **exit**
5. **show ip nbar protocol-pack** {*protocol-pack* | **active**} [**detail**]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# 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	ip nbar protocol-pack <i>protocol-pack</i> [force] 例： Device(config)# ip nbar protocol-pack flash:defProtoPack 例： Device(config)# default ip nbar protocol-pack	プロトコルパックをロードします。 <ul style="list-style-type: none"> • 基本のプロトコルパックバージョンとは異なる、より低いバージョンのプロトコルパックを指定し、ロードするには、force キーワードを使用します。これにより、スイッチの現在のプロトコルパックでサポートされていない設定も削除されます。 組み込みのプロトコルパックに戻るには、次のコマンドを使用します。
ステップ 4	exit 例： Device(config)# exit	特権 EXEC モードに戻ります。

例：NBAR2 プロトコルパックのロード

	コマンドまたはアクション	目的
ステップ 5	<p>show ip nbar protocol-pack {protocol-pack active} [detail]</p> <p>例：</p> <pre>Device# show ip nbar protocol-pack active</pre>	<p>プロトコルパック情報を表示します。</p> <ul style="list-style-type: none"> このコマンドを使用して、ロードされたプロトコルパックのバージョン、パブリッシャ、その他の詳細を確認します。 指定されたプロトコルパックの情報を表示するには、<i>protocol-pack</i> 引数を使用します。 アクティブなプロトコルパックの情報を表示するには、active キーワードを使用します。 詳細なプロトコルパックの情報を表示するには、detail キーワードを使用します。

例：NBAR2 プロトコルパックのロード

次の例に、新しいプロトコルパックをロードする方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip nbar protocol-pack flash:newDefProtoPack
Device(config)# exit
```

次の例に、**force** キーワードを使用して下位バージョンのプロトコルパックをロードする方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# ip nbar protocol-pack flash:OldDefProtoPack force
Device(config)# exit
```

次の例に、組み込みのプロトコルパックに戻す方法を示します。

```
Device> enable
Device# configure terminal
Device(config)# default ip nbar protocol-pack
Device(config)# exit
```

Application Visibility and Control のモニターリング

このセクションでは、アプリケーションの可視性に関する新しいコマンドについて説明します。

次のコマンドは、およびアクセスポートのアプリケーションの可視性をモニターするために使用できます。

表 8: のアプリケーションの可視性モニターリングコマンド

コマンド	目的
------	----

<pre>show ip nbar protocol-discovery [interface interface-type interface-number] [stats{byte-count bit-rate packet-count max-bit-rate}] [protocol protocol-name top-n number]</pre>	<p>NBAR Protocol Discovery 機能によって収集された統計情報を表示します。</p> <ul style="list-style-type: none"> • (任意) 表示される統計情報を最適化するには、キーワードおよび引数を入力します。キーワードのそれぞれの詳細については、『Cisco IOS Quality of Service Solutions Command Reference』の show ip nbar protocol-discovery コマンドを参照してください。
<pre>show policy-map interface interface-type interface-number</pre>	<p>インターフェイスに適用したポリシーマップについての情報を表示します。</p>
<pre>show platform software fed switch スイッチ ID wdavc flows</pre>	<p>指定したスイッチのすべてのフローに関する統計情報を表示します。</p>

例 : Application Visibility and Control の設定

次に、`match protocol` でアプリケーション名のフィルタを適用してクラス マップを作成する例を示します。

```
デバイス# configure terminal
デバイス(config)# class-map match-any NBAR-VOICE
デバイス(config-cmap)# match protocol ms-lync-audio
デバイス(config-cmap)#end
```

次に、ポリシー マップを作成し、出力 QoS の既存のクラス マップを定義する例を示します。

```
デバイス# configure terminal
デバイス(config)# policy-map test-avc-up
デバイス(config-pmap)# class cat-browsing
デバイス(config-pmap-c)# police 150000
デバイス(config-pmap-c)# set dscp 12
デバイス(config-pmap-c)#end
```

次に、ポリシー マップを作成し、入力 QoS の既存のクラス マップを定義する例を示します。

```
デバイス# configure terminal
デバイス(config)# policy-map test-avc-down
デバイス(config-pmap)# class cat-browsing
デバイス(config-pmap-c)# police 200000
デバイス(config-pmap-c)# set dscp 10
デバイス(config-pmap-c)#end
```

次に、ポリシー マップをスイッチ ポートに適用する例を示します。

```
デバイス# configure terminal
デバイス(config)# interface GigabitEthernet 1/0/1
デバイス(config-if)# switchport mode access
デバイス(config-if)# switchport access vlan 20
```

```

デバイス(config-if)# service-policy input POLICING_IN
デバイス(config-if)#end

```

show コマンドによる設定の表示

show ip nbar protocol-discovery

インターフェイスごとのプロトコル検出統計情報のレポートを表示します。

次に、インターフェイスごとの統計情報の出力例を示します。

```

デバイスqos-cat9k-reg2-r1# show ip nbar protocol-discovery int GigabitEthernet1/0/1

```

```

GigabitEthernet1/0/1
Last clearing of "show ip nbar protocol-discovery" counters 00:03:16

```

	Input
Output	-----

Protocol	Packet Count
Packet Count	Byte Count
Byte Count	30sec Bit Rate (bps)
30sec Bit Rate (bps)	30sec Max Bit Rate (bps)
30sec Max Bit Rate (bps)	-----
-----	-----
ms-lync	60580
55911	31174777
28774864	3613000
93000	3613000
3437000	60580
Total	31174777
55911	3613000
28774864	3613000
93000	3613000
3437000	

show policy-map interface

すべてのインターフェイス上の QoS 統計情報および設定済みのポリシーマップを表示します。

次に、すべてのインターフェイスに設定されたポリシーマップの出力例を示します。

```

デバイスqos-cat9k-reg2-r1# show policy-map int

GigabitEthernet1/0/1
Service-policy input: MARKING-IN

Class-map: NBAR-VOICE (match-any)
  718 packets
Match: protocol ms-lync-audio
  0 packets, 0 bytes
  30 second rate 0 bps
QoS Set
  dscp ef

Class-map: NBAR-MM_CONFERENCING (match-any)
  6451 packets
Match: protocol ms-lync
  0 packets, 0 bytes
  30 second rate 0 bps
Match: protocol ms-lync-video
  0 packets, 0 bytes
  30 second rate 0 bps
QoS Set
  dscp af41

Class-map: class-default (match-any)
  34 packets
Match: any

```

show コマンドによるフロー モニター設定の表示

show flow monitor wdavc

指定した有線 AVC フロー モニターに関する情報を表示します。

```

デバイス # show flow monitor wdavc

Flow Monitor wdavc:
Description:      User defined
Flow Record:     wdavc
Flow Exporter:   wdavc-exp (inactive)
Cache:
Type:            normal (Platform cache)
Status:         not allocated
Size:           12000 entries
Inactive Timeout: 15 secs
Active Timeout: 1800 secs

```

show flow monitor wdavc statistics

有線 AVC フロー モニターの統計情報を表示します。

```

デバイス# show flow monitor wdavc statistics
Cache type: Normal (Platform cache)
Cache size: 12000

```

```

Current entries:                               13

Flows added:                                  26
Flows aged:                                   13
  - Active timeout      ( 1800 secs)         1
  - Inactive timeout    (   15 secs)         12

```

clear flow monitor wdvac statistics

指定したフロー モニターの統計情報をクリアします。**clear flow monitor wdvac statistics** を使用した後に **show flow monitor wdvac statistics** コマンドを使用して、すべての統計情報がリセットされたことを確認します。以下に、フローモニター統計情報をクリアした後の **show flow monitor wdvac statistics** コマンドのサンプル出力を示します。

```

デバイス# show flow monitor wdvac statistics
Cache type:                               Normal (Platform cache)
Cache size:                               12000
Current entries:                           0

Flows added:                               0
Flows aged:                                0

```

show コマンドによるキャッシュの内容の表示

show flow monitor wdvac cache format table

表形式でフロー キャッシュの内容を表示します。

```

デバイス# show flow monitor wdvac cache format table
Cache type:                               Normal (Platform cache)
Cache size:                               12000
Current entries:                           13

Flows added:                               26
Flows aged:                                13
  - Active timeout      ( 1800 secs)         1
  - Inactive timeout    (   15 secs)         12

CONN IPV4 INITIATOR ADDR  CONN IPV4 RESPONDER ADDR  CONN RESPONDER PORT
FLOW OBSPOINT ID  IP VERSION  IP PROT  APP NAME
flow dirn .....
-----
-----
64.103.125.147          144.254.71.184
53          4294967305          4          17  port dns
  Input .....
64.103.121.103          10.1.1.2
67          4294967305          4          17  layer7 dhcp
  Input ....contd.....
64.103.125.3            64.103.125.97
68          4294967305          4          17  layer7 dhcp
  Input .....
10.0.2.6                157.55.40.149

```

443


```

          4294967305          4          6 layer7 ms-lync
Input .....
64.103.126.28          66.163.36.139          443
          4294967305          4          6 layer7 cisco-jabber-im
Input ....contd.....
64.103.125.2          64.103.125.29
68          4294967305          4          17 layer7 dhcp
  Input .....
64.103.125.97          64.103.101.181
67          4294967305          4          17 layer7 dhcp
  Input .....
192.168.100.6          10.10.20.1          5060
          4294967305          4          17 layer7 cisco-jabber-control
Input ....contd.....
64.103.125.3          64.103.125.29
68          4294967305          4          17 layer7 dhcp
  Input .....
10.80.101.18          10.80.101.6          5060
          4294967305          4          6 layer7 cisco-collab-control
Input .....
10.1.11.4          66.102.11.99
80          4294967305          4          6 layer7 google-services
  Input ....contd.....
64.103.125.2          64.103.125.97
68          4294967305          4          17 layer7 dhcp
  Input .....
64.103.125.29          64.103.101.181
67          4294967305          4          17 layer7 dhcp
  Input .....

```

show flow monitor wdvac cache format record

フロー レコードと同様の形式でフロー キャッシュの内容を表示します。

デバイス# **show flow monitor wdvac cache format record**

```

Cache type:                               Normal (Platform cache)
Cache size:                               12000
Current entries:                          13

Flows added:                              26
Flows aged:                               13
  - Active timeout      ( 1800 secs)      1
  - Inactive timeout    (   15 secs)      12

CONNECTION IPV4 INITIATOR ADDRESS:        64.103.125.147
CONNECTION IPV4 RESPONDER ADDRESS:        144.254.71.184
CONNECTION RESPONDER PORT:                53
FLOW OBSPOINT ID:                        4294967305
IP VERSION:                               4
IP PROTOCOL:                              17
APPLICATION NAME:                         port dns
flow direction:                           Input
timestamp abs first:                      08:55:46.917

```

```
timestamp abs last:                08:55:46.917
connection initiator:              Initiator
connection count new:              2
connection server packets counter: 1
connection client packets counter: 1
connection server network bytes counter: 190
connection client network bytes counter: 106

CONNECTION IPV4 INITIATOR ADDRESS: 64.103.121.103
CONNECTION IPV4 RESPONDER ADDRESS: 10.1.1.2
CONNECTION RESPONDER PORT:        67
FLOW OBSPOINT ID:                 4294967305
IP VERSION:                        4
IP PROTOCOL:                       17
APPLICATION NAME:                   layer7 dhcp
flow direction:                     Input
timestamp abs first:                08:55:47.917
timestamp abs last:                08:55:47.917
connection initiator:              Initiator
connection count new:              1
connection server packets counter: 0
connection client packets counter: 1
connection server network bytes counter: 0
connection client network bytes counter: 350

CONNECTION IPV4 INITIATOR ADDRESS: 64.103.125.3
CONNECTION IPV4 RESPONDER ADDRESS: 64.103.125.97
CONNECTION RESPONDER PORT:        68
FLOW OBSPOINT ID:                 4294967305
IP VERSION:                        4
IP PROTOCOL:                       17
APPLICATION NAME:                   layer7 dhcp
flow direction:                     Input
timestamp abs first:                08:55:47.917
timestamp abs last:                08:55:53.917
connection initiator:              Initiator
connection count new:              1
connection server packets counter: 0
connection client packets counter: 4
connection server network bytes counter: 0
connection client network bytes counter: 1412

CONNECTION IPV4 INITIATOR ADDRESS: 10.0.2.6
CONNECTION IPV4 RESPONDER ADDRESS: 157.55.40.149
CONNECTION RESPONDER PORT:        443
FLOW OBSPOINT ID:                 4294967305
IP VERSION:                        4
IP PROTOCOL:                       6
APPLICATION NAME:                   layer7 ms-lync
flow direction:                     Input
timestamp abs first:                08:55:46.917
```

```
timestamp abs last:                08:55:46.917
connection initiator:              Initiator
connection count new:              2
connection server packets counter: 10
connection client packets counter: 14
connection server network bytes counter: 6490
connection client network bytes counter: 1639

CONNECTION IPV4 INITIATOR ADDRESS: 64.103.126.28
CONNECTION IPV4 RESPONDER ADDRESS: 66.163.36.139
CONNECTION RESPONDER PORT:        443
FLOW OBSPOINT ID:                 4294967305
IP VERSION:                        4
IP PROTOCOL:                       6
APPLICATION NAME:                   layer7 cisco-jabber-im
flow direction:                     Input
timestamp abs first:                08:55:46.917
timestamp abs last:                08:55:46.917
connection initiator:              Initiator
connection count new:              2
connection server packets counter: 12
connection client packets counter: 10
connection server network bytes counter: 5871
connection client network bytes counter: 2088

CONNECTION IPV4 INITIATOR ADDRESS: 64.103.125.2
CONNECTION IPV4 RESPONDER ADDRESS: 64.103.125.29
CONNECTION RESPONDER PORT:        68
FLOW OBSPOINT ID:                 4294967305
IP VERSION:                        4
IP PROTOCOL:                       17
APPLICATION NAME:                   layer7 dhcp
flow direction:                     Input
timestamp abs first:                08:55:47.917
timestamp abs last:                08:55:47.917
connection initiator:              Initiator
connection count new:              1
connection server packets counter: 0
connection client packets counter: 2
connection server network bytes counter: 0
connection client network bytes counter: 712

CONNECTION IPV4 INITIATOR ADDRESS: 64.103.125.97
CONNECTION IPV4 RESPONDER ADDRESS: 64.103.101.181
CONNECTION RESPONDER PORT:        67
FLOW OBSPOINT ID:                 4294967305
IP VERSION:                        4
IP PROTOCOL:                       17
APPLICATION NAME:                   layer7 dhcp
flow direction:                     Input
timestamp abs first:                08:55:47.917
```

```
timestamp abs last:                08:55:47.917
connection initiator:              Initiator
connection count new:              1
connection server packets counter: 0
connection client packets counter: 1
connection server network bytes counter: 0
connection client network bytes counter: 350

CONNECTION IPV4 INITIATOR ADDRESS: 192.168.100.6
CONNECTION IPV4 RESPONDER ADDRESS: 10.10.20.1
CONNECTION RESPONDER PORT:         5060
FLOW OBSPOINT ID:                  4294967305
IP VERSION:                         4
IP PROTOCOL:                       17
APPLICATION NAME:                   layer7 cisco-jabber-control
flow direction:                    Input
timestamp abs first:               08:55:46.917
timestamp abs last:               08:55:46.917
connection initiator:              Initiator
connection count new:              1
connection server packets counter: 0
connection client packets counter: 2
connection server network bytes counter: 0
connection client network bytes counter: 2046

CONNECTION IPV4 INITIATOR ADDRESS: 64.103.125.3
CONNECTION IPV4 RESPONDER ADDRESS: 64.103.125.29
CONNECTION RESPONDER PORT:         68
FLOW OBSPOINT ID:                  4294967305
IP VERSION:                         4
IP PROTOCOL:                       17
APPLICATION NAME:                   layer7 dhcp
flow direction:                    Input
timestamp abs first:               08:55:47.917
timestamp abs last:               08:55:47.917
connection initiator:              Initiator
connection count new:              1
connection server packets counter: 0
connection client packets counter: 2
connection server network bytes counter: 0
connection client network bytes counter: 712

CONNECTION IPV4 INITIATOR ADDRESS: 10.80.101.18
CONNECTION IPV4 RESPONDER ADDRESS: 10.80.101.6
CONNECTION RESPONDER PORT:         5060
FLOW OBSPOINT ID:                  4294967305
IP VERSION:                         4
IP PROTOCOL:                       6
APPLICATION NAME:                   layer7 cisco-collab-control
flow direction:                    Input
timestamp abs first:               08:55:46.917
```

```
timestamp abs last:                08:55:47.917
connection initiator:              Initiator
connection count new:              2
connection server packets counter: 23
connection client packets counter: 27
connection server network bytes counter: 12752
connection client network bytes counter: 8773

CONNECTION IPV4 INITIATOR ADDRESS: 10.1.11.4
CONNECTION IPV4 RESPONDER ADDRESS: 66.102.11.99
CONNECTION RESPONDER PORT:        80
FLOW OBSPOINT ID:                 4294967305
IP VERSION:                        4
IP PROTOCOL:                       6
APPLICATION NAME:                  layer7 google-services
flow direction:                    Input
timestamp abs first:               08:55:46.917
timestamp abs last:               08:55:46.917
connection initiator:              Initiator
connection count new:              2
connection server packets counter: 3
connection client packets counter: 5
connection server network bytes counter: 1733
connection client network bytes counter: 663

CONNECTION IPV4 INITIATOR ADDRESS: 64.103.125.2
CONNECTION IPV4 RESPONDER ADDRESS: 64.103.125.97
CONNECTION RESPONDER PORT:        68
FLOW OBSPOINT ID:                 4294967305
IP VERSION:                        4
IP PROTOCOL:                       17
APPLICATION NAME:                  layer7 dhcp
flow direction:                    Input
timestamp abs first:               08:55:47.917
timestamp abs last:               08:55:53.917
connection initiator:              Initiator
connection count new:              1
connection server packets counter: 0
connection client packets counter: 4
connection server network bytes counter: 0
connection client network bytes counter: 1412

CONNECTION IPV4 INITIATOR ADDRESS: 64.103.125.29
CONNECTION IPV4 RESPONDER ADDRESS: 64.103.101.181
CONNECTION RESPONDER PORT:        67
FLOW OBSPOINT ID:                 4294967305
IP VERSION:                        4
IP PROTOCOL:                       17
APPLICATION NAME:                  layer7 dhcp
flow direction:                    Input
timestamp abs first:               08:55:47.917
```

```

timestamp abs last:                08:55:47.917
connection initiator:              Initiator
connection count new:              1
connection server packets counter: 0
connection client packets counter: 1
connection server network bytes counter: 0
connection client network bytes counter: 350

```

show flow monitor wdacv cache format csv

CSV 形式でフロー キャッシュの内容を表示します。

```
デバイス# show flow monitor wdacv cache format csv
```

```

Cache type:                Normal (Platform cache)
Cache size:                 12000
Current entries:           13

Flows added:                26
Flows aged:                 13
- Active timeout           ( 1800 secs) 1
- Inactive timeout         (   15 secs) 12

```

```

CONN IPV4 INITIATOR ADDR,CONN IPV4 RESPONDER ADDR,CONN RESPONDER
PORT,FLOW OBSPOINT ID,IP VERSION,IP
PROT,APP NAME,flow dirn,time abs first,time abs last,conn initiator,conn
count new,conn server packets
cnt,conn client packets cnt,conn server network bytes cnt,conn client
network bytes cnt
64.103.125.147,144.254.71.184,53,4294967305,4,17,port
dns,Input,08:55:46.917,08:55:46.917,Initiator,2,1,1,190,106
64.103.121.103,10.1.1.2,67,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:47.917,Initiator,1,0,1,0,350
64.103.125.3,64.103.125.97,68,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:53.917,Initiator,1,0,4,0,1412
10.0.2.6,157.55.40.149,443,4294967305,4,6,layer7 ms-
lync,Input,08:55:46.917,08:55:46.917,Initiator,2,10,14,6490,1639
64.103.126.28,66.163.36.139,443,4294967305,4,6,layer7 cisco-jabber-
im,Input,08:55:46.917,08:55:46.917,Initiator,2,12,10,5871,2088
64.103.125.2,64.103.125.29,68,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:47.917,Initiator,1,0,2,0,712
64.103.125.97,64.103.101.181,67,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:47.917,Initiator,1,0,1,0,350
192.168.100.6,10.10.20.1,5060,4294967305,4,17,layer7 cisco-jabber-
control,Input,08:55:46.917,08:55:46.917,Initiator,1,0,2,0,2046
64.103.125.3,64.103.125.29,68,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:47.917,Initiator,1,0,2,0,712
10.80.101.18,10.80.101.6,5060,4294967305,4,6,layer7 cisco-collab-
control,Input,08:55:46.917,08:55:47.917,Initiator,2,23,27,12752,8773
10.1.11.4,66.102.11.99,80,4294967305,4,6,layer7 google-
services,Input,08:55:46.917,08:55:46.917,Initiator,2,3,5,1733,663
64.103.125.2,64.103.125.97,68,4294967305,4,17,layer7
dhcp,Input,08:55:47.917,08:55:53.917,Initiator,1,0,4,0,1412

```

```
64.103.125.29, 64.103.101.181, 67, 4294967305, 4, 17, layer7  
dhcp, Input, 08:55:47.917, 08:55:47.917, Initiator, 1, 0, 1, 0, 350
```

基本的なトラブルシューティング：質問と回答

以下に、有線 Application Visibility and Control のトラブルシューティングに関する基本的な質問と回答を示します。

- 質問：** IPv6 トラフィックが分類されていません。

回答： 現在は IPv4 トラフィックのみがサポートされています。
- 質問：** マルチキャスト トラフィックが分類されていません。

回答： 現在はユニキャスト トラフィックのみがサポートされています。
- 質問：** ping を送信したときに、分類されているかを確認できません。

回答： TCP/UDP プロトコルのみがサポートされています。
- 質問：** SVI に NBAR を接続できないのはなぜですか。

回答： NBAR は物理インターフェイスでのみサポートされています。
- 質問：** ほとんどのトラフィックが CAPWAP トラフィックになっているのですが、なぜですか。

回答： ワイヤレス アクセス ポートに接続されていないアクセス ポートで NBAR が有効になっていることを確認してください。AP から着信するすべてのトラフィックは capwap として分類されます。この場合、実際の分類は AP または WLC で行われます。
- 質問：** プロトコル検出で、トラフィックが片側でしか確認できません。さらに、多くの未知のトラフィックがあります。

回答： これは通常、NBAR が非対称トラフィックを確認していることを示します。片側のトラフィックは1つのスイッチメンバーに分類され、もう一方は別のメンバーに分類されます。トラフィックの両側が確認されるアクセスポートにのみNBARを接続することを推奨します。複数のアップリンクがある場合は、この問題のためそれらにNBARを接続することはできません。ポートチャネルの一部であるインターフェイスにNBARを設定した場合にも同様の問題が発生します。
- 質問：** プロトコル検出で、すべてのアプリケーションの集約ビューが表示されます。時間経過に伴うトラフィック分布を確認するにはどうしたらいいですか。

回答： WebUI を使用して、過去 48 時間の経時的なトラフィックを表示できます。
- 質問：** `match protocol protocol-name` コマンドを使用してキューベースのイーグレスポリシーを設定できません。

回答： NBAR2 ベースの分類子が含まれるポリシーでは、**shape** および **set DSCP** のみがサポートされています。一般的な方法としては、入力で DSCP を設定し、DSCP に基づいて出力でシェーピングを実行します。

9. 質問：インターフェイスに接続している NBAR2 はありませんが、NBAR2 がいまだにアクティブになっています。

回答：match protocol protocol-name を含むクラスマップがあると、NBAR はスタックでグローバルにアクティブになりますが、トラフィックは NBAR 分類の対象にはなりません。これは予期された動作であり、リソースを消費しません。

10. 質問：デフォルトの QoS キューの下にトラフィックがあります。どうしてですか。

回答：新しい各フローでは、フローを分類してハードウェアに結果をインストールするためにいくつかの packets が使われます。この間に、分類は「不明」となり、トラフィックはデフォルト キューに入ります。

Application Visibility and Control に関する追加情報

関連資料

関連項目	マニュアルタイトル
QoS	<i>NBAR Configuration Guide, Cisco IOS XE Release 16.x</i>
NBAR2 プロトコル パック ヒットレス アップグレード	<i>NBAR Configuration Guide, Cisco IOS XE Release 16.x</i>

シスコのテクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	http://www.cisco.com/support

有線ネットワークでの Application Visibility and Control の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	有線ネットワークでの Application Visibility and Control	AVC は、アプリケーションへの適応力やアプリケーションへのインテリジェンス性に基づいて、厳密なパケットおよび接続からブランチおよびキャンパスソリューションを発展させるためのシスコの取り組みの重要な部分です。
Cisco IOS XE Fuji 16.8.1a	有線アプリケーションの表示およびコントロール (有線 AVC) 属性ベース QoS (EasyQoS)	特定のプロトコルではなく、Network-Based Application Recognition (NBAR) 属性に基づいて QoS クラスとポリシーを定義できるようになりましたが、いくつかの制限があります。サポートされる NBAR 属性は、business-relevance および traffic-class のみです。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 6 章

SDM テンプレートの設定

- [SDM テンプレート \(187 ページ\)](#)
- [SDM テンプレートとスイッチ スタック \(187 ページ\)](#)
- [SDM テンプレートの設定方法 \(188 ページ\)](#)
- [SDM テンプレートのモニターリングおよびメンテナンス \(189 ページ\)](#)
- [SDM テンプレートの設定例 \(189 ページ\)](#)
- [SDM テンプレートに関する追加情報 \(190 ページ\)](#)
- [SDM テンプレートの機能履歴 \(191 ページ\)](#)

SDM テンプレート

SDM テンプレートを使用してシステム リソースを設定すると、特定の機能に対するサポートをネットワーク内でのデバイスの使用方法に応じて最適化することができます。一部の機能に最大システム使用率を提供するようにテンプレートを選択できます。

Cisco Catalyst 4500 シリーズ スイッチは、次のテンプレートをサポートしています。

テンプレートを変更し、システムを再起動した後、**show sdm prefer** 特権 EXEC コマンドを使用して、新しいテンプレート設定を確認できます。**reload** 特権 EXEC コマンドを入力する前に、**show sdm prefer** コマンドを入力すると、**show sdm prefer** コマンドにより、現在使用しているテンプレートおよびリロード後にアクティブになるテンプレートが表示されます。

SDM テンプレートとスイッチ スタック

1つのスイッチスタックで、すべてのスタックメンバにより、アクティブなスイッチに保存された同一の SDM テンプレートを使用する必要があります。新規スイッチがスタックに追加されると、アクティブ スイッチに保存された SDM コンフィギュレーションは、個々のスイッチに設定されているテンプレートを上書きします。

show switch 特権 EXEC コマンドを使用すると、スタック メンバが SDM 不一致モードになっているかどうかを確認できます。

SDM テンプレートの設定方法

SDM テンプレートの設定

SDM テンプレートを使用して機能動作を最適にサポートするには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **sdm prefer access**
4. **end**
5. **reload**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： Device# 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	sdm prefer access 例： Device(config)# sdm prefer access	スイッチをアクセステンプレートに設定します。
ステップ 4	end 例： Device(config)# end	特権 EXEC モードに戻ります。
ステップ 5	reload 例： Device# reload	オペレーティング システムをリロードします。 システムの再起動後、 show sdm prefer 特権 EXEC コマンドを使用して、新しいテンプレート設定を確認できます。 reload 特権 EXEC コマンドを入力する

コマンドまたはアクション	目的
	前に、 show sdm prefer コマンドを入力すると、 show sdm prefer コマンドにより、現在使用しているテンプレートおよびリロード後にアクティブになるテンプレートが表示されます。

SDM テンプレートのモニターリングおよびメンテナンス

コマンド	目的
show sdm prefer	使用中の SDM テンプレートを表示します。
reload	スイッチをリロードして、新しく設定した SDM テンプレートをアクティブにします。



- (注) SDM テンプレートには、テンプレートの一部として定義されているコマンドのみが含まれています。テンプレートで定義されていない別の関連コマンドがテンプレートで有効になっている場合、**show running config** コマンドを入力すると、該当するコマンドが表示されます。たとえば、SDM テンプレートで **switchport voice vlan** コマンドが有効になっている場合、(SDM テンプレートでは定義されていませんが) **spanning-tree portfast edge** コマンドも有効にすることができます。

SDM テンプレートを削除すると、そのような他の関連するコマンドも削除されるため、明示的に再設定しなければなりません。

SDM テンプレートの設定例

例：SDM テンプレートの表示

次に、Cisco Catalyst 9300 シリーズ スイッチのアクセステンプレート情報を表示する出力例を示します。

```
Device# show sdm prefer access
This is the Access template.
Number of VLANs:                4094
Unicast MAC addresses:          32768
Overflow Unicast MAC addresses: 1024
L2 Multicast entries:           8192
Overflow L2 Multicast entries:   512
L3 Multicast entries:           8192
Overflow L3 Multicast entries:   512
```

```

Directly connected routes:                24576
Indirect routes:                          8192
STP Instances:                            1024
Security Access Control Entries:          5120
QoS Access Control Entries:               5120
Policy Based Routing ACEs:                1024
Netflow Input ACEs:                       256
Netflow Output ACEs:                      768
Ingress Netflow ACEs:                     256
Egress Netflow ACEs:                      768
Flow SPAN ACEs:                           1024
Tunnels:                                  512
LISP Instance Mapping Entries:             512
Control Plane Entries:                    512
Input Netflow flows:                      32768
Output Netflow flows:                     32768
SGT/DGT (or) MPLS VPN entries:            8192
SGT/DGT (or) MPLS VPN Overflow entries:   512
Wired clients:                            2048
MACSec SPD Entries:                       256
MPLS L3 VPN VRF:                          255
MPLS Labels:                              2048
MPLS L3 VPN Routes VRF Mode:              7168
MPLS L3 VPN Routes Prefix Mode:          3072
MVPN MDT Tunnels:                         256
L2 VPN EOMPLS Attachment Circuit:         256
MAX VPLS Bridge Domains :                 128
MAX VPLS Peers Per Bridge Domain:         32
MAX VPLS/VPWS Pseudowires :               1024

```

These numbers are typical for L2 and IPv4 features.
Some features such as IPv6, use up double the entry size;
so only half as many entries can be created.
* values can be modified by sdm cli.

例 : SDM テンプレートの設定

```

Device(config)# sdm prefer access
Device(config)# exit
Device# reload
Proceed with reload? [confirm]

```

SDM テンプレートに関する追加情報

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	<i>Command Reference (Catalyst 9300 Series Switches)</i>

SDM テンプレートの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	SDM テンプレート	標準の SDM テンプレートを使用すると、システムリソースを設定して、特定の機能のサポートを最適化できます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 7 章

システム メッセージ ログの設定

- 機能情報の確認 (193 ページ)
- システム メッセージ ログの設定に関する情報 (193 ページ)
- システム メッセージ ログの設定方法 (196 ページ)
- システム メッセージ ログのモニタリングおよびメンテナンス (206 ページ)
- システム メッセージ ログの設定例 (206 ページ)
- システム メッセージ ログの機能履歴と情報 (207 ページ)

機能情報の確認

ご使用のソフトウェアリリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェアリリースの **Bug Search Tool** およびリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、**Cisco Feature Navigator** を使用します。**Cisco Feature Navigator** には、<http://www.cisco.com/go/cfn> からアクセスします。**Cisco.com** のアカウントは必要ありません。

システム メッセージ ログの設定に関する情報

システム メッセージ ロギング

スイッチはデフォルトで、システム メッセージおよび **debug** 特権 EXEC コマンドの出力をロギング プロセスに送信します。ロギング プロセスはログ メッセージを各宛先（設定に応じて、ログバッファ、端末回線、UNIX Syslog サーバなど）に配信する処理を制御します。ロギング プロセスは、コンソールにもメッセージを送信します。

ロギング プロセスがディisable の場合、メッセージはコンソールにのみ送信されます。メッセージは生成時に送信されるため、メッセージおよびデバッグ出力にはプロンプトや他のコマ

ンドの出力が割り込みます。メッセージがアクティブなコンソールに表示されるのは、メッセージを生成したプロセスが終了してからです。

メッセージの重大度を設定して、コンソールおよび各宛先に表示されるメッセージのタイプを制御できます。ログメッセージにタイムスタンプを設定したり、Syslog 送信元アドレスを設定したりして、リアルタイムのデバッグ機能および管理機能を強化できます。表示されるメッセージについては、このリリースに対応するシステムメッセージガイドを参照してください。

ロギングされたシステムメッセージにアクセスするには、スイッチのコマンドラインインターフェイス (CLI) を使用するか、または適切に設定された Syslog サーバにこれらのシステムメッセージを保存します。スイッチソフトウェアは、Syslog メッセージをスタンドアロンスイッチ上の内部バッファに保存します。スタンドアロンスイッチに障害が発生すると、ログをフラッシュメモリに保存していなかった場合、ログは失われます。

システムメッセージをリモートで監視するには、Syslog サーバ上でログを表示するか、あるいは Telnet、コンソールポート、またはイーサネット管理ポート経由でスイッチにアクセスします。



(注) Syslog フォーマットは 4.3 Berkeley Standard Distribution (BSD) UNIX と互換性があります。

システム ログメッセージのフォーマット

システム ログメッセージは最大 80 文字とパーセント記号 (%)、およびその前に配置されるオプションのシーケンス番号やタイムスタンプ情報 (設定されている場合) で構成されています。スイッチに応じて、メッセージは次のいずれかの形式で表示されます。

- *seq no:timestamp: %facility-severity-MNEMONIC:description (hostname-n)*
- *seq no:timestamp: %facility-severity-MNEMONIC:description*

パーセント記号の前にあるメッセージの部分は、次のグローバル コンフィギュレーション コマンドの設定によって異なります。

- **service sequence-numbers**
- **service timestamps log datetime**
- **service timestamps log datetime[localtime] [msec] [show-timezone]**
- **service timestamps log uptime**

表 9: システム ログメッセージの要素

要素	説明
<i>seq no:</i>	service sequence-numbers グローバル コンフィギュレーション コマンドが設定されている場合にのみ、ログメッセージにシーケンス番号をスタンプします。

要素	説明
<p><i>timestamp</i> formats: <i>mm/dd h h:mm:ss</i> または <i>hh:mm:ss</i> (短時間) または <i>d h</i> (長時間)</p>	メッセージまたはイベントの日時です。この情報が表示されるのは、 service timestamps log[datetime log] グローバル コンフィギュレーション コマンドが設定されている場合のみです。
<i>facility</i>	メッセージが参照する機能 (SNMP、SYS など) です。
<i>severity</i>	メッセージの重大度を示す 0 ~ 7 の 1 桁のコードです。
<i>MNEMONIC</i>	メッセージを一意に示すテキスト ストリングです。
<i>description</i>	レポートされているイベントの詳細を示すテキスト ストリングです。
<i>hostname-n</i> (ホスト名 -n)	スタック メンバーのホスト名およびスタック内のスイッチ番号。アクティブスイッチはスタック メンバですが、そのホスト名はシステム メッセージの末尾に追加されません。

デフォルトのシステムメッセージロギングの設定

表 10: デフォルトのシステムメッセージロギングの設定

機能	デフォルト設定
コンソールへのシステムメッセージロギング	イネーブル
コンソールの重大度	デバッグ
ログファイル設定	ファイル名の指定なし
ログバッファサイズ	4096 バイト
ログ履歴サイズ	1 メッセージ
タイムスタンプ	ディセーブル
同期ロギング	ディセーブル
ロギングサーバ	ディセーブル
Syslog サーバの IP アドレス	未設定

機能	デフォルト設定
サーバ機能	local7
サーバの重大度	通知

syslog メッセージの制限

snmp-server enable trap 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、SNMP ネットワーク管理ステーションに送信されるようにsyslogメッセージトラップが設定されている場合は、スイッチの履歴テーブルに送信および格納されるメッセージの重大度を変更できます。また、履歴テーブルに格納されるメッセージの数を変更することもできます。

SNMPトラップは宛先への到達が保証されていないため、メッセージは履歴テーブルに格納されます。デフォルトでは、syslogトラップが有効でない場合も、レベルが **warning** であるメッセージや数値的に下位レベルのメッセージの1つが履歴テーブルに格納されます。

履歴テーブルがいっぱいの場合 (**logging history size** グローバルコンフィギュレーションコマンドで指定した最大メッセージエントリ数に達している場合) は、新しいメッセージエントリを格納できるように、最も古いエントリがテーブルから削除されます。

履歴テーブルは、levelキーワードおよび重大度を示します。SNMPを使用している場合は、重大度の値が1だけ増えます。たとえば、*emergencies* は0ではなく1に、*critical* は2ではなく3になります。

システムメッセージログの設定方法

メッセージ表示宛先デバイスの設定

メッセージロギングがイネーブルの場合、コンソールだけでなく特定の場所にもメッセージを送信できます。

このタスクはオプションです。

手順の概要

1. **configure terminal**
2. **logging buffered** *[size]*
3. **logging** *host*
4. **logging file flash:** *filename* [*max-file-size* [*min-file-size*]] [*severity-level-number* | *type*]
5. **end**
6. **terminal monitor**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : デバイス# 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	logging buffered [size] 例 : デバイス (config)# logging buffered 8192	<p>スイッチ上、またはスタンドアロンスイッチ上、あるいはスイッチスタックの場合はアクティブスイッチ上で、ログメッセージを内部バッファに保存します。指定できる範囲は 4096 ~ 2147483647 バイトです。デフォルトのバッファ サイズは 4096 バイトです。</p> <p>スタンドアロンスイッチまたはアクティブスイッチに障害が発生すると、ログファイルをフラッシュメモリに保存していなかった場合、ログファイルは失われます。ステップ 4 を参照してください。</p> <p>(注) バッファ サイズを大きすぎる値に設定しないでください。他の作業に使用するメモリが不足することがあります。スイッチ上の空きプロセスメモリを表示するには、show memory 特権 EXEC コマンドを使用します。ただし、表示される値は使用できる最大値であるため、バッファ サイズをこの値に設定しないでください。</p>
ステップ 3	logging host 例 : デバイス (config)# logging 125.1.1.100	<p>UNIX Syslog サーバ ホストにメッセージを保存します。</p> <p><i>host</i> には、syslog サーバとして使用するホストの名前または IP アドレスを指定します。</p> <p>ログメッセージを受信する Syslog サーバのリストを作成するには、このコマンドを複数回入力します。</p>
ステップ 4	logging file flash: filename [max-file-size [min-file-size]] [severity-level-number type] 例 : デバイス (config)# logging file flash:log_msg.txt 40960 4096 3	<p>スタンドアロンスイッチ上か、または、スイッチスタックの場合はアクティブスイッチ上で、フラッシュメモリにあるファイルにログメッセージを保存します。</p> <ul style="list-style-type: none"> • <i>filename</i> : ログメッセージのファイル名を入力します。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • (任意) max-file-size —には、ログファイルの最大サイズを指定します。指定できる範囲は 4096 ~ 2147483647 です。デフォルトは 4096 バイトです。 • (任意) min-file-size : ログファイルの最小サイズを指定します。指定できる範囲は 1024 ~ 2147483647 です。デフォルトは 2048 バイトです。 • (任意) severity-level-number type : ログイングの重大度またはログイングタイプを指定します。重大度に指定できる範囲は 0 ~ 7 です。
ステップ 5	end 例 : デバイス (config) # end	特権 EXEC モードに戻ります。
ステップ 6	terminal monitor 例 : デバイス # terminal monitor	現在のセッション間、非コンソール端末にメッセージを保存します。 端末パラメータ コンフィギュレーション コマンドはローカルに設定され、セッションの終了後は無効になります。デバッグメッセージを表示する場合は、セッションごとにこのステップを実行する必要があります。

ログメッセージの同期化

特定のコンソールポート回線または仮想端末回線に対して、非送信請求メッセージおよび **debug** 特権 EXEC コマンドの出力を送信請求デバイスの出力およびプロンプトと同期させることができます。重大度に応じて非同期に出力されるメッセージのタイプを特定できます。また、端末の非同期メッセージが削除されるまで保存しておくバッファの最大数を設定することもできます。

非送信請求メッセージおよび **debug** コマンド出力の同期ログイングがイネーブルの場合、送信請求デバイス出力がコンソールに表示または印刷された後に、非送信請求デバイスからの出力が表示または印刷されます。非送信請求メッセージおよび **debug** コマンドの出力は、ユーザ入力プロンプトが返された後に、コンソールに表示されます。したがって、非送信請求メッセージおよび **debug** コマンドの出力は、送信請求デバイス出力およびプロンプトに割り込まれることはありません。非送信請求メッセージが表示された後に、コンソールはユーザプロンプトを再表示します。

このタスクはオプションです。

手順の概要

1. **configure terminal**
2. **line [console | vty] line-number [ending-line-number]**
3. **logging synchronous [level [severity-level | all] | limit number-of-buffers]**
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# 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	line [console vty] line-number [ending-line-number] 例： デバイス (config) # line console	メッセージの同期ロギングに設定する回線を指定します。 <ul style="list-style-type: none"> • console : スイッチコンソールポートまたはイーサネット管理ポートでの設定を指定します。 • line vty line-number : どの vty 回線の同期ロギングをイネーブルにするかを指定します。Telnet セッションを介して行われる設定には、vty 接続を使用します。回線番号に指定できる範囲は 0 ~ 15 です。 16 個の vty 回線の設定をすべて一度に変更するには、次のように入力します。 line vty 0 15 また、現在の接続に使用されている 1 つの vty 回線の設定を変更することもできます。たとえば、vty 回線 2 の設定を変更するには、次のように入力します。 line vty 2 このコマンドを入力すると、ライン コンフィギュレーション モードになります。
ステップ 3	logging synchronous [level [severity-level all] limit number-of-buffers] 例： デバイス (config) # logging synchronous level 3 limit 1000	メッセージの同期ロギングをイネーブルにします。 <ul style="list-style-type: none"> • (任意) level severity-level : メッセージの重大度レベルを指定します。重大度がこの値以上であるメッセージは、非同期に出力されます。値が小さいほど重大度は大きく、値が大きいほど

	コマンドまたはアクション	目的
		<p>重大度は小さくなります。デフォルトは2です。</p> <ul style="list-style-type: none"> • (任意) level all : 重大度に関係なく、すべてのメッセージが非同期に出力されます。 • (任意) limit number-of-buffers : キューイングされる端末のバッファ数を指定します。これを超える新しいメッセージは廃棄されます。指定できる範囲は0～2147483647です。デフォルトは20です。
ステップ 4	<p>end</p> <p>例 :</p> <p>デバイス(config)# end</p>	特権 EXEC モードに戻ります。

メッセージロギングのディセーブル化

メッセージロギングはデフォルトでイネーブルに設定されています。コンソール以外のいずれかの宛先にメッセージを送信する場合は、メッセージロギングをイネーブルにする必要があります。メッセージロギングがイネーブルの場合、ログメッセージはロギングプロセスに送信されます。ロギングプロセスは、メッセージを生成元プロセスと同期しないで指定場所に記録します。

ロギングプロセスをディセーブルにすると、メッセージがコンソールに書き込まれるまでプロセスは処理続行を待機する必要があるため、スイッチの処理速度が低下することがあります。ロギングプロセスがディセーブルの場合、メッセージは生成後すぐに（通常はコマンド出力に割り込む形で）コンソールに表示されます。

logging synchronous グローバルコンフィギュレーションコマンドも、コンソールへのメッセージ表示に影響します。このコマンドをイネーブルにすると、**Return**を押さなければメッセージが表示されません。

メッセージロギングをディセーブルにした後に再びイネーブルにするには、**logging on** 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。

このタスクはオプションです。

手順の概要

1. **configure terminal**
2. **no logging console**
3. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# 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	no logging console 例： デバイス(config)# no logging console	メッセージ ログイングをディセーブルにします。
ステップ 3	end 例： デバイス(config)# end	特権 EXEC モードに戻ります。

ログメッセージのタイムスタンプのイネーブル化およびディセーブル化

デフォルトでは、ログメッセージにはタイムスタンプが適用されません。

このタスクはオプションです。

手順の概要

1. **configure terminal**
2. 次のいずれかのコマンドを使用します。
 - **service timestamps log uptime**
 - **service timestamps log datetime[msec | localtime | show-timezone]**
3. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# 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。

	コマンドまたはアクション	目的
ステップ 2	<p>次のいずれかのコマンドを使用します。</p> <ul style="list-style-type: none"> • service timestamps log uptime • service timestamps log datetime[msec localtime show-timezone] <p>例 :</p> <pre>デバイス(config)# service timestamps log uptime</pre> <p>または</p> <pre>デバイス(config)# service timestamps log datetime</pre>	<p>ログのタイムスタンプをイネーブルにします。</p> <ul style="list-style-type: none"> • log uptime : ログメッセージのタイムスタンプをイネーブルにして、システムの再起動以降の経過時間を表示します。 • log datetime : ログメッセージのタイムスタンプをイネーブルにします。選択したオプションに応じて、ローカルタイムゾーンを基準とした日付、時間（ミリ秒）、タイムゾーン名をタイムスタンプとして表示できます。
ステップ 3	<p>end</p> <p>例 :</p> <pre>デバイス(config)# end</pre>	<p>特権 EXEC モードに戻ります。</p>

ログメッセージのシーケンス番号のイネーブル化およびディセーブル化

タイムスタンプが同じログメッセージが複数ある場合、これらのメッセージを表示するには、シーケンス番号を使用してメッセージを表示できます。デフォルトでは、ログメッセージにシーケンス番号は表示されません。

このタスクはオプションです。

手順の概要

1. **configure terminal**
2. **service sequence-numbers**
3. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>configure terminal</p> <p>例 :</p> <pre>デバイス# configure terminal</pre>	<p>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。</p>

	コマンドまたはアクション	目的
ステップ 2	service sequence-numbers 例 : デバイス (config) # service sequence-numbers	シーケンス番号をイネーブルにします。
ステップ 3	end 例 : デバイス (config) # end	特権 EXEC モードに戻ります。

メッセージ重大度の定義

メッセージの重大度を指定して、選択したデバイスに表示されるメッセージを制限します。
このタスクはオプションです。

手順の概要

1. **configure terminal**
2. **logging console level**
3. **logging monitor level**
4. **logging trap level**
5. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例 : デバイス # 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	logging console level 例 : デバイス (config) # logging console 3	コンソールに保存するメッセージを制限します。 デフォルトで、コンソールはデバッグメッセージ、および数値的により低いレベルのメッセージを受信します。
ステップ 3	logging monitor level 例 :	端末回線に出力するメッセージを制限します。

	コマンドまたはアクション	目的
	デバイス(config)# logging monitor 3	デフォルトで、端末はデバッグメッセージ、および数値的により低いレベルのメッセージを受信します。
ステップ 4	logging trap level 例： デバイス(config)# logging trap 3	Syslog サーバに保存するメッセージを制限します。 デフォルトで、Syslog サーバは通知メッセージ、および数値的により低いレベルのメッセージを受信します。
ステップ 5	end 例： デバイス(config)# end	特権 EXEC モードに戻ります。

履歴テーブルおよび SNMP に送信される syslog メッセージの制限

このタスクでは、履歴テーブルおよび SNMP に送信される syslog メッセージを制限する方法について説明します。

このタスクはオプションです。

手順の概要

1. **configure terminal**
2. **logging history level**
3. **logging history size number**
4. **end**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# 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	logging history level 例： デバイス(config)# logging history 3	履歴ファイルに保存され、SNMP サーバに送信される syslog メッセージのデフォルトレベルを変更します。 デフォルトでは warnings 、 errors 、 critical 、 alerts 、および emergencies メッセージは送信されません。

	コマンドまたはアクション	目的
ステップ 3	logging history size number 例： デバイス(config)# logging history size 200	履歴テーブルに保存できる Syslog メッセージの数を指定します。 デフォルトでは 1 つのメッセージが格納されます。指定できる範囲は 0 ~ 500 です。
ステップ 4	end 例： デバイス(config)# end	特権 EXEC モードに戻ります。

UNIX Syslog デーモンへのメッセージのロギング

このタスクはオプションです。



- (注) 最新バージョンの UNIX Syslog デーモンの中には、デフォルトでネットワークからの Syslog パケットを受け入れないものがあります。このようなシステムの場合に、Syslog メッセージのリモートロギングをイネーブルにするには、Syslog コマンドラインに追加または削除する必要があるオプションを、UNIX の **man syslogd** コマンドを使用して判別します。

始める前に

- root としてログインします。
- システム ログメッセージを UNIX Syslog サーバに送信する前に、UNIX サーバ上で Syslog デーモンを設定する必要があります。

手順の概要

1. /etc/syslog.conf ファイルに次の行を追加します。
2. UNIX シェルプロンプトに次のコマンドを入力します。
3. Syslog デーモンに新しい設定を認識させます。

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	/etc/syslog.conf ファイルに次の行を追加します。 例： local7.debug /usr/adm/logs/cisco.log	<ul style="list-style-type: none"> • local7 : ロギング機能を指定します。 • debug : syslog レベルを指定します。このファイルは、syslog デーモンに書き込み権限がある既存ファイルである必要があります。

	コマンドまたはアクション	目的
ステップ 2	UNIX シェルプロンプトに次のコマンドを入力します。 例： \$ touch /var/log/cisco.log \$ chmod 666 /var/log/cisco.log	ログファイルを作成します。syslog デーモンは、このレベルまたはこのファイルのより高い重大度レベルでメッセージを送信します。
ステップ 3	Syslog デーモンに新しい設定を認識させます。 例： \$ kill -HUP `cat /etc/syslog.pid`	詳細については、ご使用の UNIX システムの man syslog.conf および man syslogd コマンドを参照してください。

システムメッセージログのモニタリングおよびメンテナンス

コンフィギュレーションアーカイブログのモニタリング

コマンド	目的
show archive log config { all number [end-number] user <i>username</i> [session number] <i>number</i> [end-number] statistics } [provisioning]	コンフィギュレーションログ全体、または指定されたパラメータのログを表示します。

システムメッセージログの設定例

例：システムメッセージのスタック構成

次の例では、アクティブスタックとスタックメンバー（ホスト名は *Switch-2*）に対応するスイッチシステムメッセージの一部分を示します。

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0/1, changed
state to down 2
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

```
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)

00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up (Switch-2)
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet2/0/1, changed state to up (Switch-2)
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet2/0/2, changed state to up (Switch-2)
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
(Switch-2)
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet2/0/1, changed
state to down 2 (Switch-2)
```

例：スイッチ システム メッセージ

次に、スイッチ上のスイッチ システム メッセージの一部を示します。

```
00:00:46: %LINK-3-UPDOWN: Interface Port-channel1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
00:00:47: %LINK-3-UPDOWN: Interface GigabitEthernet0/2, changed state to up
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
00:00:48: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed
state to down 2
*Mar 1 18:46:11: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
18:47:02: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
*Mar 1 18:48:50.483 UTC: %SYS-5-CONFIG_I: Configured from console by vty2 (10.34.195.36)
```

システム メッセージ ログの機能履歴と情報

リリース	変更内容
Cisco IOS XE Everest 16.5.1a	この機能が導入されました。



第 8 章

オンライン診断の設定

- [オンライン診断の設定に関する情報](#) (209 ページ)
- [オンライン診断の設定方法](#) (211 ページ)
- [オンライン診断のモニタリングおよびメンテナンス](#) (217 ページ)
- [オンライン診断テストの設定例](#) (217 ページ)
- [オンライン診断に関する追加情報](#) (219 ページ)
- [オンライン診断設定の機能情報](#) (220 ページ)

オンライン診断の設定に関する情報

オンライン診断機能を使用すると、デバイスをアクティブネットワークに接続したまま、デバイスのハードウェア機能をテストして確認できます。オンライン診断には、異なるハードウェアコンポーネントをチェックするパケット交換テストが含まれ、データパスおよび制御信号が確認されます。

オンライン診断では、次の領域の問題が検出されます。

- ハードウェア コンポーネント
- インターフェイス (イーサネット ポートなど)
- はんだ接合

オンライン診断は、オンデマンド診断、スケジュール診断、ヘルスマニタリング診断に分類できます。オンデマンド診断は、CLIから実行されます。スケジュールされた診断は、動作中のネットワークにデバイスが接続されているときに、ユーザが指定した間隔または指定した時刻に実行されます。ヘルスマニタリングは、バックグラウンドでユーザが指定した間隔で実行されます。ヘルスマニタリングテストは、テストに基づいて 90、100、または 150 秒ごとに実行されます。

オンライン診断を設定したあと、手動で診断テストを開始したり、テスト結果を表示したりできます。また、デバイスまたはスイッチスタックに設定されているテストの種類、およびすでに実行された診断テスト名を確認できます。

Generic Online Diagnostics (GOLD)



- (注)
- オンライン診断テストをイネーブルにする前に、コンソールログをイネーブルにしてすべての警告メッセージを表示してください。
 - テストの実行中、ポートを内部的にループしてストレステストを行いますが、外部トラフィックがテスト結果に影響を与えることがあるため、すべてのポートがシャットダウンされます。スイッチを正常な稼働に戻すために、スイッチをリロードしなければなりません。スイッチをリロードするコマンドを入力すると、コンフィギュレーションを保存するかどうかを聞かれます。
 - コンフィギュレーションは保存しないでください。
 - 他のモジュール上でテストを実行している場合、テストが開始され、完了したら、モジュールをリセットする必要があります。

TestPortTxMonitoring

このテストは、UP ステータスのデバイスに物理的に接続されている各ネットワークポートの送信方向のデータパストラフィックを定期的に監視し、ポートあたり1ミリ秒以内に完了します。また、このテストでは、ASIC レベルで送信カウンタをモニターして、ポートがスタックしていないことを確認します。テストでは syslog メッセージが表示され、ユーザーは IOS Embedded Event Manager (EEM) を使用して修正アクションを実行できます。 **diagnostic monitor interval** および **diagnostic monitor threshold** コマンドをそれぞれ入力して、時間間隔としきい値を設定します。テストでは、パケットを送信する Cisco Discovery Protocol (CDP) プロトコルを利用します。テストは 75 秒ごとに実行され、障害しきい値はデフォルトで 5 に設定されています。

属性	説明
ディスラプティブまたはノンディスラプティブ	ノンディスラプティブ
推奨事項	ディセーブルにしないでください。
Default	オン
最初のリリース	IOS XE Everest 16.9.1.
修正処置	障害の発生したポートを示す Syslog メッセージを表示します。
ハードウェア サポート	スーパーバイザ エンジンを含むすべてのモジュール

TestUnusedPortLoopback

このテストは、実行時にスーパーバイザモジュールとモジュールのネットワークポート間のデータパスを定期的を確認し、着信ネットワーク インターフェイス ポートがロックされているかどうかを判断します。このテストでは、レイヤ2の packets はテストポートおよびスーパーバイザ エンジンのインバンド ポートに関連付けられた VLAN にフラッドされます。パケットはテストポート内をループバックして、同じ VLAN のスーパーバイザエンジンに戻ります。このテストは、ケーブルが接続されているかどうかに関係なく、未使用の（管理上のダウン、つまりポートがシャットダウンされている）ネットワークポートでのみ実行され、ポートあたり1ミリ秒以内に完了します。このテストは、現在の ASIC にノンディスラプティブループバック テストがないため、代用として使用します。このテストは、60 秒ごとに実行されません。

属性	説明
ディスラプティブまたはノンディスラプティブ	ノンディスラプティブ
推奨事項	ディセーブルにしないでください。CPU 使用率の急上昇中、このテストは精度を維持するために自動的にディセーブルになります。
Default	オン
最初のリリース	IOS XE Everest 16.9.1.
修正処置	障害が発生したポートを示す Syslog メッセージを表示します。スーパーバイザ エンジン以外のモジュールでは、すべてのポートグループに障害が発生した場合（たとえば、ポート ASIC ごとに最低1つのポートで、すべてのポート ASIC の障害しきい値より多く障害が発生した場合）、デフォルトのアクションではモジュールがリセットされ、リセットを2回行ったあとにモジュールの電源を切断します。
ハードウェア サポート	スーパーバイザ エンジンを含むすべてのモジュール

オンライン診断の設定方法

オンライン診断テストの開始

デバイスで実行する診断テストを設定したあと、**diagnostic start** 特権 EXEC コマンドを使用して診断テストを開始します。

テストを開始したら、テストプロセスの停止はできません。

手動でオンライン診断テストを開始するには、次の特権 EXEC コマンドを使用します。

手順の概要

1. **diagnostic start switch number test** {*name* | *test-id* | *test-id-range* | **all** | **basic** | **complete** | **minimal** | **non-disruptive** | **per-port**}

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>diagnostic start switch number test {<i>name</i> <i>test-id</i> <i>test-id-range</i> all basic complete minimal non-disruptive per-port}</p> <p>例 :</p> <pre>デバイス# diagnostic start switch 2 test basic</pre>	<p>診断テストを開始します。</p> <p>switch number キーワードは、スタック構成デバイスのみでサポートされます。</p> <p>次のいずれかのオプションを使用してテストを指定できます。</p> <ul style="list-style-type: none"> • name : テストの名前を入力します。 • test-id : テストの ID 番号を入力します。 • test-id-range : カンマとハイフンで区切ってテスト ID の範囲を整数で入力します。 • all : すべてのテストを開始します。 • basic : 基本テストスイートを開始します。 • complete : 完全なテストスイートを開始します。 • minimal : 最小限のブートアップテストスイートを開始します。 • non-disruptive : ノンディスラプティブテストスイートを開始します。 • per-port : ポート単位のテストスイートを開始します。

オンライン診断の設定

診断モニタリングをイネーブルにする前に、障害しきい値およびテストの間隔を設定する必要があります。

オンライン診断のスケジューリング

特定のデバイスについて指定した時間、または日、週、月単位でオンライン診断をスケジューリングできます。スケジューリングを削除するには、コマンドの **no** 形式を入力します。

手順の概要

1. **configure terminal**
2. **diagnostic schedule switch number test {name | test-id | test-id-range | all | basic | complete | minimal | non-disruptive | per-port} {daily | on mm dd yyyy hh:mm | port inter-port-number port-number-list | weekly day-of-week hh:mm}**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>configure terminal</p> <p>例 :</p> <p>デバイス# configure terminal</p>	<p>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。</p>
ステップ 2	<p>diagnostic schedule switch number test {name test-id test-id-range all basic complete minimal non-disruptive per-port} {daily on mm dd yyyy hh:mm port inter-port-number port-number-list weekly day-of-week hh:mm}</p> <p>例 :</p> <p>デバイス(config)# diagnostic schedule switch 3 test 1-5 on July 3 2013 23:10</p>	<p>特定日時のオンデマンド診断テストをスケジュールします。</p> <p>switch number キーワードは、スタック構成スイッチだけでサポートされます。範囲は 1 ~ 8 です。</p> <p>スケジュールするテストを指定する場合は、次のオプションを使用します。</p> <ul style="list-style-type: none"> • name : show diagnostic content コマンドの出力に表示されるテストの名前です。 • test-id : show diagnostic content コマンドの出力に表示されるテストの ID 番号です。 • test-id-range : show diagnostic content コマンドの出力に表示されるテストの ID 番号です。 • all : すべてのテスト ID • basic : 基本的なオンデマンドの診断テストを開始します。 • complete : 完全なテストスイートを開始します。 • minimal : 最小限のブートアップテストスイートを開始します。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • non-disruptive : ノンディスラプティブ テストスイートを開始します。 • per-port : ポート単位のテストスイートを開始します。 <p>テストは次のようにスケジュールできます。</p> <ul style="list-style-type: none"> • 毎日 : daily hh:mm パラメータを使用します。 • 特定日時 : on mm dd yyyy hh:mm パラメータを使用します。 • 毎週 : weekly day-of-week hh:mm パラメータを使用します。

ヘルス モニタリング診断の設定

デバイスが稼働中のネットワークに接続されている間に、スイッチに対しヘルスモニタリング診断テストを設定できます。各ヘルスモニタリングテストの実行間隔を設定したり、デバイスをイネーブルにし、テスト失敗時のsyslogメッセージを生成したり、特定のテストをイネーブルにできます。

テストをディセーブルにするには、コマンドの **no** 形式を入力します。

デフォルトでは、ヘルスモニタリングはディセーブルですが、デバイスはテストの失敗時に Syslog メッセージを生成します。

ヘルスモニタリング診断テストを設定し、イネーブルにするには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **diagnostic monitor interval switchnumber test {name | test-id | test-id-range | all} hh:mm:ss milliseconds day**
4. **diagnostic monitor syslog**
5. **diagnostic monitor threshold switch number number test {name | test-id | test-id-range | all} failure count count**
6. **diagnostic monitor switchnumber test {name | test-id | test-id-range | all}**
7. **end**
8. **show diagnostic { content | post | result | schedule | status | switch }**
9. **show running-config**
10. **copy running-config startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>デバイス> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <pre>デバイス# configure terminal</pre>	<p>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。</p>
ステップ 3	<p>diagnostic monitor interval switchnumber test {name test-id test-id-range all} hh:mm:ss milliseconds day</p> <p>例 :</p> <pre>デバイス (config)# diagnostic monitor interval switch 2 test 1 12:30:00 750 5</pre>	<p>指定のテストに対し、ヘルス モニタリングの実行間隔を設定します。</p> <p>switch number キーワードは、スタック構成スイッチだけでサポートされます。指定できる範囲は1～9です。</p> <p>テストを指定する場合は、次のいずれかのパラメータを使用します。</p> <ul style="list-style-type: none"> • name : show diagnostic content コマンドの出力に表示されるテストの名前です。 • test-id : show diagnostic content コマンドの出力に表示されるテストの ID 番号です。 • test-id-range : show diagnostic content コマンドの出力に表示されるテストの ID 番号です。 • all : すべての診断テスト。 <p>間隔を指定する場合は、次のパラメータを設定します。</p> <ul style="list-style-type: none"> • hh:mm:ss : モニタリング間隔 (時間、分、秒)。指定できる範囲は <i>hh</i> が 0～24、<i>mm</i> および <i>ss</i> が 0～60 です。 • milliseconds : モニタリング間隔 (ミリ秒 (ms))。指定できる範囲は 0～999 です。 • day : モニタリング間隔 (日数)。指定できる範囲は 0～20 です。

	コマンドまたはアクション	目的
ステップ 4	diagnostic monitor syslog 例 : デバイス(config)# diagnostic monitor syslog	(任意) ヘルス モニタリング テストの失敗時にスイッチが Syslog メッセージを生成するように設定します。
ステップ 5	diagnostic monitor threshold switch number number test {name test-id test-id-range all} failure count count 例 : デバイス(config)# diagnostic monitor threshold switch 2 test 1 failure count 20	(任意) ヘルス モニタリング テストの失敗しきい値を設定します。 テストを指定する場合は、次のいずれかのパラメータを使用します。 <ul style="list-style-type: none"> • name : show diagnostic content コマンドの出力に表示されるテストの名前です。 • test-id : show diagnostic content コマンドの出力に表示されるテストの ID 番号です。 • test-id-range : show diagnostic content コマンドの出力に表示されるテストの ID 番号です。 • all : すべての診断テスト。 失敗しきい値 <i>count</i> に指定できる範囲は 0 ~ 99 です。
ステップ 6	diagnostic monitor switch number test {name test-id test-id-range all} 例 : デバイス(config)# diagnostic monitor switch 2 test 1	指定のヘルス モニタリング テストをイネーブルにします。 switch number キーワードは、スタック構成スイッチだけでサポートされます。 テストを指定する場合は、次のいずれかのパラメータを使用します。 <ul style="list-style-type: none"> • name : show diagnostic content コマンドの出力に表示されるテストの名前です。 • test-id : show diagnostic content コマンドの出力に表示されるテストの ID 番号です。 • test-id-range : show diagnostic content コマンドの出力に表示されるテストの ID 番号です。 • all : すべての診断テスト。
ステップ 7	end 例 :	特権 EXEC モードに戻ります。

	コマンドまたはアクション	目的
	デバイス (config) # end	
ステップ 8	show diagnostic { content post result schedule status switch }	オンライン診断のテスト結果およびサポートされるテストスイートを表示します。
ステップ 9	show running-config 例： デバイス # show running-config	入力を確認します。
ステップ 10	copy running-config startup-config 例： デバイス # 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。

オンライン診断のモニタリングおよびメンテナンス

オンライン診断テストの設定例

例：診断テストの開始

次に、テスト名を指定して診断テストを開始する例を示します。

```
Device# diagnostic start switch 2 test DiagPOETest
```

次に、すべての基本診断テストを開始する例を示します。

```
Device# diagnostic start switch 1 test all
```

例：ヘルス モニタリング テストの設定

次に、ヘルス モニタリング テストを設定する例を示します。

```
Device(config)# diagnostic monitor threshold switch 1 test 1 failure count 50
```

```
Device(config)# diagnostic monitor interval switch 1 test TestPortAsicStackPortLoopback
```

例：診断テストのスケジューリング

次に、特定のスイッチに対して、特定の日時に診断テストを実行するようにスケジューリングする例を示します。

```
Device(config)# diagnostic schedule test DiagThermalTest on June 3 2013 22:25
```

次の例では、指定されたスイッチで毎週特定の時間に診断テストを実行するようにスケジューリングする方法を示します。

```
Device(config)# diagnostic schedule switch 1 test 1,2,4-6 weekly saturday 10:30
```

例：オンライン診断の表示

次に、オンデマンド診断設定を表示する例を示します。

```
Device# show diagnostic ondemand settings
```

```
Test iterations = 1
Action on test failure = continue
```

次に、障害の診断イベントを表示する例を示します。

```
Device# show diagnostic events event-type error
```

```
Diagnostic events (storage for 500 events, 0 events recorded)
Number of events matching above criteria = 0

No diagnostic log entry exists.
```

次に、診断テストの説明を表示する例を示します。

```
Device# show diagnostic description switch 1 test all
```

```
DiagGoldPktTest :
  The GOLD packet Loopback test verifies the MAC level loopback
  functionality. In this test, a GOLD packet, for which doppler
  provides the support in hardware, is sent. The packet loops back
  at MAC level and is matched against the stored packet. It is a non-
  -disruptive test.

DiagThermalTest :
  This test verifies the temperature reading from the sensor is below the yellow
  temperature threshold. It is a non-disruptive test and can be run as a health
  monitoring test.

DiagFanTest :
  This test verifies all fan modules have been inserted and working properly on
  the board
  It is a non-disruptive test and can be run as a health monitoring test.

DiagPhyLoopbackTest :
  The PHY Loopback test verifies the PHY level loopback
  functionality. In this test, a packet is sent which loops back
```

at PHY level and is matched against the stored packet. It is a disruptive test and cannot be run as a health monitoring test.

DiagScratchRegisterTest :

The Scratch Register test monitors the health of application-specific integrated circuits (ASICs) by writing values into registers and reading back the values from these registers. It is a non-disruptive test and can be run as a health monitoring test.

DiagPoETest :

This test checks the PoE controller functionality. This is a disruptive test and should not be performed during normal switch operation.

DiagStackCableTest :

This test verifies the stack ring loopback functionality in the stacking environment. It is a disruptive test and cannot be run as a health monitoring test.

DiagMemoryTest :

This test runs the exhaustive ASIC memory test during normal switch operation. NG3K utilizes mbist for this test. Memory test is very disruptive in nature and requires switch reboot after the test.

Device#

オンライン診断に関する追加情報

関連資料

関連項目	マニュアルタイトル
システム管理コマンド	<i>Command Reference (Catalyst 9300 Series Switches)</i> <i>Command Reference (Catalyst 9400 Series Switches)</i>

MIB

MIB	MIB のリンク
本リリースでサポートするすべての MIB	選択したプラットフォーム、Cisco IOS リリース、およびフィチャーセットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

シスコのテクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	http://www.cisco.com/support

オンライン診断設定の機能情報

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	オンライン診断	オンライン診断機能を使用すると、デバイスをアクティブ ネットワークに接続したまま、デバイスのハードウェア機能をテストして確認できます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 9 章

コンフィギュレーション ファイルの管理

- [コンフィギュレーション ファイルの管理の前提条件](#) (221 ページ)
- [コンフィギュレーション ファイルの管理の制約事項](#) (221 ページ)
- [コンフィギュレーション ファイルの管理について](#) (222 ページ)
- [コンフィギュレーション ファイル情報の管理方法](#) (230 ページ)
- [コンフィギュレーション ファイルの管理の機能履歴](#) (263 ページ)

コンフィギュレーション ファイルの管理の前提条件

- ユーザーには、少なくとも Cisco IOS 環境とコマンドラインインターフェイスに関する基本的な知識が必要です。
- システムでは、少なくとも最小限の設定が実行されていることが必要です。基本コンフィギュレーション ファイルは、**setup** コマンドを使用して作成できます。

コンフィギュレーション ファイルの管理の制約事項

- このドキュメントで説明されている Cisco IOS コマンドの多くは、device の特定のコンフィギュレーション モードでのみ使用可能であり機能します。
- Cisco IOS コンフィギュレーション コマンドのいくつかは、特定の device プラットフォームでのみ使用可能であり、コマンド構文はプラットフォームによって異なる可能性があります。

コンフィギュレーションファイルの管理について

コンフィギュレーションファイルのタイプ

コンフィギュレーションファイルには、シスコ製deviceの機能をカスタマイズするためのCisco IOS ソフトウェアコマンドが含まれています。コマンドは、システムを起動したとき (startup-config ファイルから)、またはコンフィギュレーションモードでCLIにコマンドを入力したときに、Cisco IOS ソフトウェアによって解析 (変換および実行) されます。

スタートアップコンフィギュレーションファイル (startup-config) は、ソフトウェアを設定するためにシステムの起動時に使用されます。実行コンフィギュレーションファイル (running-config) には、ソフトウェアの現在の設定が含まれています。2つのコンフィギュレーションファイルは別々の設定にできます。たとえば、コンフィギュレーションを永続的ではなく短期間だけ変更する場合があります。その場合は、**configure terminal EXEC** コマンドを使用して実行コンフィギュレーションを変更しますが、そのコンフィギュレーションは **copy running-config startup-config EXEC** コマンドを使用して保存しません。

実行コンフィギュレーションを変更するには、[コンフィギュレーションファイルの変更 \(231 ページ\)](#) の項で説明されているように、**configure terminal** コマンドを使用します。Cisco IOS コンフィギュレーションモードの使用時には、通常コマンドはすぐに実行され、入力直後またはコンフィギュレーションモードを終了した時点で実行コンフィギュレーションファイルに保存されます。

スタートアップコンフィギュレーションファイルを変更するには、**copy running-config startup-config EXEC** コマンドを使用してスタートアップコンフィギュレーションに実行コンフィギュレーションファイルを保存するか、ファイルサーバーからスタートアップコンフィギュレーションにコンフィギュレーションファイルをコピーします (詳細については、「[TFTP サーバからデバイスへのコンフィギュレーションファイルのコピー](#)」を参照してください)。

コンフィギュレーションモードおよびコンフィギュレーションソースの選択

device上でコンフィギュレーションモードを開始するには、特権EXECプロンプトで**configure** コマンドを入力します。Cisco IOS ソフトウェアは次のプロンプトで応答し、端末、メモリ、またはネットワークサーバー (ネットワーク) 上に格納されたファイルのいずれかを、コンフィギュレーションコマンドのソースとして指定するように要求されます。

```
Configuring from terminal, memory, or network [terminal]?
```

端末からの設定では、コマンドラインにコンフィギュレーションコマンドを入力できます (次の項を参照してください)。詳細については、[スタートアップコンフィギュレーションファイルでのコンフィギュレーションコマンドの再実行](#) の項を参照してください。

ネットワークからの設定では、ネットワーク経由でコンフィギュレーション コマンドをロードして実行できます。詳細については、[TFTP サーバからデバイスへのコンフィギュレーション ファイルのコピー](#) の項を参照してください。

CLI を使用したコンフィギュレーション ファイルの変更

Cisco IOS ソフトウェアは、1 行につき 1 つのコンフィギュレーション コマンドを受け入れません。コンフィギュレーション コマンドは、必要なだけ入力できます。コンフィギュレーション ファイルには、入力したコマンドを説明するコメントを追加できます。コメントの先頭には、感嘆符 (!) を付けます。コメントは NVRAM にもコンフィギュレーション ファイルのアクティブコピーにも格納されないため、**show running-config** または **more system:running-config EXEC** コマンドでアクティブな設定のリストを表示しても、コメントは表示されません。**show startup-config** または **more nvram:startup-config EXEC** モードコマンドでスタートアップ コンフィギュレーションのリストを表示しても、コメントは表示されません。コメントは、コンフィギュレーション ファイルが device にロードされたときにコンフィギュレーション ファイルから削除されます。ただし、ファイル転送プロトコル (FTP)、リモートコピープロトコル (RCP)、または Trivial File Transfer Protocol (TFTP) サーバー上に格納されているコンフィギュレーション ファイルのコメントのリストは表示できます。CLI を使用してソフトウェアは設定するときは、ユーザーの入力に従ってソフトウェアによりコマンドが実行されます。

コンフィギュレーション ファイルの場所

コンフィギュレーション ファイルは、次の場所に格納されます。

- 実行コンフィギュレーションは RAM に格納されます。
- クラス A フラッシュ ファイル システム プラットフォーム以外のすべてのプラットフォーム上では、スタートアップ コンフィギュレーションは不揮発性 RAM (NVRAM) に格納されます。
- クラス A フラッシュ ファイル システムのプラットフォーム上では、スタートアップ コンフィギュレーションは CONFIG_FILE 環境変数で指定された場所に格納されます ([クラス A フラッシュ ファイル システムでの CONFIG_FILE 環境変数の指定 \(257 ページ\)](#) の項を参照してください)。CONFIG_FILE 変数は、デフォルトでは NVRAM になりますが、次のファイル システムのファイルも指定できます。
 - **nvram:** (NVRAM)
 - **flash:** (内部フラッシュ メモリ)
 - **usbflash0:** (外部 usbflash ファイル システム)

ネットワーク サーバからデバイスへのコンフィギュレーションファイルのコピー

TFTP、rcp、またはFTPサーバーからdeviceの実行コンフィギュレーションまたはスタートアップコンフィギュレーションへコンフィギュレーションファイルをコピーできます。この機能は、次のいずれかの理由により実行する場合があります。

- バックアップコンフィギュレーションファイルを復元するため。
- 構成ファイルを別のdeviceに使用するには、次を実行します。たとえば、別のdeviceをネットワークに追加して、そのルータのコンフィギュレーションを元のdeviceと同様にする場合です。ファイルを新しいdeviceにコピーすると、ファイル全体を再作成しないで、関連部分を変更できます。
- 同一のコンフィギュレーションコマンドをネットワーク内のすべてのdeviceにロードして、すべてのdeviceのコンフィギュレーションを同様にするため。

コマンドラインにコマンドを入力した場合と同様に、`copy {ftp|rcp:} {tftp:system:running-config}` EXEC コマンドはdeviceにコンフィギュレーションファイルをロードします。deviceはコマンドを追加するまで、既存の実行コンフィギュレーションは消去されません。コピーされた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既存のコンフィギュレーションファイル内のコマンドが置き換えられると、既存のコマンドは消去されます。たとえば、コピーされたコンフィギュレーションファイルに格納されている特定のコマンドのIPアドレスが、既存のコンフィギュレーションに格納されているIPアドレスと異なる場合は、コピーされたコンフィギュレーション内のIPアドレスが使用されます。ただし、既存のコンフィギュレーション内の一部のコマンドには、置き換えられたり無効になったりしないものもあります。このようなコマンドがある場合は、既存のコンフィギュレーションファイルとコピーされたコンフィギュレーションファイルが組み合わせされた（コピーされたコンフィギュレーションファイルが優先する）コンフィギュレーションファイルが作成されます。

コンフィギュレーションファイルをサーバー上に格納されているファイルの正確なコピーとして復元するには、そのコンフィギュレーションファイルをスタートアップコンフィギュレーションに直接コピーし（`copy ftp|rcp:} tftp:} nvram:startup-config` コマンドを使用）、deviceをリロードする必要があります。

サーバーからdeviceへコンフィギュレーションファイルをコピーするには、次のセクションで説明するタスクを実行します。

使用するプロトコルは、使用中のサーバーのタイプに応じて異なります。FTP および rcp のトランスポートメカニズムは、TFTP よりも高速でデータ配信の信頼性も優れています。これらの改善は、FTP および rcp のトランスポートメカニズムがコネクション型のTCP/IP スタック上に構築されており、これを使用しているために可能になりました。

デバイスから TFTP サーバーへのコンフィギュレーションファイルのコピー

一部の TFTP 実装では、TFTP サーバー上にダミーファイルを作成し、読み取り、書き込み、および実行を許可してから、ダミーファイルを上書きする形でファイルをコピーする必要があります。詳細については、ご使用の TFTP のマニュアルを参照してください。

デバイスから RCP サーバへのコンフィギュレーションファイルのコピー

device から RCP サーバへコンフィギュレーションファイルをコピーできます。

ネットワークを UNIX コミュニティでリソースとして使用する最初の試みの 1 つは、リモートシェル (RSH) およびリモートコピー (rcp) 機能が含まれた、リモートシェルプロトコルの設計および実装につながりました。rsh および rcp により、ユーザーはリモートでコマンドを実行し、ネットワーク上のリモートホストまたはサーバにあるファイルシステムからまたはファイルシステムへファイルをコピーすることが可能になります。シスコの rsh および rcp 実装は、標準実装と相互運用できます。

RCP の **copy** コマンドは、リモートシステム上の rsh サーバ (またはデーモン) を利用します。rcp を使用してファイルをコピーするために、TFTP のようにファイル配布用のサーバを作成する必要はありません。必要なのは、リモートシェル (rsh) をサポートするサーバへのアクセスだけです (ほとんどの UNIX システムが rsh をサポートしています)。ある場所から別の場所にファイルをコピーするため、コピー元のファイルに対する読み取り権限とコピー先のファイルに対する書き込み権限が必要です。コピー先ファイルが存在しない場合は、rcp により作成されます。

シスコの rcp 実装は UNIX の rcp 実装 (ネットワーク上のシステム間でファイルをコピー) の関数をエミュレートしたのですが、シスコのコマンド構文は UNIX の rcp コマンド構文とは異なります。シスコの rcp サポートは、rcp をトランスポートメカニズムとして使用する一連の **copy** コマンドを提供しています。これらの **rcp copy** コマンドは、シスコの TFTP **copy** コマンドに類似していますが、高速で信頼性の高いデータ配信を実現する代替方法を備えているという点が異なります。これらの改善は、rcp のトランスポートメカニズムがコネクション型の TCP/IP スタック上に構築されており、これを使用しているために可能になりました。rcp コマンドを使用して、device からネットワークサーバ (またはその逆) へシステムイメージおよびコンフィギュレーションファイルをコピーできます。

また、rcp サポートをイネーブルにし、リモートシステムのユーザーが device からまたはデバイスへファイルをコピーできるようにすることも可能です。

リモートユーザーが device との間でファイルをコピーできるように Cisco IOS ソフトウェアを設定するには、**ip rcmd rcp-enable** 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。

機能制限

RCP プロトコルでは、クライアントは RCP 要求ごとにリモートユーザー名をサーバに送信する必要があります。RCP を使用して device からサーバへコンフィギュレーションファイルをコピーする場合、Cisco IOS ソフトウェアは次の順番で最初に発見した有効なユーザー名を送信します。

1. **copy EXEC** コマンドで指定されたユーザ名 (ユーザ名が指定されている場合)。
2. **ip rcmd remote-username** グローバルコンフィギュレーションコマンドで設定されたユーザー名 (コマンドが設定されている場合)。
3. 現在の TTY (端末) プロセスに関連付けられているリモートユーザー名。たとえば、ユーザーが Telnet を介して device に接続されており、**username** コマンドを介して認証された場

合は、リモートユーザー名として Telnet ユーザー名が device ソフトウェアによって送信されます。

4. device ホスト名。

RCP コピー要求を正常に実行するためには、ネットワーク サーバー上にリモートユーザー名のアカウントを定義する必要があります。このサーバーがディレクトリ構造をとっている場合、コンフィギュレーションファイルまたはイメージは、サーバー上のリモートユーザー名と関連付けられたディレクトリに書き込まれるか、そのディレクトリからコピーされます。たとえば、システムイメージがサーバー上のユーザーのホーム ディレクトリにある場合は、そのユーザーの名前をリモートユーザー名として指定できます。

ip rcmd remote-username コマンドを使用して、すべてのコピーに対してユーザー名を指定します。(rcmd は、スーパーユーザー レベルで使用される UNIX ルーチンで、予約されたポート番号に基づいた認証スキームを使用してリモート マシン上でコマンドを実行します。rcmd は「Remote Command (リモート コマンド)」の略です)。特定のコピー操作にのみ使用するユーザー名を指定する場合は、**copy** コマンド内でユーザー名を指定します。

サーバーに書き込む場合、device 上のユーザーからの RCP 書き込み要求を受け入れるように、RCP サーバーを適切に設定する必要があります。UNIX システムの場合は、RCP サーバ上のリモートユーザー用の `.rhosts` ファイルにエントリを追加する必要があります。たとえば、device に次の設定行が含まれているとします。

```
hostname デバイス1
ip rcmd remote-username User0
```

device の IP アドレスが `device1.example.com` に変換される場合、RCP サーバー上の User0 の `.rhosts` ファイルには、次の行が含まれることになります。

```
デバイス1.example.com デバイス1
```

RCP ユーザー名に関する要件

RCP プロトコルでは、クライアントは RCP 要求ごとにリモートユーザー名をサーバーに送信する必要があります。RCP を使用して device からサーバーにコンフィギュレーションファイルをコピーする場合、Cisco IOS ソフトウェアは次の順番で最初に発見した有効なユーザー名を送信します。

1. **copy EXEC** コマンドで指定されたユーザ名 (ユーザ名が指定されている場合)。
2. **ip rcmd remote-username** グローバルコンフィギュレーションコマンドで設定されたユーザー名 (コマンドが設定されている場合)。
3. 現在の TTY (端末) プロセスに関連付けられているリモートユーザー名。たとえば、ユーザーが Telnet を介して device に接続されており、**username** コマンドを介して認証されている場合は、リモートユーザー名として Telnet ユーザー名が device ソフトウェアによって送信されます。
4. device ホスト名。

RCP コピー要求を実行するためには、ネットワーク サーバー上にリモートユーザー名のアカウントを定義する必要があります。このサーバーがディレクトリ構造をとっている場合、コンフィギュレーションファイルまたはイメージは、サーバー上のリモートユーザー名と関連付けられたディレクトリに書き込まれるか、そのディレクトリからコピーされます。たとえば、システムイメージがサーバー上のユーザーのホームディレクトリにある場合は、そのユーザーの名前をリモートユーザー名として指定します。

詳細については、ご使用の RCP サーバーのマニュアルを参照してください。

デバイスから FTP サーバーへのコンフィギュレーションファイルのコピー

deviceから FTP サーバーにコンフィギュレーションファイルをコピーできます。

FTP ユーザ名およびパスワードの概要



(注) パスワードに特殊文字「@」を含めることはできません。文字「@」を使用すると、コピーでサーバの IP アドレスを解析できません。

FTP プロトコルでは、FTP 要求ごとにリモートユーザ名およびパスワードを、クライアントがサーバに送信する必要があります。FTP を使用してデバイスからサーバへコンフィギュレーションファイルをコピーする場合、Cisco IOS ソフトウェアは次の順番で最初に発見した有効なユーザ名を送信します。

1. **copy EXEC** コマンドで指定されたユーザ名（ユーザ名が指定されている場合）。
2. **ip ftp username** グローバル コンフィギュレーション コマンドで設定されたユーザ名（コマンドが設定されている場合）。
3. Anonymous

デバイスは、次の順番で最初に発見した有効なパスワードを送信します。

1. **copy** コマンドで指定されたパスワード（パスワードが指定されている場合）。
2. **ip ftp password** コマンドで設定されたパスワード（コマンドが設定されている場合）。
3. デバイスは、*username@devicename.domain* というパスワードを生成します。変数 *username* は現在のセッションに関連付けられたユーザ名、*devicename* は設定済みのホスト名、*domain* はデバイスのドメインです。

ユーザ名およびパスワードは、FTP サーバのアカウントに関連付けられている必要があります。サーバに書き込む場合、デバイス上のユーザからの FTP 書き込み要求を受け入れるように、FTP サーバを適切に設定する必要があります。

このサーバがディレクトリ構造をとっている場合、コンフィギュレーションファイルまたはイメージは、サーバ上のユーザ名と関連付けられたディレクトリに書き込まれるか、そのディレクトリからコピーされます。たとえば、システムイメージがサーバ上のユーザのホームディレクトリにある場合は、そのユーザの名前をリモートユーザ名として指定します。

詳細については、ご使用の FTP サーバのマニュアルを参照してください。

すべてのコピー操作に使用するユーザ名およびパスワードを指定するには、**ip ftp username** および **ip ftp password** 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。特定のコピー操作にのみ使用するユーザ名を指定する場合は、**copy EXEC** コマンド内でユーザ名を指定します。

VRF によるファイルのコピー

copy コマンドで指定した VRF インターフェイス経由でファイルをコピーできます。設定の変更リクエストを使用せずに直接送信元インターフェイスを変更できるので、**copy** コマンドで VRF を指定するほうが簡単で効率的です。

例

次の例に、**copy** コマンドを使用して VRF 経由でファイルをコピーする方法を示します。

```

デバイス# copy scp: flash-1: vrf test-vrf
Address or name of remote host [10.1.2.3]?
Source username [ScpUser]?
Source filename [/auto/tftp-server/ScpUser/vrf_test.txt]?
Destination filename [vrf_test.txt]?
Getting the vrf name as test-vrf
Password:
Sending file modes: C0644 10 vrf_test.txt
!
223 bytes copied in 22.740 secs (10 bytes/sec)
    
```

スイッチから別のスイッチへのコンフィギュレーションファイルのコピー

あるスイッチから別のスイッチに設定をコピーすることができます。これは2ステッププロセスです。スイッチから TFTP サーバーに設定をコピーし、次に TFTP から別のスイッチに設定をコピーします。

スイッチから現在の設定をコピーするには、**copy startup-config tftp:** コマンドを実行し、続く指示に従います。設定が TFTP サーバにコピーされます。

次に、別のスイッチへログインし、**copy tftp: startup-config** コマンドを実行して、続く指示に従います。これで、設定は別のスイッチにコピーされます。

設定をコピーした後、その設定を保存するには、**write memory** コマンドを使用し、その後スイッチをリロードするか、または **copy startup-config running-config** コマンドを実行します。

NVRAM より大きいコンフィギュレーションファイル

NVRAM より大きいコンフィギュレーションファイルを維持管理するには、以降の項の情報を知っておく必要があります。

コンフィギュレーションファイルの圧縮

service compress-config グローバル コンフィギュレーション コマンドは、コンフィギュレーション ファイルを圧縮して NVRAM に格納することを指定します。コンフィギュレーション ファイルが圧縮されると、**device**は正常に機能します。システムの起動時に、システムはコンフィギュレーション ファイルが圧縮されていることを認識し、圧縮されたコンフィギュレーション ファイルを展開して、正常に処理を進めます。**more nvram:startup-config EXEC** コマンドにより、コンフィギュレーションが展開されてから表示されます。

コンフィギュレーションファイルを圧縮する前に、適切なハードウェアのインストレーション およびメンテナンス マニュアルを参照してください。ご利用のシステムのROMがファイル圧縮をサポートしていることを確認します。サポートしていない場合、ファイル圧縮をサポートしている新しいROMをインストールできます。

コンフィギュレーションのサイズは、NVRAMのサイズの3倍を超えてはいけません。NVRAMのサイズが128 KBの場合、展開できる最大のコンフィギュレーションファイルのサイズは384 KBです。

service compress-config グローバル コンフィギュレーション コマンドは、Cisco IOS ソフトウェア リリース 10.0 以降のブート ROM を使用している場合に限り実行できます。新しいROMをインストールするのは1回限りの操作で、ROMにCisco IOS Release 10.0がない場合だけ必要です。ブートROMが圧縮コンフィギュレーションを認識しない場合は、次のメッセージが表示されます。

```
Boot ROMs do not support NVRAM compression Config NOT written to NVRAM
```

コンフィギュレーションのクラス A フラッシュ ファイル システム上のフラッシュ メモリへの格納

クラス A フラッシュ ファイル システムの**device**上では、内部フラッシュ メモリのファイルまたは PCMCIA スロットのフラッシュ メモリのファイルに **CONFIG_FILE** 環境変数を設定することにより、スタートアップ コンフィギュレーションをフラッシュ メモリに格納できます。

詳細については、[クラス A フラッシュ ファイル システムでの CONFIG_FILE 環境変数の指定 \(257 ページ\)](#) を参照してください。

大きいコンフィギュレーションを編集または変更する場合は、注意する必要があります。フラッシュ メモリ領域は **copy system:running-config nvram:startup-config EXEC** コマンドが発行されるたびに使用されます。フラッシュ メモリのファイル管理 (空き領域の最適化などの) は自動的に行われなため、利用可能なフラッシュ メモリに十分注意を払う必要があります。**squeeze** コマンドを使用して、使用済み領域を再要求します。20 MB 以上の大容量フラッシュ カードを使用することを推奨します。

ネットワークからのコンフィギュレーション コマンドのロード

コンフィギュレーションが大きい場合は、FTP、RCP、TFTPのいずれかのサーバーに格納しておき、システムの起動時にダウンロードすることもできます。ネットワークサーバーを使用して大規模な設定を格納するには、[デバイスから TFTP サーバーへのコンフィギュレーション ファイルのコピー \(233 ページ\)](#) および [コンフィギュレーションファイルをダウンロードするデバイスの設定 \(230 ページ\)](#) の項でこれらのコマンドの詳細を参照してください。

コンフィギュレーションファイルをダウンロードするデバイスの設定

システムの起動時に1つまたは2つのコンフィギュレーションファイルをロードするように device を設定できます。コンフィギュレーションファイルは、コマンドラインにコマンドを入力した場合と同様に、メモリにロードされ読み込まれます。そのため、device のコンフィギュレーションは、元のスタートアップコンフィギュレーションと1つまたは2つのダウンロードされたコンフィギュレーションファイルが混在したものになります。

ネットワークとホストのコンフィギュレーションファイル

歴史的な理由から、device が最初にダウンロードするファイルは、ネットワーク コンフィギュレーションファイルと呼ばれます。device が2番目にダウンロードするファイルは、ホスト コンフィギュレーションファイルと呼ばれます。2つのコンフィギュレーションファイルは、ネットワーク上のすべての device が、同一コマンドの多くを使用する場合に使用できます。ネットワーク コンフィギュレーションファイルには、すべての device を設定するために使用される標準コマンドが含まれます。ホスト コンフィギュレーションファイルには、特定の1つのホストに固有のコマンドが含まれます。2つのコンフィギュレーションファイルをロードする場合、ホスト コンフィギュレーションファイルを、もう1つのファイルより優先させる必要があります。ネットワーク コンフィギュレーションファイルとホスト コンフィギュレーションファイルの両方とも、TFTP、RCP、FTPのいずれかを介して到達可能なネットワーク サーバー上にあり、読み取り可能である必要があります。

コンフィギュレーションファイル情報の管理方法

コンフィギュレーションファイル情報の表示

コンフィギュレーションファイルに関する情報を表示するには、このセクションの手順を実行します。

手順の概要

1. **enable**
2. **show boot**
3. **more file-url**
4. **show running-config**
5. **show startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	show boot 例： デバイス# show boot	BOOT 環境変数の内容（設定されている場合）、CONFIG_FILE 環境変数によって指定されているコンフィギュレーション ファイルの名前、および BOOTLDR 環境変数の内容を示します。
ステップ 3	more file-url 例： デバイス# more 10.1.1.1	指定されたファイルの内容を表示します。
ステップ 4	show running-config 例： デバイス# show running-config	実行コンフィギュレーションファイルの内容を表示します（ more system:running-config コマンドのコマンドエイリアスです）。
ステップ 5	show startup-config 例： デバイス# show startup-config	スタートアップ コンフィギュレーション ファイルの内容を表示します。（ more nvram:startup-config コマンドのコマンドエイリアスです）。 クラス A フラッシュ ファイル システム プラットフォーム以外のすべてのプラットフォーム上では、通常、デフォルトの startup-config ファイルは NVRAM に格納されます。 クラス A フラッシュ ファイル システム プラットフォーム上では、CONFIG_FILE 環境変数はデフォルトの startup-config ファイルを指定します。 CONFIG_FILE 変数のデフォルトは NVRAM になります。

コンフィギュレーション ファイルの変更

Cisco IOS ソフトウェアは、1 行につき 1 つのコンフィギュレーション コマンドを受け入れます。コンフィギュレーション コマンドは、必要なだけ入力できます。コンフィギュレーション ファイルには、入力したコマンドを説明するコメントを追加できます。コメントの先頭には、感嘆符 (!) を付けます。コメントは NVRAM にもコンフィギュレーション ファイルのアクティブコピーにも格納されないため、**show running-config** または **more system:running-config EXEC** コマンドでアクティブな設定のリストを表示しても、コメントは表示されません。**show startup-config** または **more nvram:startup-config EXEC** モードコマンドでスタートアップ コンフィギュレーションのリストを表示しても、コメントは表示されません。コメントは、コンフィギュレーションファイルが device にロードされたときにコンフィギュレーションファイルから削除されます。ただし、ファイル転送プロトコル (FTP)、リモートコピープロトコル (RCP)、または Trivial File Transfer Protocol (TFTP) サーバー上に格納されているコンフィギュレーションファイルのコメントのリストは表示できます。CLI を使用してソフトウェアは

設定するときは、ユーザーの入力に従ってソフトウェアによりコマンドが実行されます。CLIを使用してソフトウェアを設定するには、特権EXECモードを開始して次のコマンドを使用します。

手順の概要

1. **enable**
2. **configure terminal**
3. **configuration command**
4. 次のいずれかを実行します。
 - end
 - ^Z
5. **copy system:running-config nvram:startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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	configuration command 例： デバイス(config)# configuration command	必要なコンフィギュレーションコマンドを入力します。Cisco IOS マニュアルセットに、テクノロジー別に編成されたコンフィギュレーションコマンドが説明されています。
ステップ 4	次のいずれかを実行します。 • end • ^Z 例： デバイス(config)# end	コンフィギュレーションセッションを終了し、EXEC モードに戻ります。 (注) Ctrl キーと Z キーを同時に押すと、画面に ^Z と表示されます。
ステップ 5	copy system:running-config nvram:startup-config 例： デバイス# copy system:running-config nvram:startup-config	実行コンフィギュレーション ファイルをスタートアップ コンフィギュレーション ファイルとして保存します。 copy running-config startup-config コマンドエイリアスも使用できますが、このコマンドは精度が高くな

	コマンドまたはアクション	目的
		いため、注意する必要があります。ほとんどのプラットフォーム上では、このコマンドによりコンフィギュレーションはNVRAMに保存されます。クラス A フラッシュ ファイル システムのプラットフォーム上では、この手順によりコンフィギュレーションは CONFIG_FILE 環境変数によって指定された場所に保存されます（デフォルトの CONFIG_FILE 変数では、ファイルの保存先はNVRAMに指定されています）。

例

次の例では、deviceのdeviceプロンプト名を設定しています。感嘆符 (!) で示されたコメント行では、いずれのコマンドも実行されません。hostname コマンドを使用して、device名を device から new_name に変更しています。Ctrl+Z (^Z) キーを押すか、end コマンドを入力すると、コンフィギュレーションモードが終了します。copy system:running-config nvram:startup-config コマンドにより、現在のコンフィギュレーションがスタートアップ コンフィギュレーションに保存されます。

```

デバイス# configure terminal
デバイス(config)# !The following command provides the switch host name.
デバイス(config)# hostname new_name
new_name(config)# end
new_name# copy system:running-config nvram:startup-config
    
```

スタートアップコンフィギュレーションがNVRAMにある場合は、現在の設定情報がコンフィギュレーションコマンドとしてテキスト形式で格納され、デフォルト以外の設定だけが記録されます。破損データから保護するために、メモリはチェックサム算出されます。



(注) 一部の特定のコマンドは、NVRAM に保存されない場合があります。これらのコマンドは、マシンをリブートしたときに再入力する必要があります。これらのコマンドは、マニュアルに記載されています。リブート後にすばやくdeviceを再設定できるように、これらの設定のリストを保管しておくことを推奨します。

デバイスから TFTP サーバーへのコンフィギュレーションファイルのコピー

TFTP ネットワーク サーバー上の設定をコピーするには、以下の手順を実行します。

次の作業

手順の概要

1. **enable**
2. **copy system:running-config tftp: [[[//location]/directory]/filename]**
3. **copy nvram:startup-config tftp: [[[//location]/directory]/filename]**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	copy system:running-config tftp: [[[//location]/directory]/filename] 例： デバイス# copy system:running-config tftp: //server1/topdir/file10	TFTP サーバーへ実行コンフィギュレーション ファイルをコピーします。
ステップ 3	copy nvram:startup-config tftp: [[[//location]/directory]/filename] 例： デバイス# copy nvram:startup-config tftp: //server1/lstdir/file10	TFTP サーバーへスタートアップ コンフィギュレーション ファイルをコピーします。

例

次に、device から TFTP サーバーにコンフィギュレーション ファイルをコピーする例を示します。

```
デバイス# copy system:running-config tftp://172.16.2.155/tokyo-config
Write file tokyo-config on host 172.16.2.155? [confirm] Y
Writing tokyo-config!!! [OK]
```

次の作業

copy コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

デバイスから RCP サーバへのコンフィギュレーションファイルのコピー

deviceからRCPサーバへスタートアップコンフィギュレーションファイルまたは実行コンフィギュレーションファイルをコピーするには、特権 EXEC モードを開始して次のコマンドを使用します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-username *username***
4. **end**
5. 次のいずれかを実行します。
 - **copy system:running-config rcp:** [[[/[*username*@]*location*]/*directory*]/*filename*]
 - **copy nvram:startup-config rcp:** [[[/[*username*@]*location*]/*directory*]/*filename*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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	ip rcmd remote-username <i>username</i> 例： デバイス(config)# ip rcmd remote-username NetAdmin1	(任意) デフォルトのリモートユーザー名を変更します。
ステップ 4	end 例： デバイス(config)# end	(任意) グローバル コンフィギュレーション モードを終了します。
ステップ 5	次のいずれかを実行します。 • copy system:running-config rcp: [[[/[<i>username</i> @] <i>location</i>]/ <i>directory</i>]/ <i>filename</i>]	• device の実行コンフィギュレーションファイルがRCPサーバー上に格納されるように指定します。

	コマンドまたはアクション	目的
	<p>• copy nvram:startup-config rcp: [[[/[username@]location]/directory]/filename]</p> <p>例 :</p> <pre>デバイス# copy system:running-config rcp: //NetAdmin1@example.com/dir-files/file1</pre>	<p>または</p> <ul style="list-style-type: none"> • deviceのスタートアップ コンフィギュレーションファイルが RCP サーバー上に格納されるように指定します。

例

RCP サーバーへの実行コンフィギュレーションファイルの格納

次に、`rtr2-config` という名前の実行コンフィギュレーションファイルを IP アドレス `172.16.101.101` のリモート ホスト上の `netadmin1` ディレクトリにコピーする例を示します。

```
デバイス# copy system:running-config rcp://netadmin1@172.16.101.101/runfile2-config
Write file runfile2-config on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
デバイス#
```

RCP サーバーへのスタートアップコンフィギュレーションファイルの格納

次に、RCP を使用してファイルをコピーすることによって、サーバー上にスタートアップコンフィギュレーションファイルを格納する例を示します。

```
デバイス# configure terminal

デバイス(config)# ip rcmd remote-username netadmin2

デバイス(config)# end

デバイス# copy nvram:startup-config rcp:

Remote host[ ]? 172.16.101.101

Name of configuration file to write [start-config]?
Write file start-config on host 172.16.101.101?[confirm]
![OK]
```

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

デバイスから FTP サーバへのコンフィギュレーションファイルのコピー

deviceからFTPサーバへスタートアップコンフィギュレーションファイルまたは実行コンフィギュレーションファイルをコピーするには、以下の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip ftp username *username***
4. **ip ftp password *password***
5. **end**
6. 次のいずれかを実行します。
 - **copy system:running-config ftp:** [[[/[*username* [:*password*]@]*location*]/*directory*]/*filename*] または
 - **copy nvram:startup-config ftp:** [[[/[*username* [:*password*]@]*location*]/*directory*]/*filename*]

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 : デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。
ステップ 2	configure terminal 例 : デバイス# configure terminal	device上で、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	ip ftp username <i>username</i> 例 : デバイス(config)# ip ftp username NetAdmin1	(任意) デフォルトのリモートユーザー名を指定します。
ステップ 4	ip ftp password <i>password</i> 例 : デバイス(config)# ip ftp password adminpassword	(任意) デフォルトのパスワードを指定します。
ステップ 5	end 例 :	(任意) グローバル コンフィギュレーションモードを終了します。この手順は、デフォルトのリモート

	コマンドまたはアクション	目的
	デバイス(config)# end	トユーザー名またはパスワードを上書きする場合に だけが必要です (ステップ 2 および 3 を参照)。
ステップ 6	<p>次のいずれかを実行します。</p> <ul style="list-style-type: none"> • copy system:running-config ftp: [[[//[username[:password]@]location]/directory]/filename] または • copy nvram:startup-config ftp: [[[//[username[:password]@]location]/directory]/filename] <p>例 :</p> <pre>デバイス# copy system:running-config ftp:</pre>	FTP サーバーの指定された場所へ実行コンフィギュレーションまたはスタートアップコンフィギュレーションファイルをコピーします。

例

FTP サーバーへの実行コンフィギュレーションファイルの格納

次に、runfile-config という名前の実行コンフィギュレーションファイルを実行 IP アドレス 172.16.101.101 のリモートホスト上の netadmin1 ディレクトリにコピーする例を示します。

```
デバイス# copy system:running-config ftp://netadmin1:mypass@172.16.101.101/runfile-config
Write file runfile-config on host 172.16.101.101?[confirm]
Building configuration...[OK]
Connected to 172.16.101.101
デバイス#
```

FTP サーバーへのスタートアップコンフィギュレーションファイルの格納

次に、FTPを使用してファイルをコピーすることによって、サーバー上にスタートアップコンフィギュレーションファイルを格納する例を示します。

```
デバイス# configure terminal

デバイス(config)# ip ftp username netadmin2

デバイス(config)# ip ftp password mypass

デバイス(config)# end

デバイス# copy nvram:startup-config ftp:

Remote host[]? 172.16.101.101

Name of configuration file to write [start-config]?
Write file start-config on host 172.16.101.101?[confirm]
![OK]
```

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

TFTP サーバからデバイスへのコンフィギュレーション ファイルのコピー

TFTP サーバからdeviceへコンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順の概要

1. **enable**
2. **copy tftp: [[[//location]/directory]/filename] system:running-config**
3. **copy tftp: [[[//location]/directory]/filename] nvram:startup-config**
4. **copy tftp: [[[//location]/directory]/filename] flash-[n]:/directory/startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	copy tftp: [[[//location]/directory]/filename] system:running-config 例： デバイス# copy tftp://server1/dir10/datasource system:running-config	TFTP サーバから実行コンフィギュレーションへコンフィギュレーションファイルをコピーします。
ステップ 3	copy tftp: [[[//location]/directory]/filename] nvram:startup-config 例： デバイス# copy tftp://server1/dir10/datasource nvram:startup-config	TFTP サーバからスタートアップコンフィギュレーションへコンフィギュレーションファイルをコピーします。
ステップ 4	copy tftp: [[[//location]/directory]/filename] flash-[n]:/directory/startup-config 例：	TFTP サーバからスタートアップコンフィギュレーションへコンフィギュレーションファイルをコピーします。

次の作業

	コマンドまたはアクション	目的
	デバイス# copy tftp://server1/dir10/datasource flash:startup-config	

例

次に、IP アドレス 172.16.2.155 にある、**tokyo-config** という名前のファイルからソフトウェアを設定する例を示します。

```
デバイス# copy tftp://172.16.2.155/tokyo-config system:running-config
Configure using tokyo-config from 172.16.2.155? [confirm] Y
Booting tokyo-config from 172.16.2.155:!!! [OK - 874/16000 bytes]
```

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

rcpサーバからデバイスへのコンフィギュレーションファイルのコピー

rcp サーバから実行コンフィギュレーションまたはスタートアップ コンフィギュレーションへコンフィギュレーションファイルをコピーするには、以下の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-username *username***
4. **end**
5. 次のいずれかを実行します。
 - **copy rcp:[[[//[*username*@]*location*]/*directory*]/*filename*]system:running-config**
 - **copy rcp:[[[//[*username*@]*location*]/*directory*]/*filename*]nvram:startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	configure terminal 例： デバイス# <code>configure terminal</code>	(任意) 端末からコンフィギュレーションモードを開始します。この手順は、デフォルトのリモートユーザー名を上書きする場合にだけ必要です (ステップ 3 を参照)。
ステップ 3	ip rcmd remote-username <i>username</i> 例： デバイス(config)# <code>ip rcmd remote-username NetAdmin1</code>	(任意) リモートユーザー名を指定します。
ステップ 4	end 例： デバイス(config)# <code>end</code>	(任意) グローバルコンフィギュレーションモードを終了します。この手順は、デフォルトのリモートユーザー名またはパスワードを上書きする場合にだけ必要です (ステップ 2 を参照)。
ステップ 5	次のいずれかを実行します。 <ul style="list-style-type: none"> • copy <code>rcp[[//[username@]location]directory]filename]system:running-config</code> • copy <code>rcp[[//[username@]location]directory]filename]nvram:startup-config</code> 例： デバイス# <code>copy</code> <code>rcp://[user1@example.com/dir10/fileone]</code> <code>nvram:startup-config</code>	rcp サーバーから実行コンフィギュレーションまたはスタートアップコンフィギュレーションへコンフィギュレーションファイルをコピーします。

例

rcp の Running-Config のコピー

次に、`host1-config` という名前のコンフィギュレーションファイルを、IP アドレスが `172.16.101.101` のリモートサーバ上の `netadmin1` ディレクトリからコピーし、`device` でコマンドをロードし実行する例を示します。

```

デバイス# copy rcp://netadmin1@172.16.101.101/host1-config system:running-config
Configure using host1-config from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-config:![OK]
デバイス#
%SYS-5-CONFIG: Configured from host1-config by rcp from 172.16.101.101
    
```

rcp の Startup-Config のコピー

次に、リモートユーザー名 `netadmin1` を指定する例を示します。次に `host2-config` という名前のコンフィギュレーションファイルを、IP アドレスが `172.16.101.101` のリモートサーバ上の `netadmin1` ディレクトリからスタートアップコンフィギュレーションへコピーします。

```

デバイス# configure terminal
デバイス(config)# ip rcmd remote-username netadmin1
デバイス(config)# end
デバイス# copy rcp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[rtr2-config]? host2-config
Configure using host2-config from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-config:![OK]
[OK]
デバイス#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by rcp from 172.16.101.101
    
```

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

FTP サーバからデバイスへのコンフィギュレーションファイルのコピー

FTP サーバから実行コンフィギュレーションまたはスタートアップコンフィギュレーションへコンフィギュレーションファイルをコピーするには、以下の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip ftp username** *username*
4. **ip ftp password** *password*
5. **end**
6. 次のいずれかを実行します。

- **copy ftp:** [[[//[*username*[:*password*]@]*location*]/*directory*]/*filename*]**system:running-config**

- **copy ftp:** [[[//[*username*[:*password*]@]*location*]/*directory*]/*filename*]**nvram:startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 : デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。

	コマンドまたはアクション	目的
ステップ 2	configure terminal 例： デバイス# <code>configure terminal</code>	(任意) グローバル コンフィギュレーション モードを開始できます。この手順は、デフォルトのリモートユーザー名またはパスワードを上書きする場合にだけ必要です (ステップ 3 および 4 を参照)。
ステップ 3	ip ftp username <i>username</i> 例： デバイス(config)# <code>ip ftp username NetAdmin1</code>	(任意) デフォルトのリモートユーザー名を指定します。
ステップ 4	ip ftp password <i>password</i> 例： デバイス(config)# <code>ip ftp password adminpassword</code>	(任意) デフォルトのパスワードを指定します。
ステップ 5	end 例： デバイス(config)# <code>end</code>	(任意) グローバル コンフィギュレーション モードを終了します。この手順は、デフォルトのリモートユーザー名またはパスワードを上書きする場合にだけ必要です (ステップ 3 および 4 を参照)。
ステップ 6	次のいずれかを実行します。 <ul style="list-style-type: none">• copy ftp: [[[//[<i>username</i>[:<i>password</i>]@]<i>location</i>]/<i>directory</i>]/<i>filename</i>]system:running-config• copy ftp: [[[//[<i>username</i>[:<i>password</i>]@]<i>location</i>]/<i>directory</i>]/<i>filename</i>]nvr:startup-config 例： デバイス# <code>copy ftp:nvram:startup-config</code>	FTPを使用して、ネットワークサーバーから実行メモリまたはスタートアップコンフィギュレーションへコンフィギュレーションファイルをコピーします。

例

FTP の Running-Config のコピー

次に、`host1-config` という名前のコンフィギュレーションファイルを、IP アドレスが 172.16.101.101 のリモートサーバ上の `netadmin1` ディレクトリからコピーし、`device` でコマンドをロードし実行する例を示します。

```

デバイス# copy ftp://netadmin1:mypass@172.16.101.101/host1-config system:running-config
Configure using host1-config from 172.16.101.101? [confirm]
Connected to 172.16.101.101
Loading 1112 byte file host1-config:![OK]
デバイス#
%SYS-5-CONFIG: Configured from host1-config by ftp from 172.16.101.101
    
```

FTP の Startup-Config のコピー

次に、リモートユーザー名 `netadmin1` を指定する例を示します。次に `host2-config` という名前のコンフィギュレーション ファイルを、IP アドレスが `172.16.101.101` のリモート サーバー上の `netadmin1` ディレクトリからスタートアップ コンフィギュレーションへコピーします。

```

デバイス# configure terminal
デバイス(config)# ip ftp username netadmin1
デバイス(config)# ip ftp password mypass
デバイス(config)# end
デバイス# copy ftp: nvram:startup-config
Address of remote host [255.255.255.255]? 172.16.101.101
Name of configuration file[host1-config]? host2-config
Configure using host2-config from 172.16.101.101?[confirm]
Connected to 172.16.101.101
Loading 1112 byte file host2-config:[OK]
[OK]
デバイス#
%SYS-5-CONFIG_NV:Non-volatile store configured from host2-config by ftp from 172.16.101.101

```

次の作業

`copy EXEC` コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、`copy` コマンドで入力した情報量および `file prompt` グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

NVRAM より大きいコンフィギュレーション ファイルの保守

NVRAM のサイズを超えるコンフィギュレーション ファイルを保守するには、以降のセクションで説明するタスクを実行します。

コンフィギュレーション ファイルの圧縮

コンフィギュレーション ファイルを圧縮するには、このセクションの手順を実行してください。

手順の概要

1. `enable`
2. `configure terminal`
3. `service compress-config`
4. `end`
5. 次のいずれかを実行します。
 - 新しいコンフィギュレーションをコピーするには、FTP、RCP、TFTP を使用します。
 - `configure terminal`
6. `copy system:running-config nvram:startup-config`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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	service compress-config 例： デバイス(config)# service compress-config	コンフィギュレーションファイルを圧縮することを指定します。
ステップ 4	end 例： デバイス(config)# end	グローバル コンフィギュレーション モードを終了します。
ステップ 5	次のいずれかを実行します。 • 新しいコンフィギュレーションをコピーするには、FTP、RCP、TFTP を使用します。 • configure terminal 例： デバイス# configure terminal	新しいコンフィギュレーションを入力します。 • NVRAM のサイズの 3 倍以上のコンフィギュレーションをロードしようとする、次のエラーメッセージが表示されます。 「[buffer overflow - file-size /buffer-size bytes]。」
ステップ 6	copy system:running-config nvram:startup-config 例： デバイス(config)# copy system:running-config nvram:startup-config	実行コンフィギュレーションの変更が終わったら、新しいコンフィギュレーションを保存します。

例

次に、129 KB のコンフィギュレーションファイルを 11 KB に圧縮する例を示します。

デバイス# **configure terminal**

デバイス(config)# **service compress-config**

```

デバイス (config) # end

デバイス # copy tftp://172.16.2.15/tokyo-config system:running-config

Configure using tokyo-config from 172.16.2.155? [confirm] y

Booting tokyo-config from 172.16.2.155:!!! [OK - 874/16000 bytes]
デバイス # copy system:running-config nvram:startup-config

Building configuration...
Compressing configuration from 129648 bytes to 11077 bytes
[OK]
    
```

コンフィギュレーションのクラス A フラッシュ ファイル システム上のフラッシュ メモリへの格納

スタートアップ コンフィギュレーションをフラッシュ メモリに格納するには、このセクションの手順を実行してください。

手順の概要

1. **enable**
2. **copy nvram:startup-config flash-filesystem:filename**
3. **configure terminal**
4. **boot config flash-filesystem: filename**
5. **end**
6. 次のいずれかを実行します。
 - 新しいコンフィギュレーションをコピーするには、FTP、RCP、TFTPを使用します。NVRAM サイズの 3 倍を超える大きさのコンフィギュレーションをロードしようとすると、次のエラー メッセージが表示されます。「[buffer overflow - file-size /buffer-size bytes]」
 - **configure terminal**
7. **copy system:running-config nvram:startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 : デバイス > enable	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。
ステップ 2	copy nvram:startup-config flash-filesystem:filename 例 : デバイス # copy nvram:startup-config usbflash0:switch-config	新しい場所に現在のスタートアップ コンフィギュレーションをコピーして、コンフィギュレーション ファイルを作成します。

	コマンドまたはアクション	目的
ステップ 3	configure terminal 例： デバイス# <code>configure terminal</code>	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 4	boot config flash-filesystem: filename 例： デバイス(config)# <code>boot config usbflash0:switch-config</code>	CONFIG_FILE 環境変数を設定することにより、フラッシュメモリにスタートアップコンフィギュレーション ファイルを格納することを指定します。
ステップ 5	end 例： デバイス(config)# <code>end</code>	グローバル コンフィギュレーション モードを終了します。
ステップ 6	次のいずれかを実行します。 <ul style="list-style-type: none"> 新しいコンフィギュレーションをコピーするには、FTP、RCP、TFTP を使用します。NVRAM サイズの 3 倍を超える大きさのコンフィギュレーションをロードしようとする、次のエラーメッセージが表示されます。「[buffer overflow - file-size /buffer-size bytes]」 configure terminal 例： デバイス# <code>configure terminal</code>	新しいコンフィギュレーションを入力します。
ステップ 7	copy system:running-config nvram:startup-config 例： デバイス(config)# <code>copy system:running-config nvram:startup-config</code>	実行コンフィギュレーションの変更が終わったら、新しいコンフィギュレーションを保存します。

例

以下に、usbflash0: に格納したコンフィギュレーションの例を示します。

デバイス# `copy nvram:startup-config usbflash0:switch-config`

デバイス# `configure terminal`

デバイス(config)# `boot config usbflash0:switch-config`

```
デバイス(config)# end
```

```
デバイス# copy system:running-config nvram:startup-config
```

ネットワークからのコンフィギュレーションコマンドのロード

ネットワークサーバーを使用して、大きなコンフィギュレーションを保存するには、このセクションの手順を実行します。

手順の概要

1. **enable**
2. **copy system:running-config {ftp: | rcp: | tftp:}**
3. **configure terminal**
4. **boot network {ftp:[[//[username [:password]@]location]/directory]/filename] | rcp:[[//[username@]location]/directory]/filename] | tftp:[[//[location]/directory]/filename]}**
5. **service config**
6. **end**
7. **copy system:running-config nvram:startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none">パスワードを入力します（要求された場合）。
ステップ 2	copy system:running-config {ftp: rcp: tftp:} 例： デバイス# copy system:running-config ftp:	実行コンフィギュレーションをFTP、RCP、TFTPのいずれかのサーバーに保存します。
ステップ 3	configure terminal 例： デバイス# 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 4	boot network {ftp:[[//[username [:password]@]location]/directory]/filename] rcp:[[//[username@]location]/directory]/filename] tftp:[[//[location]/directory]/filename]} 例： デバイス(config)# boot network ftp://user1:guessme@example.com/dir10/file1	起動時にスタートアップコンフィギュレーションファイルをネットワークサーバーからロードすることを指定します。

	コマンドまたはアクション	目的
ステップ 5	service config 例： デバイス(config)# service config	システムの起動時にコンフィギュレーションファイルをダウンロードするようにスイッチをイネーブにします。
ステップ 6	end 例： デバイス(config)# end	グローバル コンフィギュレーション モードを終了します。
ステップ 7	copy system:running-config nvram:startup-config 例： デバイス# copy system:running-config nvram:startup-config	設定を保存します。

フラッシュメモリからスタートアップまたは実行コンフィギュレーションへのコンフィギュレーションファイルのコピー

フラッシュメモリから現在の NVRAM にあるスタートアップ コンフィギュレーションまたは実行コンフィギュレーションへコンフィギュレーションファイルを直接コピーするには、ステップ 2 のいずれかのコマンドを入力します。

手順の概要

1. **enable**
2. 次のいずれかを実行します。
 - **copy filesystem: [partition-number:][filename] nvram:startup-config**
 - **copy filesystem: [partition-number:][filename] system:running-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • パスワードを入力します（要求された場合）。
ステップ 2	次のいずれかを実行します。 <ul style="list-style-type: none"> • copy filesystem: [partition-number:][filename] nvram:startup-config 	<ul style="list-style-type: none"> • NVRAM にコンフィギュレーションファイルを直接ロードする、または

	コマンドまたはアクション	目的
	<p>• copy filesystem: <i>[partition-number:]</i><i>[filename]</i> system:running-config</p> <p>例 :</p> <pre>デバイス# copy usbflash0:4:ios-upgrade-1 nvram:startup-config</pre>	<ul style="list-style-type: none"> 現在の実行コンフィギュレーションにコンフィギュレーション ファイルをコピーします。

例

次に、usbflash0 にあるフラッシュメモリ PC カードのパーティション 4 から device のスタートアップコンフィギュレーションへ ios-upgrade-1 という名前のファイルをコピーする例を示します。

```
デバイス# copy usbflash0:4:ios-upgrade-1 nvram:startup-config
Copy 'ios-upgrade-1' from flash device as 'startup-config' ? [yes/no] yes
[OK]
```

フラッシュメモリ ファイル システム間でのコンフィギュレーション ファイルのコピー

複数のフラッシュメモリ ファイルシステムを備えたプラットフォーム上では、内部フラッシュメモリなどのフラッシュメモリ ファイルシステムから他のフラッシュメモリ ファイルシステムへファイルをコピーできます。異なるフラッシュメモリ ファイルシステムへファイルをコピーすると、使用中のコンフィギュレーションのバックアップコピーを作成し、他の device にコンフィギュレーションを複製できます。フラッシュメモリ ファイルシステム間でコンフィギュレーション ファイルをコピーするには、EXEC モードで次のコマンドを使用します。

手順の概要

- enable**
- show source-filesystem:**
- copy source-filesystem:** *[partition-number:]**[filename]* *dest-filesystem:**[partition-number:]**[filename]*

手順の詳細

	コマンドまたはアクション	目的
<p>ステップ 1</p>	<p>enable</p> <p>例 :</p> <pre>デバイス> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> パスワードを入力します (要求された場合)。

	コマンドまたはアクション	目的
ステップ 2	<p>show <i>source-filesystem</i>:</p> <p>例 :</p> <p>デバイス# show flash:</p>	<p>フラッシュメモリのレイアウトと内容を表示して、ファイル名を確認します。</p>
ステップ 3	<p>copy <i>source-filesystem</i>: [<i>partition-number</i>:][<i>filename</i>]</p> <p><i>dest-filesystem</i>: [<i>partition-number</i>:][<i>filename</i>]</p> <p>例 :</p> <p>デバイス# copy flash: usbflash0:</p>	<p>フラッシュメモリデバイス間でコンフィギュレーションファイルをコピーします。</p> <ul style="list-style-type: none"> コピー元デバイスとコピー先デバイスは同じにはできません。たとえば、copy usbflash0: usbflash0: コマンドが無効です。

例

次に、内部フラッシュメモリのパーティション 1 から device 上の **usbflash0** のパーティション 1 へ **running-config** という名前のファイルをコピーする例を示します。この例では、コピー元のパーティションが指定されていないため、device からパーティション番号を要求されます。

デバイス# **copy flash: usbflash0:**

```
System flash
Partition  Size  Used    Free      Bank-Size  State      Copy Mode
   1         4096K   3070K   1025K     4096K      Read/Write Direct
   2        16384K  1671K   14712K    8192K      Read/Write Direct
[Type ?<no> for partition directory; ? for full directory; q to abort]
Which partition? [default = 1]
System flash directory, partition 1:
File Length  Name/status
   1  3142748  dirt/network/mars-test/c3600-j-mz.latest
   2    850   running-config
[3143728 bytes used, 1050576 available, 4194304 total]
usbflash0 flash directory:
File Length  Name/status
   1  1711088  dirt/gate/c3600-i-mz
   2    850   running-config
[1712068 bytes used, 2482236 available, 4194304 total]
Source file name? running-config

Destination file name [running-config]?
Verifying checksum for 'running-config' (file # 2)... OK
Erase flash device before writing? [confirm]
Flash contains files. Are you sure you want to erase? [confirm]
Copy 'running-config' from flash: device
  as 'running-config' into usbflash0: device WITH erase? [yes/no] yes

Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
...erased!
[OK - 850/4194304 bytes]
Flash device copy took 00:00:30 [hh:mm:ss]
Verifying checksum... OK (0x16)
```

FTP サーバーからフラッシュメモリ デバイスへのコンフィギュレーションファイルのコピー

FTP サーバーからフラッシュメモリ デバイスへコンフィギュレーションファイルをコピーするには、以下の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip ftp username *username***
4. **ip ftp password *password***
5. **end**
6. **copy ftp: [[//location]/directory]/bundle_name flash:**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# configure terminal	（任意）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。この手順は、デフォルトのリモートユーザー名またはパスワードを上書きする場合にだけ必要です（ステップ 3 および 4 を参照）。
ステップ 3	ip ftp username <i>username</i> 例： デバイス(config)# ip ftp username Admin01	（任意）リモートユーザー名を指定します。
ステップ 4	ip ftp password <i>password</i> 例： デバイス(config)# ip ftp password adminpassword	（任意）リモートパスワードを指定します。
ステップ 5	end 例： デバイス(config)# end	（任意）コンフィギュレーションモードを終了します。このステップが必要になるのは、デフォルトのリモートユーザー名を上書きする場合のみです（ステップ 3 および 4 を参照）。

	コマンドまたはアクション	目的
ステップ 6	<p>copy ftp: [[[location]/directory] /bundle_name flash:</p> <p>例 :</p> <pre>デバイス>copy ftp:/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin flash:</pre>	FTP を使用してネットワーク サーバーからフラッシュメモリ デバイスへコンフィギュレーション ファイルをコピーします。

次の作業

copy EXEC コマンドを発行した後、追加情報またはアクションの確認を求めるプロンプトが表示される場合があります。表示されるプロンプトは、**copy** コマンドで入力した情報量および **file prompt** グローバル コンフィギュレーション コマンドの現在の設定によって異なります。

RCP サーバーからフラッシュ メモリ デバイスへのコンフィギュレーション ファイルのコピー

RCP サーバーからフラッシュ メモリ デバイスへコンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **ip rcmd remote-username username**
4. **end**
5. **copy rcp:** [[[username@]location] /directory] /bundle_name] flash:

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>デバイス> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します（要求された場合）。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <pre>デバイス# configure terminal</pre>	(任意) 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。この手順は、デフォルトのリモートユーザー名またはパスワードを上書きする場合にのみ必要です（ステップ 3 を参照）。
ステップ 3	<p>ip rcmd remote-username username</p> <p>例 :</p>	(任意) リモート ユーザー名を指定します。

TFTP サーバーからフラッシュ メモリ デバイスへのコンフィギュレーション ファイルのコピー

	コマンドまたはアクション	目的
	デバイス(config)# ip rcmd remote-username Admin01	
ステップ 4	end 例： デバイス(config)# end	(任意) コンフィギュレーションモードを終了します。この手順は、デフォルトのリモートユーザー名またはパスワードを上書きする場合にのみ必要です (ステップ 3 を参照)。
ステップ 5	copy rcp: [[[/username@]location]/directory] /bundle_name flash: 例： デバイス# copy rcp://netadmin@172.16.101.101/bundle1 flash:	RCP を使用してネットワーク サーバーからフラッシュ メモリ デバイスへコンフィギュレーション ファイルをコピーします。追加情報または確認を要求する device からのプロンプトに対し応答します。このプロンプトは、 copy コマンドで入力した情報量および file prompt コマンドの現在の設定によって異なります。

TFTP サーバーからフラッシュ メモリ デバイスへのコンフィギュレーション ファイルのコピー

TFTP サーバーからフラッシュ メモリ デバイスへコンフィギュレーション ファイルをコピーするには、以下の手順を実行します。

手順の概要

1. **enable**
2. **copy tftp: [[[/location]/directory]/bundle_name flash:**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します (要求された場合)。
ステップ 2	copy tftp: [[[/location]/directory]/bundle_name flash: 例： デバイス# copy tftp:/cat3k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin flash:	TFTP サーバーからフラッシュ メモリ デバイスへ ファイルをコピーします。追加情報または確認を要求する device からのプロンプトに対し応答します。このプロンプトは、 copy コマンドで入力した情報量および file prompt コマンドの現在の設定によって異なります。

例

次に、TFTP サーバーから `usbflash0` に挿入されているフラッシュ メモリ カードへ、`switch-config` という名前のコンフィギュレーションファイルをコピーする例を示します。コピーされたファイルの名前は `new-config` に変更されます。

```
デバイス#
copy tftp:switch-config usbflash0:new-config
```

スタートアップコンフィギュレーションファイルでのコンフィギュレーションコマンドの再実行

スタートアップコンフィギュレーションファイルのコマンドを再実行するには、このセクションの手順を実行します。

手順の概要

1. **enable**
2. **configure memory**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure memory 例： デバイス# configure memory	スタートアップ コンフィギュレーション ファイルでコンフィギュレーション コマンドを再実行します。

スタートアップコンフィギュレーションのクリア

スタートアップコンフィギュレーションから設定情報を消去できます。`device`をスタートアップコンフィギュレーションなしで再起動した場合は、`device`を最初から設定できるように、`device`は、Setup コマンドファシリティに移行します。スタートアップコンフィギュレーションの内容をクリアするには、次の手順を実行します。

手順の概要

1. **enable**

2. erase nvram

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <p>デバイス> enable</p>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> パスワードを入力します (要求された場合)。
ステップ 2	<p>erase nvram</p> <p>例 :</p> <p>デバイス# erase nvram</p>	<p>スタートアップコンフィギュレーションの内容をクリアします。</p> <p>(注) クラス A フラッシュファイルシステムのプラットフォーム以外のすべてのプラットフォームでは、このコマンドにより NVRAM が消去されます。スタートアップコンフィギュレーションファイルは、いったん削除すると復元できません。クラス A フラッシュファイルシステムのプラットフォーム上では、erase startup-configEXEC コマンドを使用すると、CONFIG_FILE 環境変数により指定されたコンフィギュレーションが、deviceにより削除されます。この変数が NVRAM を指定している場合は、deviceにより NVRAM が消去されます。CONFIG_FILE 環境変数がフラッシュメモリデバイスとコンフィギュレーションファイル名を指定している場合は、deviceによりコンフィギュレーションファイルが削除されます。つまり、そのコンフィギュレーションファイルは deviceにより消去されるのではなく、「削除済み」としてマークされます。この機能では、削除されたファイルを回復できません。</p>

指定されたコンフィギュレーションファイルの削除

特定のフラッシュデバイスの指定された設定を削除するには、このセクションの手順を実行します。

手順の概要

1. **enable**
2. **delete flash-filesystem:filename**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>デバイス> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>delete flash-filesystem:filename</p> <p>例 :</p> <pre>デバイス# delete usbflash0:myconfig</pre>	<p>特定のフラッシュデバイス上の指定されたコンフィギュレーション ファイルを削除します。</p> <p>(注) クラス A および B フラッシュファイルシステムでは、フラッシュメモリ内の特定のファイルを削除すると、そのファイルは削除済みとしてシステムによりマークされます。これにより、undelete EXEC コマンドを使用して、削除したファイルを後で回復できるようになります。消去されたファイルは回復できません。コンフィギュレーション ファイルを完全に消去するには、squeeze EXEC コマンドを使用します。クラス C フラッシュファイルシステムでは、削除されたファイルは回復できません。CONFIG_FILE 環境変数で指定されたコンフィギュレーション ファイルを消去または削除しようとした場合、システムにより削除の確認を求めるプロンプトが表示されます。</p>

クラス A フラッシュ ファイル システムでの CONFIG_FILE 環境変数の指定

クラス A フラッシュ ファイル システムでは、CONFIG_FILE 環境変数で指定されたスタートアップコンフィギュレーションファイルを読み取るように Cisco IOS ソフトウェアを設定できます。CONFIG_FILE 変数のデフォルトは NVRAM になります。CONFIG_FILE 環境変数を変更するには、このセクションの手順を実行してください。

手順の概要

1. **enable**

2. **copy** *[flash-url | ftp-url | rcp-url | tftp-url]* **system:running-config | nvram:startup-config**
dest-flash-url
3. **configure terminal**
4. **boot config** *dest-flash-url*
5. **end**
6. **copy system:running-config nvram:startup-config**
7. **show boot**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	copy <i>[flash-url ftp-url rcp-url tftp-url system:running-config nvram:startup-config]</i> <i>dest-flash-url</i> 例： デバイス# copy system:running-config nvram:startup-config	フラッシュ ファイル システムにコンフィギュレーションファイルのコピーします。再起動時には、ここからdeviceにファイルがロードされます。
ステップ 3	configure terminal 例： デバイス# 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 4	boot config <i>dest-flash-url</i> 例： デバイス(config)# boot config 172.16.1.1	CONFIG_FILE 環境変数を設定します。この手順により、実行時の CONFIG_FILE 環境変数が変更されます。
ステップ 5	end 例： デバイス(config)# end	グローバル コンフィギュレーション モードを終了します。
ステップ 6	copy system:running-config nvram:startup-config 例： デバイス# copy system:running-config nvram:startup-config	スタートアップ コンフィギュレーションにステップ 3 で実行されたコンフィギュレーションを保存します。

	コマンドまたはアクション	目的
ステップ 7	show boot 例 : デバイス# show boot	(任意) CONFIG_FILE 環境変数の内容を確認できます。

例

次の例は、実行コンフィギュレーションファイルをdeviceにコピーします。その後、システムが再起動されるとこのコンフィギュレーションがスタートアップコンフィギュレーションとして使用されます。

```

デバイス# copy system:running-config usbflash0:config2
デバイス# configure terminal
デバイス(config)# boot config usbflash0:config2
デバイス(config)# end
デバイス# copy system:running-config nvram:startup-config
[ok]
デバイス# show boot
BOOT variable = usbflash0:rsp-boot-m
CONFIG_FILE variable = nvram:
Current CONFIG_FILE variable = usbflash0:config2
Configuration register is 0x010F
    
```

次の作業

スタートアップコンフィギュレーションファイルの場所を指定すると、**nvram:startup-config** コマンドは、スタートアップコンフィギュレーションファイルの新しい場所のエイリアスとなります。**more nvram:startup-config EXEC** コマンドにより、スタートアップコンフィギュレーションの場所に関係なく、スタートアップコンフィギュレーションが表示されます。**erase nvram:startup-config EXEC** コマンドにより、NVRAMの内容が消去され、CONFIG_FILE 環境変数で指定されたファイルが削除されます。

copy system:running-config nvram:startup-config コマンドを使用してコンフィギュレーションを保存した場合、deviceによりコンフィギュレーションファイルの完全バージョンは CONFIG_FILE 環境変数により指定された場所に保存され、抽出バージョンはNVRAMに保存されます。抽出バージョンとは、アクセスリスト情報を含まないバージョンです。NVRAMに完全バージョンのコンフィギュレーションファイルが含まれている場合は、deviceは完全バージョンを抽出バージョンで上書きすることを確認するプロンプトを表示します。NVRAMに抽出コンフィギュレーションが含まれている場合は、deviceは確認のプロンプトを表示しないでNVRAMにある既存の抽出バージョンのコンフィギュレーションファイルを上書きする処理を進めます。



- (注) フラッシュデバイスにあるファイルを CONFIG_FILE 環境変数として指定した場合、**copy system:running-config nvram:startup-config** コマンドでコンフィギュレーションファイルを保存するたびに、古いコンフィギュレーションファイルは「削除済み」とマークされ、新しいコンフィギュレーションファイルがそのデバイスに保存されます。それでも古いコンフィギュレーションファイルがメモリを使用するため、最終的にフラッシュメモリは一杯になります。**squeeze EXEC** コマンドを使用して古いコンフィギュレーションファイルを完全に削除し、領域を解放してください。

コンフィギュレーションファイルをダウンロードするデバイスの設定

ネットワーク コンフィギュレーションおよびホスト コンフィギュレーション ファイル名の順序付きリストを指定できます。Cisco IOS XE ソフトウェアは、適切なネットワークまたはホスト コンフィギュレーション ファイルをロードするまで、このリストをスキャンします。

システムの起動時にコンフィギュレーションファイルをダウンロードするように device を設定するには、次のセクションで説明するタスクを少なくとも 1 つ実行します。

- [ネットワーク コンフィギュレーション ファイルをダウンロードするデバイスの設定](#)
- [ホスト コンフィギュレーション ファイルをダウンロードするデバイスの設定](#)

起動中にコンフィギュレーションファイルをロードできなかった場合、要求されたファイルがホストから提供されるまで、device は 10 分ごと（デフォルト設定）に再試行します。試行が失敗するごとに、device により以下のメッセージがコンソール端末に表示されます。

```
Booting host-config... [timed out]
```

スタートアップ コンフィギュレーション ファイルになんらかの問題がある場合、またはコンフィギュレーション レジスタが NVRAM を無視するように設定されている場合は、device は Setup コマンドファシリティに移行します。

ネットワーク コンフィギュレーション ファイルをダウンロードするデバイスの設定

起動時にサーバーからネットワーク コンフィギュレーション ファイルをダウンロードするように Cisco IOS ソフトウェアを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **boot network** {ftp:[[/[/[username [:password]@]location]/directory]/filename] | rep:[[/[/[username@]location]/directory]/filename] | tftp:[[/[/[location]/directory]/filename]}
4. **service config**
5. **end**
6. **copy system:running-config nvram:startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <pre>デバイス> enable</pre>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <pre>デバイス# configure terminal</pre>	<p>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。</p>
ステップ 3	<p>boot network {ftp:[[[[/username [:password]@]location]/directory]/filename] rcp:[[[[/username@]location]/directory]/filename] tftp:[[[[/location]/directory]/filename]}</p> <p>例 :</p> <pre>デバイス(config)# boot network tftp:hostfile1</pre>	<p>起動時にダウンロードするネットワーク コンフィギュレーションファイルおよび使用されるプロトコル (TFTP、RCP、または FTP) を指定します。</p> <ul style="list-style-type: none"> ネットワーク コンフィギュレーション ファイル名を指定しない場合、Cisco IOS ソフトウェアはデフォルトのファイル名の <code>network-config</code> を使用します。アドレスを省略した場合、<code>device</code> はブロードキャストアドレスを使用します。 複数のネットワーク コンフィギュレーション ファイルを指定できます。ソフトウェアは、ネットワーク コンフィギュレーション ファイルをロードできるまで、入力された順に試行します。この手順は、異なる設定情報を持つ、ネットワーク サーバー上にロードされるファイルを複数保持する場合に役立ちます。
ステップ 4	<p>service config</p> <p>例 :</p> <pre>デバイス(config)# service config</pre>	<p>再起動時にネットワーク ファイルを自動的にロードするようにシステムをイネーブルにします。</p>
ステップ 5	<p>end</p> <p>例 :</p> <pre>デバイス(config)# end</pre>	<p>グローバル コンフィギュレーション モードを終了します。</p>
ステップ 6	<p>copy system:running-config nvram:startup-config</p> <p>例 :</p> <pre>デバイス# copy system:running-config nvram:startup-config</pre>	<p>実行コンフィギュレーションをスタートアップコンフィギュレーション ファイルに保存します。</p>

ホストコンフィギュレーションファイルをダウンロードするデバイスの設定

起動時にサーバーからホストコンフィギュレーションファイルをダウンロードするように Cisco IOS ソフトウェアを設定するには、次の手順を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **boot host {ftp:[[/[/username [:password]@]location]/directory]/filename] | rcp:[[/[/username@]location]/directory]/filename] | tftp:[[/[/location]/directory]/filename] }**
4. **service config**
5. **end**
6. **copy system:running-config nvram:startup-config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <p>デバイス> enable</p>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure terminal</p> <p>例 :</p> <p>デバイス# configure terminal</p>	<p>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。</p>
ステップ 3	<p>boot host {ftp:[[/[/username [:password]@]location]/directory]/filename] rcp:[[/[/username@]location]/directory]/filename] tftp:[[/[/location]/directory]/filename] }</p> <p>例 :</p> <p>デバイス(config)# boot host tftp:hostfile1</p>	<p>起動時にダウンロードするホストコンフィギュレーションファイルおよび使用されるプロトコル (FTP、RCP、または TFTP) を指定します。</p> <ul style="list-style-type: none"> • ホストコンフィギュレーションファイルの名前を指定しない場合、deviceは、それ自身の名前を使用してホストコンフィギュレーションファイル名を形成します。このとき、その名前はすべて小文字に変換され、すべてのドメイン情報は削除され、「-config」が追加されます。ホスト名の情報を利用できない場合は、ソフトウェアはデフォルトのホストコンフィギュレーションファイル名の device-config を使用します。アドレスを省略した場合、deviceはブロードキャストアドレスを使用します。 • 複数のホストコンフィギュレーションファイルを指定できます。Cisco IOS ソフトウェアは、ホストコンフィギュレーションファイルをロー

	コマンドまたはアクション	目的
		ドできるまで、入力された順に試行します。この手順は、異なる設定情報を持つ、ネットワークサーバー上にロードされるファイルを複数保持する場合に役立ちます。
ステップ 4	service config 例： デバイス(config)# service config	再起動時にホストファイルを自動的にロードするようにシステムをイネーブルにします。
ステップ 5	end 例： デバイス(config)# end	グローバル コンフィギュレーション モードを終了します。
ステップ 6	copy system:running-config nvram:startup-config 例： デバイス# copy system:running-config nvram:startup-config	実行コンフィギュレーションをスタートアップコンフィギュレーションファイルに保存します。

例

次に、hostfile1 という名前のホストコンフィギュレーションファイルおよびnetworkfile1 という名前のネットワークコンフィギュレーションファイルをダウンロードするようにdeviceを設定する例を示します。deviceは TFTP およびブロードキャストアドレスを使用してファイルを取得します。

```

デバイス# configure terminal
デバイス(config)# boot host tftp:hostfile1
デバイス(config)# boot network tftp:networkfile1
デバイス(config)# service config
デバイス(config)# end
デバイス# copy system:running-config nvram:startup-config
    
```

コンフィギュレーションファイルの管理の機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	コンフィギュレーションファイルの管理	コンフィギュレーションファイルには、シスコ製デバイスの機能をカスタマイズするためのCiscoIOSソフトウェアコマンドが含まれています。コマンドは、システムを起動したとき（startup-configファイルから）、またはコンフィギュレーションモードでCLIにコマンドを入力したときに、CiscoIOSソフトウェアによって解析（変換および実行）されます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 10 章

コンフィギュレーションの置換とロールバック

- [コンフィギュレーションの置換とロールバックの前提条件](#) (265 ページ)
- [コンフィギュレーションの置換とロールバックの制約事項](#) (266 ページ)
- [コンフィギュレーションの置換とロールバックについて](#) (266 ページ)
- [コンフィギュレーションの置換とロールバックの使用方法](#) (269 ページ)
- [コンフィギュレーションの置換とロールバックの設定例](#) (277 ページ)
- [コンフィギュレーションの置換およびコンフィギュレーションのロールバックの機能履歴](#) (279 ページ)

コンフィギュレーションの置換とロールバックの前提条件

コンフィギュレーションの置換とロールバックの機能に対する入力となるコンフィギュレーションファイルの形式は、標準の Cisco ソフトウェア コンフィギュレーションファイルの、次に示すインデント規則に準拠している必要があります。

- 新しい行のすべてのコマンドは、コマンドがコンフィギュレーションサブモードにない限り、インデントなしで開始します。
- レベル1 コンフィギュレーションサブモード内のコマンドは、スペース1個分インデントします。
- レベル2 コンフィギュレーションサブモード内のコマンドは、スペース2個分インデントします。
- 以下、続くサブモード内のコマンドは、同じようにインデントします。

これらのインデント規則には、ソフトウェアが **show running-config** や **copy running-config destination-url** などのコマンドのコンフィギュレーションファイルを作成する方法が記述されています。シスコデバイスで生成されるコンフィギュレーションファイルは、いずれもこうした規則に従います。

2つのコンフィギュレーションファイル（現在の実行コンフィギュレーションと、保存された置換用コンフィギュレーション）を合わせたサイズより大きな空きメモリが必要です。

コンフィギュレーションの置換とロールバックの制約事項

デバイスに、2つのコンフィギュレーションファイル（現在の実行コンフィギュレーションと、保存された置換用コンフィギュレーション）を合わせたサイズより大きな空きメモリがない場合、コンフィギュレーション置換操作は実行されません。

ネットワークデバイスの物理コンポーネント（物理インターフェイスなど）に関連する特定の Cisco コンフィギュレーション コマンドは、実行コンフィギュレーションについて追加または削除することはできません。たとえば、コンフィギュレーション置換操作を行っても、そのインターフェイスがデバイス上に物理的に存在する場合、現在の実行コンフィギュレーションから **interface ethernet 0** コマンド行を削除することはできません。同様に、**interface ethernet 1** コマンド行は、そのようなインターフェイスがデバイス上に物理的に存在しない場合、実行コンフィギュレーションに追加することはできません。コンフィギュレーション置換操作でこのタイプの変更を試行すると、その特定のコマンド行が失敗したことを示すエラーメッセージが表示されます。

非常にまれなケースですが、ルータをリロードしないと特定の Cisco 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から削除できないことがあります。コンフィギュレーション置換操作でこのタイプのコマンドの削除を試行すると、その特定のコマンド行が失敗したことを示すエラーメッセージが表示されます。

コンフィギュレーションの置換とロールバックについて

コンフィギュレーション アーカイブ

Cisco IOS コンフィギュレーション アーカイブは、**configure replace** コマンドにより提供されるコンフィギュレーションのロールバック機能を強化するために、Cisco IOS コンフィギュレーションファイルのアーカイブの保存、整理、管理を行うことを目的としたメカニズムです。この機能の導入前にも、実行コンフィギュレーションのコピーを **copy running-config destination-url** コマンドを使用して保存し、ローカルやリモートに置換ファイルを保管できました。ただし、この方法ではファイルの自動管理を行うことはできませんでした。一方、コンフィギュレーションの置換とロールバック機能では、実行コンフィギュレーションファイルを自動的に Cisco IOS コンフィギュレーション アーカイブに保存できます。アーカイブされたファイルはコンフィギュレーションのチェックポイントとして参照することができ、**configure replace** コマンドを使用して以前のコンフィギュレーション状態に戻すために利用できます。

archive config コマンドを使用すると、Cisco IOS コンフィギュレーションをコンフィギュレーションアーカイブに保存できます。その場合、標準のディレクトリとファイル名のプレフィク

スが使用され、バージョン番号（およびオプションでタイムスタンプ）が自動的に付加されます。バージョン番号は連続したファイルを保存するごとに、1つずつ大きくなります。この機能により、保存した Cisco IOS コンフィギュレーション ファイルを一貫して識別できます。アーカイブに保存する実行コンフィギュレーションの数は指定することができます。アーカイブ内のファイル数が上限値に達すると、次に最新のファイルが保存されるときに、最も古いファイルが自動的に消去されます。**show archive** コマンドを使用すると、Cisco IOS コンフィギュレーション アーカイブに保存されているすべてのコンフィギュレーション ファイルに関する情報が表示されます。

コンフィギュレーション ファイルを保存する Cisco IOS コンフィギュレーション アーカイブは、**configure replace** コマンドで使用することによって、FTP、HTTP、RCP、TFTP のファイルシステム上に配置できます。

コンフィギュレーションの置換

configure replace 特権 EXEC コマンドにより、現在の実行コンフィギュレーションを、保存しておいた Cisco IOS コンフィギュレーション ファイルで置換することができます。この機能は、コンフィギュレーションを保存しておいた状態へ戻すために使用することができます。そのコンフィギュレーション状態が保存された後にどのような変更が加えられても、効果的にロールバックさせることができます。

configure replace コマンドを使用するときは、現在の実行コンフィギュレーションと置換するための、保存された Cisco IOS コンフィギュレーション ファイルを指定する必要があります。置換ファイルは、Cisco IOS デバイスによって作成された完全なコンフィギュレーション (**copy running-config destination-url** コマンドによって作成されたものなど) であることが必要です。あるいは、置換ファイルを外部的に作成する場合は Cisco IOS デバイスが作成するファイル形式に完全に準拠していなければなりません。**configure replace** コマンドを入力すると、現在の実行コンフィギュレーションが指定された置換コンフィギュレーションと比較され、一連の diff が生成されます。2つのファイルの比較に使用されるアルゴリズムは、**show archive config differences** コマンドで使用されるものと同じです。置換コンフィギュレーションの状態になるよう、diffの結果が Cisco IOS パーサーによって適用されます。diffのみが適用されるため、現在の実行コンフィギュレーション上にすでに存在していた設定コマンドを再適用することにより生じる、潜在的なサービスの中断を避けられます。このアルゴリズムでは、順序に依存するコマンド（アクセスリストなど）へのコンフィギュレーション変更を、複数のパス プロセスを通して効果的に実行します。通常的环境下では、コンフィギュレーション置換操作の完了に必要なパスは3つまでであり、ループ動作を防ぐためのパスは最大5つまでに制限されます。

Cisco IOS **copy source-url running-config** 特権 EXEC コマンドは、保存された Cisco IOS コンフィギュレーション ファイルを実行コンフィギュレーションへコピーするためによく使用されます。**copy source-url running-config** コマンドを **configure replace target-url** 特権 EXEC コマンドの代わりに使用する場合、主な相違点として次の点に注意が必要です。

- **copy source-url running-config** コマンドはマージ動作であり、ソースファイルと現在の実行コンフィギュレーションの両方のコマンドがすべて保持されます。このコマンドでは、現在の実行コンフィギュレーションにのみ含まれ、ソースファイルには存在しないコマンドが削除されることはありません。これに対して、**configure replace target-url** コマンドで

は、置換ファイルに存在しないコマンドが現在の実行コンフィギュレーションから削除され、追加する必要があるコマンドが現在の実行コンフィギュレーションに追加されます。

- **copy source-url running-config** コマンドでは、現在の実行コンフィギュレーションにすでに存在しているかどうかにかかわらず、ソースファイル中のすべてのコマンドが適用されます。このアルゴリズムは効率的でない上、場合によってはサービスの停止が発生します。これに対して、**configure replace target-url** コマンドでは適用が必要なコマンドのみを適用し、現在の実行コンフィギュレーションに存在しているコマンドは再適用されません。
- **copy source-url running-config** コマンドでは部分的なコンフィギュレーション ファイルもコピー元として使用できますが、**configure replace target-url** コマンドの置換ファイルとして使用できるのは、完全な Cisco IOS コンフィギュレーション ファイルのみです。

コンフィギュレーション置換操作にロック機能が導入されました。**configure replace** コマンドが使用されると、コンフィギュレーション置換の動作中、デフォルトで実行コンフィギュレーション ファイルがロックされます。このロック メカニズムによって、置換動作の実行中に他のユーザーが実行コンフィギュレーションを変更しようとしたために、置換動作の不正終了が発生することを防止できます。**no lock** キーワードを **configure replace** コマンドの実行時に使用すると、実行コンフィギュレーションのロックをディセーブルにできます。

実行コンフィギュレーションのロックは、コンフィギュレーションの置換動作終了時に自動的にクリアされます。**show configuration lock** コマンドを使用すると、現在実行コンフィギュレーションに適用されているロックをすべて表示できます。

コンフィギュレーション ロールバック

ロールバックの概念は、データベースの操作ではトランザクション プロセス モデルに由来します。データベース トランザクションでは、あるデータベースのテーブルに一連の変更を加えることがあります。その後、変更を実行する（変更を恒久的に適用する）か、変更をロールバックする（変更を破棄してテーブルを以前の状態に戻す）かを選択することになります。ここでロールバックが意味するのは、変更のログを含んだジャーナルファイルが破棄され、何の変更も加えられないということです。ロールバック操作の結果として、加えた変更が適用される前の状態に戻ります。

configure replace コマンドを使用することで、以前のコンフィギュレーション状態へ戻ることが可能になり、コンフィギュレーション状態の保存後に加えた変更を効率的にロールバックさせることができます。Cisco IOS コンフィギュレーション ロールバックは、適用された一連の変更をもとにロールバック動作を行うのではなく、保存された Cisco コンフィギュレーション ファイルに基づいた特定のコンフィギュレーション状態へ戻るといったコンセプトを採用しています。このコンセプトは、チェックポイント（データベースの保存されたバージョン）に特定の状態を保存しておくという、データベースの考え方に類似しています。

コンフィギュレーションのロールバック機能が必要な場合、コンフィギュレーションの変更に先立って Cisco IOS 実行コンフィギュレーションを保存する必要があります。次に、コンフィギュレーションを変更した後に（**configure replace target-url** コマンドを使用し）保存したコンフィギュレーション ファイルを使って変更をロールバックします。保存された Cisco IOS コン

フィギュレーションファイルならどれでも置換コンフィギュレーションとして指定できるため、一部のロールバックモデルのように、ロールバックの数が制限されることもありません。

コンフィギュレーション ロールバック変更確認

コンフィギュレーションロールバック変更確認機能により、コンフィギュレーション変更の実行に際して確認を要求するようオプションで設定できます。この確認が受信できない場合、コンフィギュレーションは変更が適用される前の状態に戻されます。このメカニズムは、ネットワークデバイスとユーザーまたは管理アプリケーションとの接続において、コンフィギュレーション変更に起因する切断を防止するものです。

コンフィギュレーションの置換とロールバックの利点

- コンフィギュレーションの変更を効率的にロールバックさせて、以前のコンフィギュレーション状態へ戻ることが可能。
- `device` をリロードしたり、CLI で実行コンフィギュレーションファイルに加えた変更を手動で元に戻したりすることなく、現在の実行コンフィギュレーションファイルをスタートアップコンフィギュレーションファイルと置換できるため、システムのダウンタイムが減少。
- 保存しておいたどの Cisco IOS コンフィギュレーション状態に戻すことも可能。
- 追加や削除が必要なコマンドだけが影響される場合、`device` に完全なコンフィギュレーションファイルを適用することができるため、コンフィギュレーションの変更がシンプルに。
- `configure replace` コマンドを `copy source-url running-config` コマンドの代用として使用すると、現在の実行コンフィギュレーションにある既存のコマンドが再度適用されないため、効率が向上し、サービス停止のリスクが回避されます。

コンフィギュレーションの置換とロールバックの使用方法

コンフィギュレーションアーカイブの作成

`configure replace` コマンドを使用するうえで前提条件となる設定はありません。`configure replace` コマンドと、Cisco IOS コンフィギュレーションアーカイブおよび `archive config` コマンドとの併用は任意ですが、コンフィギュレーションロールバックのシナリオでは大きな利点があります。`archive config` コマンドを使用する前に、コンフィギュレーションアーカイブを設定しておく必要があります。コンフィギュレーションアーカイブの特性を設定するには、次の作業を実行します。

手順の概要

1. **enable**
2. **configure terminal**
3. **archive**
4. **path url**
5. **maximum number**
6. **time-period minutes**
7. **end**
8. **archive config**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	configure terminal 例： デバイス# 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	archive 例： デバイス(config)# archive	アーカイブ コンフィギュレーション モードを開始します。
ステップ 4	path url 例： デバイス(config-archive)# path flash:myconfiguration	Cisco IOS コンフィギュレーションアーカイブの場所と、ファイル名のプレフィックスを指定します。 (注) パスのところでファイルの代わりにディレクトリを指定する場合、ディレクトリ名は path flash:/directory/ のように後ろにスラッシュを付ける必要があります。このスラッシュはファイル名の後ろでは必要ありません。ディレクトリを指定する場合にだけ使います。
ステップ 5	maximum number 例： デバイス(config-archive)# maximum 14	(任意) Cisco IOS コンフィギュレーションアーカイブに保存される実行コンフィギュレーションのアーカイブファイル数の上限値を設定します。 • <i>number</i> 引数は、Cisco IOS コンフィギュレーションアーカイブに保存される実行コンフィギュレーションのアーカイブファイル数の上限値を

	コマンドまたはアクション	目的
		<p>示します。有効な値は 1 ～ 14 で、デフォルトは 10 です。</p> <p>(注) このコマンドを使用する前に、path コマンドを設定して Cisco IOS コンフィギュレーションアーカイブの位置とファイル名プレフィックスを指定しておく必要があります。</p>
<p>ステップ 6</p>	<p>time-period <i>minutes</i></p> <p>例 :</p> <p>デバイス(config-archive)# time-period 1440</p>	<p>(任意) Cisco IOS コンフィギュレーションアーカイブに実行コンフィギュレーションのアーカイブファイルを自動保存する間隔を設定します。</p> <ul style="list-style-type: none"> • Cisco IOS コンフィギュレーションアーカイブに現在の実行コンフィギュレーションのアーカイブファイルをどれほどの頻度で自動保存するかを、<i>minutes</i> 引数により分単位で指定します。 <p>(注) このコマンドを使用する前に、path コマンドを設定して Cisco IOS コンフィギュレーションアーカイブの位置とファイル名プレフィックスを指定しておく必要があります。</p>
<p>ステップ 7</p>	<p>end</p> <p>例 :</p> <p>デバイス(config-archive)# end</p>	<p>特権 EXEC モードに戻ります。</p>
<p>ステップ 8</p>	<p>archive config</p> <p>例 :</p> <p>デバイス# archive config</p>	<p>現在の実行設定ファイルを設定アーカイブに保存します。</p> <p>(注) このコマンドを使用する前に、path コマンドを設定する必要があります。</p>

コンフィギュレーションの置換またはロールバックの実行

保存された Cisco IOS コンフィギュレーションファイルで現在の実行コンフィギュレーションファイルを置換するには、次の作業を実行します。



(注) この手順の前に、コンフィギュレーションアーカイブを作成しておく必要があります。詳細については、[コンフィギュレーションアーカイブの作成](#)を参照してください。次に、現在の実行コンフィギュレーションで問題が生じた場合に、アーカイブしておいたコンフィギュレーションに戻す手順の詳細を示します。

手順の概要

1. **enable**
2. **configure replace** *target-url* [**nolock**] [**list**] [**force**] [**ignore case**] [**revert trigger** [**error**]] [**timer** *minutes*] | **time** *minutes*]]
3. **configure revert** { **now** | **timer** { *minutes* | **idle** *minutes* } }
4. **configure confirm**
5. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	<p>enable</p> <p>例 :</p> <p>デバイス> enable</p>	<p>特権 EXEC モードを有効にします。</p> <ul style="list-style-type: none"> • パスワードを入力します (要求された場合)。
ステップ 2	<p>configure replace <i>target-url</i> [nolock] [list] [force] [ignore case] [revert trigger [error]] [timer <i>minutes</i>] time <i>minutes</i>]]</p> <p>例 :</p> <p>デバイス# configure replace flash: startup-config time 120</p>	<p>保存しておいた Cisco IOS コンフィギュレーションファイルで現在の実行コンフィギュレーションファイルを置換します。</p> <ul style="list-style-type: none"> • target-url 引数は、archive config コマンドで作成されたコンフィギュレーションファイルなど、現在の実行コンフィギュレーションと置換する、保存された Cisco IOS コンフィギュレーションファイルの URL です (Cisco IOS ファイルシステムでアクセス可能なもの)。 • list キーワードは、コンフィギュレーション置換動作のパスごとに、Cisco IOS ソフトウェアパーサーによって適用されるコマンドラインのリストを表示します。実行されたパスの総数も表示されます。 • force キーワードは、現在の実行コンフィギュレーションから指定した Cisco IOS コンフィギュレーションファイルへの置換を、確認プロンプトを出さずに実行します。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • time minutes キーワードおよび引数は、現在の実行コンフィギュレーションファイルの置換確認のために configure confirm コマンドを入力しなければならない制限時間（分単位）を指定します。configure confirm コマンドが指定の制限時間内に入力されない場合、コンフィギュレーション置換操作は自動的に戻されます（つまり、現在の実行コンフィギュレーションファイルが configure replace コマンド入力以前のコンフィギュレーション状態へと回復されます）。 • nolock キーワードは、コンフィギュレーション置換操作中に他のユーザーが実行コンフィギュレーションを変更しないように実行コンフィギュレーションファイルをロックする機能をオフにします。 • revert trigger キーワードは、元のコンフィギュレーションへ戻すトリガーを次の内容から設定します。 <ul style="list-style-type: none"> • error : エラー時に元のコンフィギュレーションに戻します。 • timer minutes : 指定した時間が過ぎると元のコンフィギュレーションに戻します。 • ignore case キーワードで、コンフィギュレーションに確認コマンドの大文字と小文字の区別を無視させることができます。
<p>ステップ 3</p>	<p>configure revert { now timer { <i>minutes</i> <i>idle minutes</i> } }</p> <p>例 :</p> <pre>デバイス# configure revert now</pre>	<p>(任意) 時間指定ロールバックをキャンセルしてロールバックを即時トリガーする、または時間指定ロールバックのパラメータをリセットするには、特権 EXEC モードで configure revert コマンドを使用します。</p> <ul style="list-style-type: none"> • now : ロールバックをただちにトリガーします。 • timer : コンフィギュレーションを元に戻すタイマーをリセットします。 <ul style="list-style-type: none"> • 元に戻す時間を分単位で新たに指定するには、<i>minutes</i> 引数を timer キーワードとともに使用します。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> 保存されたコンフィギュレーションに戻すまでに、操作が行われないアイドル時間を最大どれほど長く許容できるかを設定するには、分単位の時間とともに idle キーワードを使用します。
ステップ 4	configure confirm 例： デバイス# <code>configure confirm</code>	(任意) 保存しておいた Cisco IOS コンフィギュレーションファイルの現在の実行コンフィギュレーションファイルへの置換を確認します。 (注) このコマンドは、 configure replace コマンドの time seconds キーワードおよび引数が指定されている場合にのみ使用します。
ステップ 5	exit 例： デバイス# <code>exit</code>	ユーザー EXEC モードに戻ります。

機能のモニタリングおよびトラブルシューティング

コンフィギュレーションの置換とロールバック機能をモニターおよびトラブルシューティングするには、この手順を実行します。

手順の概要

1. **enable**
2. **show archive**
3. **debug archive versioning**
4. **debug archive config timestamp**
5. **exit**

手順の詳細

ステップ 1 enable

このコマンドを使用して、特権 EXEC モードをイネーブルにします。パスワードを入力します（要求された場合）。

例：

```
デバイス> enable
デバイス#
```

ステップ2 show archive

CiscoIOS コンフィギュレーションアーカイブに保存されているファイルに関する情報を表示するには、次のコマンドを使用します。

例：

```

デバイス# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfiguration-2
Archive # Name
0
1      flash:myconfiguration-1 <- Most Recent
2
3
4
5
6
7
8
9
10
11
12
13
14
    
```

次に、実行コンフィギュレーションのアーカイブファイルをいくつか保存した状態で **show archive** コマンドを使用した場合の出力例を示します。この例では、保存されるアーカイブファイルの最大数が3に設定されています。

例：

```

デバイス# show archive
There are currently 3 archive configurations saved.
The next archive file will be named flash:myconfiguration-8
Archive # Name
0
1      :Deleted
2      :Deleted
3      :Deleted
4      :Deleted
5      flash:myconfiguration-5
6      flash:myconfiguration-6
7      flash:myconfiguration-7 <- Most Recent
8
9
10
11
12
13
14
    
```

ステップ3 debug archive versioning

このコマンドを使用して、CiscoIOS コンフィギュレーションアーカイブのアクティビティのデバッグを有効にして、コンフィギュレーションの置換とロールバックをモニターおよびトラブルシューティングします。

例：

```

デバイス# debug archive versioning
Jan  9 06:46:28.419:backup_running_config
Jan  9 06:46:28.419:Current = 7
Jan  9 06:46:28.443:Writing backup file flash:myconfiguration-7
Jan  9 06:46:29.547: backup worked
    
```

ステップ 4 debug archive config timestamp

このコマンドを使用して、コンフィギュレーション置換操作の各必須段階の処理時間、および操作中のコンフィギュレーションファイルのサイズのデバッグをイネーブルにします。

例：

```

デバイス# debug archive config timestamp
デバイス# configure replace flash:myconfiguration force
Timing Debug Statistics for IOS Config Replace operation:
  Time to read file usbflash0:sample_2.cfg = 0 msec (0 sec)
  Number of lines read:55
  Size of file          :1054
Starting Pass 1
  Time to read file system:running-config = 0 msec (0 sec)
  Number of lines read:93
  Size of file          :2539
  Time taken for positive rollback pass = 320 msec (0 sec)
  Time taken for negative rollback pass = 0 msec (0 sec)
  Time taken for negative incremental diffs pass = 59 msec (0 sec)
  Time taken by PI to apply changes = 0 msec (0 sec)
  Time taken for Pass 1 = 380 msec (0 sec)
Starting Pass 2
  Time to read file system:running-config = 0 msec (0 sec)
  Number of lines read:55
  Size of file          :1054
  Time taken for positive rollback pass = 0 msec (0 sec)
  Time taken for negative rollback pass = 0 msec (0 sec)
  Time taken for Pass 2 = 0 msec (0 sec)
Total number of passes:1
Rollback Done
    
```

ステップ 5 exit

このコマンドを使用して、ユーザー EXEC モードに戻ります。

例：

```

デバイス# exit
デバイス>
    
```

コンフィギュレーションの置換とロールバックの設定例

コンフィギュレーションアーカイブの作成

次の例は、Cisco IOS コンフィギュレーションアーカイブの初期設定を実行する方法を示しています。この例では、`flash:myconfiguration` がコンフィギュレーションアーカイブの保存位置およびファイル名のプレフィックスとして設定され、保存するアーカイブファイルが最大 10 個に設定されます。

```
configure terminal
!
archive
 path flash:myconfiguration
 maximum 10
end
```

現在の実行コンフィギュレーションを保存された Cisco IOS コンフィギュレーションファイルで置換

次の例では、`flash:myconfiguration` という名前で保存された Cisco IOS コンフィギュレーションファイルで現在の実行コンフィギュレーションを置換する方法を示します。`configure replace` コマンドでは、確認プロンプトでインタラクティブに操作を進めます。

```
デバイス# configure replace flash:myconfiguration
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Total number of passes: 1
Rollback Done
```

次の例では、コンフィギュレーション置換操作中に適用されるコマンドラインを表示するために、`list` キーワードを指定しています。

```
デバイス# configure replace flash:myconfiguration list
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
!Pass 1
!List of Commands:
no snmp-server community public ro
snmp-server community mystring ro

end
Total number of passes: 1
Rollback Done
```

スタートアップコンフィギュレーションファイルへの復帰

次の例に、**configure replace** コマンドを使用して Cisco IOS スタートアップコンフィギュレーションファイルへ復元する方法を示します。この例は、オプションの **force** キーワードを使用して、インタラクティブユーザープロンプトをオーバーライドする方法を示しています。

```
デバイス# configure replace flash:startup-config force
Total number of passes: 1
Rollback Done
```

configure confirm コマンドを使用したコンフィギュレーション置換操作の実行

次に、**configure replace** コマンドを **time minutes** キーワードおよび引数とともに使用する例を示します。現在の実行コンフィギュレーションファイルの置換を実行するには、指定の制限時間内に **configure confirm** コマンドを入力する必要があります。**configure confirm** コマンドが指定の制限時間内に入力されない場合、コンフィギュレーション置換操作は自動的に戻されます（つまり、現在の実行コンフィギュレーションファイルが **configure replace** コマンド入力以前のコンフィギュレーション状態へと回復されます）。

```
デバイス# configure replace flash:startup-config time 120
This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: Y
Total number of passes: 1
Rollback Done
デバイス# configure confirm
```

次に、**configure revert** コマンドを **timer** キーワードとともに使用する例を示します。時間指定ロールバックをキャンセルしてロールバックを即時トリガーする、または時間指定ロールバックのパラメータをリセットするには、**configure revert** コマンドを入力する必要があります。

```
デバイス# configure revert timer 100
```

コンフィギュレーションロールバック操作の実行

次の例は、現在実行中のコンフィギュレーションへの変更を行い、その変更をロールバックする方法を示しています。コンフィギュレーションロールバック操作の一部として、ファイルに変更を加える前に現在の実行コンフィギュレーションを保存する必要があります。この例では、現在の実行コンフィギュレーションの保存に **archive config** コマンドが使用されています。**configure replace** コマンドで生成された出力は、ロールバック操作を完了するために1つのパスのみが実行されたことを示します。



- (注) **archive config** コマンドを使用する前に、**path** コマンドを設定して Cisco IOS コンフィギュレーションアーカイブの位置とファイル名プレフィックスを指定しておく必要があります。

次のように、設定アーカイブの現在実行中のコンフィギュレーションを保存します。

```
archive config
```

それから、次の例に示すようにコンフィギュレーションの変更を入力します。

```
configure terminal
!
user netops2 password rain
user netops3 password snow
exit
```

実行コンフィギュレーションファイルに変更を加えた後、それらの変更をロールバックさせて、変更前のコンフィギュレーションに戻したくなくしたとします。**show archive** コマンドは、交換ファイルとして使用される設定のバージョンを確認するために使用されます。次の例に示すように、**configure replace** コマンドは交換コンフィギュレーションファイルへ戻すために使用されます。

```
デバイス# show archive
There are currently 1 archive configurations saved.
The next archive file will be named flash:myconfiguration-2
Archive # Name
0
1 flash:myconfiguration-1 <- Most Recent
2
3
4
5
6
7
8
9
10
デバイス# configure replace flash:myconfiguration-1
Total number of passes: 1
Rollback Done
```

コンフィギュレーションの置換およびコンフィギュレーションのロールバックの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	コンフィギュレーションの置換とロールバック	Cisco IOS コンフィギュレーションアーカイブは、 configure replace コマンドにより提供されるコンフィギュレーションのロールバック機能を強化するために、Cisco IOS コンフィギュレーションファイルのアーカイブの保存、整理、管理を行うことを目的としたメカニズムです。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 11 章

ソフトウェア メンテナンス アップグレード

ソフトウェア メンテナンス アップグレード (SMU) は、システムにインストールして修正やセキュリティ解決をリリースされたイメージに提供できるパッケージです。

- [ソフトウェア メンテナンス アップグレードの制約事項 \(281 ページ\)](#)
- [ソフトウェア メンテナンス アップグレードについて \(281 ページ\)](#)
- [ソフトウェア メンテナンスの更新の管理方法 \(283 ページ\)](#)
- [ソフトウェア メンテナンス アップグレードの設定例 \(285 ページ\)](#)
- [ソフトウェア メンテナンス アップグレードのその他の参考資料 \(289 ページ\)](#)
- [ソフトウェア メンテナンス アップグレードの機能の履歴 \(289 ページ\)](#)

ソフトウェア メンテナンス アップグレードの制約事項

- SMU は、インストールモードを使用したパッチのみをサポートします。

ソフトウェア メンテナンス アップグレードについて

次のセクションでは、ソフトウェア メンテナンス アップグレード (SMU) について詳しく説明します。

SMU の概要

SMU は、システムにインストールして修正やセキュリティ解決をリリースされたイメージに提供できるパッケージです。SMU パッケージはリリースごとおよびコンポーネントごとに提供されます。

SMU はネットワークの問題に迅速に対応できるようにするとともに、必要なテストの時間と範囲を削減するため、従来の Cisco IOS ソフトウェアには多大なメリットがあります。Cisco

IOS XE プラットフォームでは SMU の互換性を内部的に検証し、互換性のない SMU はインストールできません。

すべて SMU が後続の Cisco IOS XE ソフトウェア メンテナンス リリースに統合されています。SMU は独立した自己完結型パッケージであり、前提条件や依存関係はありません。SMU はどのような順序でもインストールまたはアンインストールできます。

SMU は拡張メンテナンスリリースでのみ、基盤となるソフトウェアリリースのライフサイクルにわたってサポートされます。

SMU をインストールするには、次の基本的な手順を実行します。

1. ファイルシステムに SMU を追加します。
2. システムで SMU をアクティブ化します。
3. リロードが繰り返されても持続させるための SMU の変更をコミットします。

SMU のワークフロー

SMU プロセスは、シスコカスタマーサポートへの要求によって開始されます。カスタマーサポートに連絡し、SMU 要求を行います。

SMU パッケージがリリースされると [Cisco Software Download]https://www.cisco.com/c/en_in/support/index.html ページに掲載されます。そのパッケージをダウンロードし、インストールします。

SMU パッケージ

SMU パッケージには、パッケージの内容を記述するいくつかのメタデータ、および SMU が要求されている報告済みの問題の修正とともに、リリースにパッチを適用するための一連のファイルがいくつか含まれています。SMU パッケージは、公開キーインフラストラクチャ (PKI) コンポーネントのパッチ適用もサポートします。

SMU のリロード

SMU タイプは、インストールされている SMU が対応するシステムに与える影響を示します。SMU がトラフィックに影響を与えない場合や、SMU によってデバイスの再起動、リロード、またはスイッチオーバーが発生する場合があります。リロードが必要かどうかを確認するには、**show install package flash: filename** コマンドを実行します。

ホットパッチを使用すると、SMU はアクティブ化後に有効になり、システムをリロードする必要がありません。SMU がコミットされると、リロードが繰り返されても変更が持続します。場合によっては、SMU でオペレーティングシステムのコールド (完全) リロードが必要になることがあります。このアクションは、リロードの間、トラフィックフローに影響します。コールドリロードが必要な場合、ユーザーにはアクションを確認するプロンプトが表示されます。

ソフトウェアメンテナンスの更新の管理方法

ここでは、SMU の管理に関する情報について説明します。

単一のコマンドまたは個別のコマンドを使用して SMU パッケージのインストール、アクティブ化、コミットを行うことができます。

SMU パッケージのインストール

このタスクでは、SMU パッケージをインストールするための **install add file activate commit** コマンドの使用方法を示します。

手順の概要

1. **enable**
2. **install add file flash: filename [activate commit]**
3. **exit**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： Device> enable	特権 EXEC モードを有効にします。プロンプトが表示されたらパスワードを入力します。
ステップ 2	install add file flash: filename [activate commit] 例： Device# install add file flash:cat9k_iosxe.BLD_SMU_20180302_085005_ TWIG_LATEST_20180306_013805.3.SSA.smu.bin activate commit	メンテナンス更新パッケージをリモートの場所から (FTP、HTTP、HTTPS、または TFTP を使用して) デバイスにコピーし、プラットフォームおよびイメージバージョンの互換性チェックを実行し、SMU パッケージをアクティブ化し、そのパッケージを複数回リロードしても維持されるようにします。このコマンドは、.bin ファイルの個別のコンポーネントをサブパッケージと packages.conf ファイルに抽出します。 (注) tftp を使用して SMU ファイルをコピーする場合は、ブートフラッシュを使用して SMU をアクティブにします。
ステップ 3	exit 例： Device# exit	特権 EXEC モードを終了し、ユーザー EXEC モードに戻ります。

SMU パッケージの管理

手順の概要

1. **enable**
2. **install add file flash: *filename***
3. **install activate file flash: *filename***
4. **install commit**
5. **install rollback to {base | committed | id *commit-ID*}**
6. **install deactivate file flash: *filename***
7. **install remove {file flash: *filename* | inactive}**
8. **show version**
9. **show install summary**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 : <pre>Device> enable</pre>	特権 EXEC モードを有効にします。プロンプトが表示されたらパスワードを入力します。
ステップ 2	install add file flash: <i>filename</i> 例 : <pre>Device# install add file flash:cat9k_iosxe.BLD_SMU_20180302_085005_ TWIG_LATEST_20180306_013805.3.SSA.smu.bin</pre>	SMU パッケージをソースの場所からデバイスにコピーし（ソースの場所がリモートの場合）、プラットフォームとイメージのバージョンの互換性チェックを実行し、必要に応じてすべてのメンバノードまたは FRU に SMU パッケージを追加します。このコマンドは、ファイルで基本的な互換性チェックを実行し、SMU パッケージがプラットフォームでサポートされていることも確認します。また、 <code>package/SMU.sta</code> ファイル内にエントリを追加することで、ステータスを監視し、維持できるようにします。
ステップ 3	install activate file flash: <i>filename</i> 例 : <pre>Device# install activate add file flash:cat9k_iosxe.BLD_SMU_20180302_085005_ TWIG_LATEST_20180306_013805.3.SSA.smu.bin</pre>	互換性チェックを実行し、パッケージをインストールして、パッケージのステータスの詳細を更新します。
ステップ 4	install commit 例 : <pre>Device# install commit</pre>	リロードが繰り返されても持続するようにアクティブ化の変更をコミットします。アクティブ化の後で、システムの起動時、または最初のリロード後にコミットできます。パッケージがアクティブになってもコミットされていない場合は、最初のリ

	コマンドまたはアクション	目的
		ロード後はアクティブの状態を保ちますが、2回目のリロード後はアクティブ状態を保ちません。
ステップ 5	install rollback to {base committed id commit-ID} 例： Device# install rollback to committed	デバイスを以前のインストール状態に戻します。
ステップ 6	install deactivate file flash: filename 例： Device# install deactivate file flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin	アクティブパッケージを非アクティブ化し、パッケージのステータスを更新します。
ステップ 7	install remove {file flash: filename inactive} 例： Device# install remove file flash:cat9k_iosxe.BLD_SMU_20180302_085005_TWIG_LATEST_20180306_013805.3.SSA.smu.bin	指定した SMU が非アクティブかどうかを確認し、非アクティブの場合はファイルシステムから削除します。 inactive オプションは、非アクティブなパッケージをファイルシステムからすべて削除します。
ステップ 8	show version 例： Device# show version	デバイスのイメージバージョンを表示します。
ステップ 9	show install summary 例： Device# show install summary	パッケージのインストールステータスに関する情報を表示します。このコマンドの出力は、設定されている install コマンドに応じて変化します。

ソフトウェアメンテナンスアップグレードの設定例

次に、SMU の設定例を示します。

次に、SMU ファイルを TFTP にコピーする例を示します。

```
Device# copy copy
ftp://172.19.1.250//auto/ftpboot/user/cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin

tftp:Destination filename [cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin]?

Accessing
tftp://172.16.0.1//auto/tftpboot/folder1/cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin...
Loading /auto/tftpboot/folder1/cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin from
172.16.0.1 (via GigabitEthernet0): !
[OK - 17668 bytes]
17668 bytes copied in 0.058 secs (304621 bytes/sec)
```

次に、**show install summary** コマンドの出力例を示します。

```
Device# show install summary
```

```
[ Switch 1 ] Installed Package(s) Information:
State (St): I - Inactive, U - Activated & Uncommitted,
C - Activated & Committed, D - Deactivated & Uncommitted
```

```
-----
Type St Filename/Version
-----
```

```
SMU C flash:cat9k_lite_iosxe.16.09.04.CSCvk70181.SPA.smu.bin
IMG C 16.6.3.0
```

次に、メンテナンス更新プログラムパッケージファイルを追加する例を示します。

```
Device# install add file flash:cat9k-
universalk9.2017-01-10_13.15.1.CSCxxxxxxx.SSA.dmp.bin
```

```
install_add: START Sat Feb 26 14:06:04 PST 2017
SUCCESS: install_add flash:cat9k_iosxe.2017-01-10_13.15.1.CSCxxxxxxx.SSA.dmp.bin
Sat Feb 26 14:06:12 PST 2017
Device#
```

次に、SMU パッケージファイルをデバイスに追加した後の **show install summary** コマンドの出力例を示します。

```
Device# show install summary
```

```
Active Packages:
No packages
Inactive Packages:
flash:cat9k-universalk9.2017-01-10_13.15.1.CSCxxxxxxx.SSA.dmp.bin
Committed Packages:
No packages
Uncommitted Packages:
No packages
Device#
```

次に、追加した SMU パッケージファイルをアクティブ化する例を示します。

```
Device# install activate file flash:cat9k_iosxe.2017-01-10_13.15.1.
CSCxxxxxxx.SSA.dmp.bin
```

```
install_activate: START Sat Feb 26 14:10:55 PST 2017
The activation step would require a reload. Do you want to proceed? [y/n]y
Regular SMU. Reloading the box to complete activation of the SMU...
Feb 26 14:11:23.873 R0/0: %PMAN-5-EXITACTION: Process manager is exiting:
reload action requested
Initializing Hardware ...
Checking for PCIe device presence...done
System integrity status: 0x610
Rom image verified correctly
<after reload>
Device#
```

次に、**show version** コマンドの出力例を示します。

```
Device# show version
```

```
Cisco IOS XE Software, Version BLD_POLARIS_DEV_SMU_LATEST_20170110_13.15.1 -
SMU-PATCHED
Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE), Experimental
```

```
Version
16.6.20170110_13.15.1 [BLD_V166_SMU_LATEST_20170127_13.15.1 SMU-PATCHED]
Copyright (c) 1986-2017 by Cisco Systems, Inc.
Compiled Sat 26-Feb-17 16:07 by mcpre
...
```

次に示すのは、**show install summary** コマンドがモデルパッケージのステータスをアクティブでありコミット未完了と表示する場合の出力例です。

```
Device# show install summary

Active Packages:
  flash:cat9k-universalk9.2017-01-10_13.15.1.CSCxxx.SSA.dmp.bin
Inactive Packages:
  No packages
Committed Packages:
  flash:cat9k-universalk9.2017-01-10_13.15.1.CSCxxx.SSA.dmp.bin
Uncommitted Packages:
  No packages
Device#
```

次に、**show install active** コマンドの出力例を示します。

```
Device# show install active

Active Packages:
flash:cat9k-universalk9.2017-01-10_13.15.1.CSCxxx.SSA.dmp.bin
```

次の例では、**install commit** コマンドの実行方法を示しています。

```
Device# install commit

install_commit: START Sat Feb 26 06:46:48 UTC 2017
SUCCESS: install_commit Sat Feb 26 06:46:52 UTC 2017
Device#
```

次に示すのは、**show install summary** コマンドが、更新パッケージがコミットされてリロードが繰り返されても持続することを表示する場合の出力例です。

```
Device# show install summary

Active Packages:
flash:cat9k-universalk9.2017-01-10_13.15.1.CSCxxx.SSA.dmp.bin
Inactive Packages:
No packages
Committed Packages:
flash:cat9k-universalk9.2017-01-10_13.15.1.CSCxxx.SSA.dmp.bin
Uncommitted Packages:
No packages
Device#
```

次に、更新プログラムパッケージをコミットしたパッケージにロールバックする例を示します。

```
Device# install rollback to base

install_rollback: START Sat Feb 26 11:27:41 PST 2017
This rollback would require a reload. Do you want to proceed? [y/n]y
2 install_rollback: Reloading the box to take effect
```

```

Initializing Hardware ...
<after reload>
Device#

```

次に、**show install summary** コマンドの出力例を示します。

```

Device# show install summary

Active Packages:
flash:cat9k-universalk9.2017-01-10_13.15.1.CSCxxx.SSA.dmp.bin
Inactive Packages:
No packages
Committed Packages:
flash:cat9k-universalk9.2017-01-10_13.15.1.CSCxxxxxxxx.SSA.dmp.bin
Uncommitted Packages:
No packages
Device#

```

次に、**show install log** コマンドの出力例を示します。

```

Device# show install log

[0|install_op_boot]: START Sat Feb 26 19:31:50 Universal 2017
[0|install_op_boot]: END SUCCESS Sat Feb 26 19:31:56 Universal 2017

```

次に、SMU パッケージ ファイルを非アクティブ化する例を示します。

```

Device# install deactivate file flash:cat9k-
universalk9.2017-01-10_13.15.1.CSCxxxxxxxx.SSA.dmp.bin

install_deactivate: START Sat Feb 26 10:49:07 PST 2017
The activation step would require a reload. Do you want to proceed? [y/n]y
Regular SMU. Reloading the box to complete activation of the SMU...

Initializing Hardware...
...
<after reload>
Device#

```

次に、**show install summary** コマンドの出力例を示します。

```

Device# show install summary

Active Packages:
No packages
Inactive Packages: flash:cat9k-universalk9.2017-01-10_13.15.1.CSCxxxxxxxx.SSA.dmp.bin
Committed Packages:
No packages
Uncommitted Packages:
No packages
Device#

```

次に、デバイスから SMU を削除する例を示します。

```

Device# install remove file flash:cat9k-
universalk9.2017-01-10_13.15.1.CSCxxxxxxxx.SSA.dmp.bin

install_remove: START Sat Feb 26 12:09:43 PST 2017
SUCCESS: install_remove /flash/cat9k-universalk9.2017-01-10_13.15.1.

```



```
CSCxxxxxxxx.SSA.dmp.bin Sat Feb 26 12:09:49 PST 2017
Device#
```

次に、**show install summary** コマンドの出力例を示します。

```
Device# show install summary

Active Packages:
No packages
Inactive Packages:
No packages
Committed Packages:
No packages
Uncommitted Packages:
No packages
```

ソフトウェアメンテナンスアップグレードのその他の参考資料

関連資料

関連項目	マニュアル タイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	<i>Command Reference (Catalyst 9300 Series Switches)</i>

ソフトウェアメンテナンスアップグレードの機能の履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.6.1	ソフトウェアメンテナンスアップグレード (SMU)	SMUは、システムにインストールして修正やセキュリティ解決をリリースされたイメージに提供ができるパッケージです。
Cisco IOS XE Fuji 16.9.1	ホットパッチ	ホットパッチを使用すると、SMUはアクティブ化後に有効になり、システムをリロードする必要がありません。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 12 章

フラッシュ ファイル システムの操作

- 機能情報の確認 (291 ページ)
- フラッシュ ファイル システムについて (291 ページ)
- 使用可能なファイル システムの表示 (292 ページ)
- デフォルト ファイル システムの設定 (295 ページ)
- ファイル システムのファイルに関する情報の表示 (295 ページ)
- ディレクトリの変更および作業ディレクトリの表示 (296 ページ)
- ディレクトリの作成 (297 ページ)
- ファイルのコピー (298 ページ)
- ファイルの作成、表示、および抽出 (300 ページ)
- フラッシュ ファイル システムに関するその他の関連資料 (302 ページ)
- フラッシュファイルシステムの機能履歴 (303 ページ)

機能情報の確認

ご使用のソフトウェアリリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェア リリースの **Bug Search Tool** およびリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、**Cisco Feature Navigator** を使用します。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> からアクセスします。Cisco.com のアカウントは必要ありません。

フラッシュ ファイル システムについて

フラッシュ ファイル システムは、ファイルを格納できる単一のフラッシュ デバイスです。ソフトウェア バンドルおよびコンフィギュレーション ファイルの管理に役立つ複数のコマンドも備えています。device のデフォルトのフラッシュファイルシステムは **flash:** です。

アクティブなdeviceまたは任意のスタック メンバから見ると、**flash:** はローカルフラッシュ デバイスを指します。これは、ファイルシステムが表示されているのと同一deviceに接続されているデバイスです。device スタックでは、さまざまなスタックメンバからの各フラッシュ デバイスを、アクティブなdeviceから参照できます。これらのフラッシュファイルシステムの名前には、対応するdeviceメンバ番号が含まれています。たとえば、アクティブなdeviceから参照できる **flash-3:** は、スタックメンバ3にある **flash:** と同じファイルシステムを指します。device スタックにあるフラッシュファイルシステムを含む、すべてのファイルシステムのリストを表示するには、**show file systems** 特権 EXEC コマンドを使用します。

device スタックでは、一度に1人のユーザのみが、ソフトウェアバンドルおよびコンフィギュレーション ファイルを管理できます。

使用可能なファイル システムの表示

deviceで使用可能なファイルシステムを表示するには、**show file systems** 特権 EXEC コマンドを使用します（次のスタンドアロンdeviceの例を参照）。

```

デバイス# show file systems
Size(b) Free(b) Type Flags Prefixes
- - opaque rw system:
- - opaque rw tmpsys:
1651314688 1559785472 disk rw crashinfo:
* 11353194496 9693396992 disk rw flash:
8049967104 7959392256 disk ro webui:
- - opaque rw null:
- - opaque ro tar:
- - network rw tftp:
2097152 2080848 nvram rw nvram:
- - opaque wo syslog:
- - network rw rcp:
- - network rw http:
- - network rw ftp:
- - network rw scp:
- - network rw https:
- - opaque ro cns:

デバイス# show file systems
File Systems:

          Size(b)          Free(b)          Type  Flags  Prefixes
          -            -            opaque  rw    system:
          -            -            opaque  rw    tmpsys:
* 11250098176  9694093312    disk    rw    bootflash: flash:
  1651314688    1232220160    disk    rw    crashinfo:
118148280320  112084115456  disk    rw    disk0:
  189628416     145387520     disk    rw    usbflash0:
  7763918848    7696850944    disk    ro    webui:
          -            -            opaque  rw    null:
          -            -            opaque  ro    tar:
          -            -            network  rw    tftp:
          33554432    33532852      nvram    rw    nvram:
          -            -            opaque  wo    syslog:
          -            -            network  rw    rcp:

```

```

-          - network      rw  http:
-          - network      rw  ftp:
-          - network      rw  scp:
-          - network      rw  https:
-          - opaque       ro  cns:

```

次の例では、**device**スタックを示します。この例では、アクティブな **device** がスタック メンバ 1 です。スタック メンバ 2 のファイル システムはフラッシュ 2 として表示されます。スタック メンバ 3 のファイル システムはフラッシュ 3 として表示されます。同様にして、スタック メンバ 9 まで、スタック メンバ 9 はフラッシュ 9、。また、この例では、次のように、**crashinfo** ディレクトリと、アクティブな **device** に接続された USB フラッシュドライブも示します。

```
デバイス# show file systems
```

```
File Systems:
```

```

Size(b) Free(b) Type Flags Prefixes
- - opaque rw system:
- - opaque rw tmpsys:
1651314688 1520742400 disk rw crashinfo: crashinfo-1:
1651507200 1516240896 disk rw crashinfo-2: stby-crashinfo:
1651507200 1517289472 disk rw crashinfo-3:
1651507200 1519386624 disk rw crashinfo-4:
1651507200 1524629504 disk rw crashinfo-5:
1651507200 1523580928 disk rw crashinfo-6:
1651507200 1517289472 disk rw crashinfo-7:
1651507200 1526726656 disk rw crashinfo-8:
* 11353194496 7916576768 disk rw flash: flash-1:
11353980928 7944011776 disk rw flash-2: stby-flash:
11353980928 7876902912 disk rw flash-3:
11353980928 7944011776 disk rw flash-4:
11353980928 7939817472 disk rw flash-5:
11353980928 7944011776 disk rw flash-6:
11353980928 7944011776 disk rw flash-7:
11353980928 7944011776 disk rw flash-8:
3824013312 3756507136 disk ro webui:
- - opaque rw null:
- - opaque ro tar:
- - network rw tftp:
2097152 2052489 nvram rw nvram:
- - opaque wo syslog:
- - network rw rcp:
- - network rw http:
- - network rw ftp:
- - network rw scp:
- - network rw https:
- - opaque ro cns:
2097152 2052489 nvram rw stby-nvram:
- - nvram rw stby-rdfs:
- - opaque rw revrdfs:

```

表 11: *show file systems* のフィールドの説明

フィールド	値
Size(b)	ファイル システムのメモリ サイズ (バイト単位) です。
Free(b)	ファイル システムの空きメモリ サイズ (バイト単位) です。

フィールド	値
Type	<p>ファイル システムのタイプです。</p> <p>disk : ファイルシステムは、フラッシュ メモリ デバイス、USB フラッシュ、crashinfo ファイル用です。</p> <p>network : ファイルシステムは、FTP サーバや HTTP サーバなどのネットワーク デバイス用です。</p> <p>nvram : ファイルシステムは NVRAM (不揮発性 RAM) デバイス用です。</p> <p>opaque : ファイルシステムは、ローカルに生成された pseudo ファイルシステム (system など)、またはダウンロード インターフェイス (brimux など) です。</p> <p>unknown : ファイル システムのタイプは不明です。</p>
Flags	<p>ファイル システムの権限です。</p> <p>ro : 読み取り専用です。</p> <p>rw : 読み取りおよび書き込みです。</p> <p>wo : 書き込み専用です。</p>
Prefixes	<p>ファイル システムのエイリアスです。</p> <p>crashinfo : crashinfo ファイルです。</p> <p>flash: : フラッシュ ファイル システムです。</p> <p>ftp : FTP サーバです。</p> <p>http : HTTP サーバです。</p> <p>https : セキュア HTTP サーバです。</p> <p>nvram : : NVRAM です。</p> <p>null : : コピーのヌル宛先です。リモート ファイルをヌルへコピーして、サイズを判別できます。</p> <p>rcp : Remote Copy Protocol (RCP) サーバです。</p> <p>scp : Session Control Protocol (SCP) サーバです。</p> <p>system : : 実行コンフィギュレーションを含むシステム メモリが格納されています。</p> <p>tftp : : TFTP ネットワーク サーバです。</p> <p>usbflash0 : USB フラッシュ メモリです。</p> <p>ymodem : : YMODEM プロトコルを使用して、ネットワーク マシンからファイルを取得します。</p>

デフォルト ファイル システムの設定

デフォルトのファイル システムとして使用されるファイル システムまたはディレクトリを指定するには、**cd filesystem:** 特権 EXEC コマンドを使用します。デフォルト ファイル システムを設定すると、関連するコマンドを実行するときに *filesystem:* 引数を省略できます。たとえば、オプションの *filesystem:* 引数を持つすべての特権 EXEC コマンドでは、**cd** コマンドで指定されたファイル システムが使用されます。

デフォルトでは、デフォルト ファイル システムは *flash:* です。

cd コマンドで指定された現在のデフォルトのファイル システムを表示するには、**pwd** 特権 EXEC コマンドを使用します。

ファイル システムのファイルに関する情報の表示

ファイル システムの内容を操作する前に、そのリストを表示できます。たとえば、新しいコンフィギュレーション ファイルをフラッシュ メモリにコピーする前に、ファイル システムに同じ名前のコンフィギュレーション ファイルが格納されていないことを確認できます。同様に、フラッシュ コンフィギュレーション ファイルを別の場所にコピーする前に、ファイル名を確認して、その名前を別のコマンドで使用できます。ファイル システムのファイルに関する情報を表示するには、次の表に記載する特権 EXEC コマンドのいずれかを使用します。

表 12: ファイルに関する情報を表示するためのコマンド

コマンド	説明
dir [<i>/all</i>] [<i>filesystem:filename</i>]	ファイル システムのファイル リストを表示します。
show file systems	ファイル システムのファイルごとの詳細を表示します。
show file information <i>file-url</i>	特定のファイルに関する情報を表示します。
show file descriptors	開いているファイルの記述子のリストを表示します。ファイル記述子は開いているファイルの内部表現です。このコマンドを使用して、別のユーザーによってファイルが開かれているかどうかを調べることができます。

たとえば、ファイル システムのすべてのファイルのリストを表示するには、次のように **dir** 特権 EXEC コマンドを使用します。

```
device# dir flash:
Directory of bootflash:/

616513  drwx           4096  Jul 15 2015 07:11:35 +00:00  .installer
608402  -rw-          33818  Sep 25 2015 11:41:35 +00:00  bootloader_evt_handle.log
608403  drwx           4096  Feb 27 2017 13:56:47 +00:00  .ssh
```

ディレクトリの変更および作業ディレクトリの表示

```

608410 -rw-          0   Jun 5 2015 10:16:17 +00:00 dc_stats.txt
608411 drwx        20480 Sep 23 2015 11:50:13 +00:00 core
624625 drwx         4096 Sep 23 2015 12:29:27 +00:00 .prst_sync
640849 drwx         4096 Feb 27 2017 13:57:30 +00:00 .rollback_timer
608412 drwx         4096 Jun 17 2015 18:12:47 +00:00 orch_test_logs
608413 -rw-    33554432 Sep 25 2015 11:43:15 +00:00 nvram_config
608417 -rw-          35   Sep 25 2015 20:17:42 +00:00 pnp-tech-time
608439 -rw-    214054 Sep 25 2015 20:17:48 +00:00 pnp-tech-discovery-summary
608419 drwx         4096 Jul 23 2015 07:50:25 +00:00 util
616514 drwx         4096 Mar 18 2015 11:09:04 +00:00 onep
608442 -rw-          55   Mar 18 2015 11:19:34 +00:00 vlan.dat
608448 -rw-    1131779 Mar 28 2015 13:13:48 +00:00 log.txt
616516 drwx         4096 Apr 1 2015 09:34:56 +00:00 gs_script
616517 drwx         4096 Apr 6 2015 09:42:38 +00:00 tools
608440 -rw-          252 Sep 25 2015 11:41:52 +00:00 boothelper.log
624626 drwx         4096 Apr 17 2015 06:10:55 +00:00 SD_AVC_AUTO_CONFIG
608488 -rw-    98869 Sep 25 2015 11:42:15 +00:00 memleak.tcl
608437 -rwx     17866 Jul 16 2015 04:01:10 +00:00 ardbeg_x86
632745 drwx         4096 Aug 20 2015 11:35:09 +00:00 CRDU
632746 drwx         4096 Sep 16 2015 08:57:44 +00:00 ardmore
608418 -rw-    1595361 Jul 8 2015 11:18:33 +00:00
system-report_RP_0_20150708-111832-UTC.tar.gz
608491 -rw-    67587176 Aug 12 2015 05:30:35 +00:00 mcln_x86_kernel_20170628.SSA
608492 -rwx     74880100 Aug 12 2015 05:30:57 +00:00 stardust.x86.idprom.0718B

11250098176 bytes total (9128050688 bytes free)
device#

```

ディレクトリの変更および作業ディレクトリの表示

ディレクトリを変更し、作業ディレクトリを表示するには、次の手順を実行します。

手順の概要

1. **enable**
2. **dir filesystem:**
3. **cd directory_name**
4. **pwd**
5. **cd**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例 : デバイス> enable	特権 EXEC モードを有効にします。 <ul style="list-style-type: none"> • パスワードを入力します（要求された場合）。
ステップ 2	dir filesystem: 例 :	指定されたファイルシステムのディレクトリを表示します。

	コマンドまたはアクション	目的
	デバイス# dir flash:	<i>filesystem:</i> には、システム ボードのフラッシュ デバイスの <i>flash:</i> を使用します。 スタックの <i>device</i> メンバーのフラッシュパーティションにアクセスするには、 <i>flash-n</i> を使用します。 <i>n</i> は、スタックメンバー番号を表します。例えば、 <i>flash-4</i> 。
ステップ 3	cd <i>directory_name</i> 例 : デバイス# cd new_configs	指定されたディレクトリへ移動します。 コマンド例では、 <i>new_configs</i> という名前のディレクトリに移動する方法を示します。
ステップ 4	pwd 例 : デバイス# pwd	作業ディレクトリを表示します。
ステップ 5	cd 例 : デバイス# cd	デフォルト ディレクトリに移動します。

ディレクトリの作成

特権 EXEC モードを開始して、ディレクトリを作成するには次の手順を実行します。

手順の概要

1. **dir** *filesystem:*
2. **mkdir** *directory_name*
3. **dir** *filesystem:*

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	dir <i>filesystem:</i> 例 : デバイス# dir flash:	指定されたファイルシステムのディレクトリを表示します。 <i>filesystem:</i> には、システム ボードのフラッシュ デバイスの <i>flash:</i> を使用します。
ステップ 2	mkdir <i>directory_name</i> 例 :	新しいディレクトリを作成します。スラッシュ (/) 間に指定できるディレクトリ名は最大 45 文字で、大

	コマンドまたはアクション	目的
	デバイス# mkdir new_configs	文字と小文字の区別があります。ディレクトリ名には制御文字、スペース、スラッシュ、引用符、セミコロン、またはコロンは使用できません。
ステップ 3	dir filesystem: 例： デバイス# dir flash:	入力を確認します。

ディレクトリの削除

ディレクトリを、その内部のすべてのファイルおよびサブディレクトリとともに削除するには、**delete /force /recursive filesystem:/file-url** 特権 EXEC コマンドを使用します。

名前で指定されたディレクトリを、その内部のすべてのサブディレクトリおよびファイルとともに削除するには、**/recursive** キーワードを使用します。ディレクトリ内のファイルごとに表示される、削除を確認するためのプロンプトを省略するには、**/force** キーワードを使用します。この削除プロセスを実行すると、最初に 1 度だけプロンプトが表示されます。

filesystem には、システム ボードのフラッシュ デバイスの **flash:** を使用します。**file-url** には、削除するディレクトリの名前を入力します。ディレクトリ内のすべてのファイルおよびディレクトリが削除されます。



注意 ディレクトリが削除された場合、その内容は回復できません。

ファイルのコピー

送信元から宛先にファイルをコピーするには、**copy source-url destination-url** 特権 EXEC コマンドを使用します。送信元および宛先の URL には、**running-config** および **startup-config** キーワードショートカットを使用できます。たとえば、**copy running-config startup-config** コマンドは、現在実行中のコンフィギュレーション ファイルをフラッシュメモリの NVRAM セクションに保存し、システム初期化の際にコンフィギュレーションファイルとして使用されるようにします。

XMODEM または YMODEM プロトコルを使用するネットワーク マシンのファイルに対する送信元として特殊なファイルシステム (**xmodem:**、**ymodem:**) を指定し、そこからコピーすることもできます。

ネットワーク ファイルシステムの URL には、ftp:、rtp:、tftp:、scp:、http:、https: などがあり、構文は次のとおりです。

- FTP : ftp:[[/username [:password]@location]/directory]/filename

- RCP : `rcp:[[/username@location]/directory]/filename`
- TFTP : `tftp:[[/location]/directory]/filename`
- SCP : `scp:[[/username [:password]@location]/directory]/filename`
- HTTP : `http:[[/username [:password]@location]/directory]/filename`
- HTTPS : `https:[[/username [:password]@location]/directory]/filename`



(注) パスワードに特殊文字「@」を含めることはできません。文字「@」を使用すると、コピーでサーバの IP アドレスを解析できません。

ローカルにある書き込み可能なファイル システムには `flash:` があります。

送信元および宛先の組み合わせによっては、無効な場合があります。特に、次に示す組み合わせの場合は、コピーできません。

- 実行コンフィギュレーションから実行コンフィギュレーションへ
- スタートアップ コンフィギュレーションからスタートアップ コンフィギュレーションへ
- デバイスから同じ名前のデバイスに（たとえば、**copy flash: flash:** コマンドは無効）

ファイルの削除

フラッシュ メモリ デバイスのファイルが不要になった場合は、そのファイルを永久に削除できます。指定されたフラッシュ デバイスからファイルまたはディレクトリを削除するには、**delete [/force] [/recursive] [filesystem:]file-url** 特権 EXEC コマンドを使用します。

ディレクトリを、その内部のすべてのサブディレクトリやファイルとともに削除するには、**/recursive** キーワードを使用します。ディレクトリ内のファイルごとに表示される、削除を確認するためのプロンプトを省略するには、**/force** キーワードを使用します。この削除プロセスを実行すると、最初に 1 度だけプロンプトが表示されます。**archive download-sw** コマンドでインストールされ、不要になった古いソフトウェアイメージを削除するには、**/force** キーワードおよび **/recursive** キーワードを使用します。

filesystem: オプションを省略すると、**device** は **cd** コマンドで指定したデフォルトのデバイスを使用します。*file-url* には、削除するファイルのパス（ディレクトリ）および名前を指定します。

ファイルを削除しようとする時、削除の確認を求めるプロンプトが表示されます。



注意 ファイルが削除された場合、その内容は回復できません。

ここでは、デフォルトのフラッシュ メモリ デバイスからファイル *myconfig* を削除する例を示します。

```
デバイス# delete myconfig
```

ファイルの作成、表示、および抽出

ファイルを作成してそこにファイルを書き込んだり、ファイル内のファイルをリスト表示したり、ファイルからファイルを抽出したりできます（次の項を参照）。

ファイルの作成、内容の表示、およびファイルの抽出を行うには、特権 EXEC コマンドで次の手順を実行します。

手順の概要

1. **archive tar /create destination-url flash: /file-url**
2. **archive tar /table source-url**
3. **archive tar /xtract source-url flash:/file-url [dir/file...]**
4. **more [/ascii | /binary | /ebcdic] /file-url**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	archive tar /create destination-url flash: /file-url 例 : <pre>device# archive tar /create tftp:172.20.10.30/saved. flash:/new-configs</pre>	ファイルを作成し、そこにファイルを追加します。 destination-url には、ローカルまたはネットワークファイルシステムの宛先 URL のエイリアス、および作成するファイルの名前を指定します。 <ul style="list-style-type: none"> • ローカルフラッシュファイルシステム構文 flash: • FTP 構文 ftp:[[/username[:password]@location]/directory]/-filename. • RCP 構文 rcp:[[/username@location]/directory]/-filename. • TFTP 構文 tftp:[[/location]/directory]/-filename. flash:/file-url には、ローカルフラッシュファイルシステム上の、新しいファイルが作成される場所を指定します。送信元ディレクトリ内に格納されている任意のファイルまたはディレクトリの一覧を指定して、新しいファイルに追加することもできます。何も指定しないと、このレベルにおけるすべてのファイルおよびディレクトリが、新規に作成されたファイルに書き込まれます。

	コマンドまたはアクション	目的
ステップ 2	<p>archive tar /table source-url</p> <p>例 :</p> <pre>device# archive tar /table flash: /new_configs</pre>	<p>ファイルの内容を表示します。</p> <p><i>source-url</i> には、ローカル ファイル システムまたはネットワーク ファイル システムの送信元 URL エイリアスを指定します。-<i>filename</i> は、表示するファイルです。次のオプションがサポートされています。</p> <ul style="list-style-type: none"> ローカル フラッシュ ファイル システム構文 <p>flash:</p> <ul style="list-style-type: none"> FTP 構文 <p>ftp:[[//username[:password]@location]/directory]/-filename.</p> <ul style="list-style-type: none"> RCP 構文 <p>rcp:[[//username@location]/directory]/-filename.</p> <ul style="list-style-type: none"> TFTP 構文 <p>tftp:[[//location]/directory]/-filename.</p> <p>ファイルのあとにファイルまたはディレクトリのリストを指定して、ファイルの表示を制限することもできます。指定したファイルだけが表示されます。何も指定しないと、すべてのファイルおよびディレクトリが表示されます。</p>
ステップ 3	<p>archive tar /xtract source-url flash:/file-url [dir/file...]</p> <p>例 :</p> <pre>device# archive tar /xtract tftp:/172.20.10.30/saved. flash:/new-configs</pre>	<p>ファイルをフラッシュ ファイル システム上のディレクトリに抽出します。</p> <p><i>source-url</i> には、ローカル ファイル システムの送信元 URL のエイリアスを指定します。-<i>filename</i> は、ファイルの抽出元のファイルです。次のオプションがサポートされています。</p> <ul style="list-style-type: none"> ローカル フラッシュ ファイル システム構文 <p>flash:</p> <ul style="list-style-type: none"> FTP 構文 <p>ftp:[[//username[:password]@location]/directory]/-filename.</p> <ul style="list-style-type: none"> RCP 構文 <p>rcp:[[//username@location]/directory]/-filename.</p> <ul style="list-style-type: none"> TFTP 構文 <p>tftp:[[//location]/directory]/-filename.</p> <p>flash:/file-url [dir/file...] には、ファイルの抽出元にするローカル フラッシュ ファイル システム上の場所を指定します。抽出対象のファイル内のファイルまたはディレクトリのリストを指定するには、<i>dir/file...</i></p>

	コマンドまたはアクション	目的
		オプションを使用します。何も指定されないと、すべてのファイルとディレクトリが抽出されます。
ステップ 4	more [/ascii /binary /ebcdic] /file-url 例 : device# more flash:/new-configs	リモートファイルシステム上のファイルを含めて、読み取り可能なファイルの内容を表示します。

フラッシュ ファイル システムに関するその他の関連資料

関連資料

関連項目	マニュアル タイトル
flash: ファイル システムの管理コマンド	<i>Cisco IOS Configuration Fundamentals Command Reference</i>

標準

標準	タイトル
この機能でサポートされる新規の標準または変更された標準はありません。また、既存の標準のサポートは変更されていません。	--

MIB

MIB	MIB のリンク
この機能によってサポートされる新しい MIB または変更された MIB はありません。またこの機能による既存 MIB のサポートに変更はありません。	選択したプラットフォーム、Cisco IOS リリース、およびフィーチャセットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。 http://www.cisco.com/go/mibs

RFC

RFC	タイトル
この機能によりサポートされた新規 RFC または改訂 RFC はありません。またこの機能による既存 RFC のサポートに変更はありません。	--

シスコのテクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	http://www.cisco.com/cisco/web/support/index.html

フラッシュファイルシステムの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	フラッシュファイルシステム	フラッシュファイルシステムは、ファイルを格納できる単一のフラッシュ デバイスです。ソフトウェア バンドルおよびコンフィギュレーション ファイルの管理に役立つ複数のコマンドも備えています。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 13 章

工場出荷時の状態へのリセット (Factory Reset)

- [初期設定へのリセット実行の前提条件 \(305 ページ\)](#)
- [工場出荷時の状態へのリセットの実行に関する制限 \(305 ページ\)](#)
- [初期設定へのリセットに関する情報 \(306 ページ\)](#)
- [工場出荷時の状態へのリセット方法 \(307 ページ\)](#)
- [工場出荷時の状態へのリセットに関する機能履歴と情報 \(307 ページ\)](#)

初期設定へのリセット実行の前提条件

- 工場出荷時設定へのリセット操作を実行する前に、すべてのソフトウェアイメージ、設定、および個人データがバックアップされていることを確認してください。
- 工場出荷時設定へのリセットはスタンドアロンモードでのみサポートされているため、デバイスがスタッキングモードになっていないことを確認してください。高可用性モードのモジュラーシャーシの場合、工場出荷時設定へのリセットはスーパーバイザごとに適用されます。
- プロセスが進行中の場合は、電源の中断がないことを確認してください。
- 工場出荷時の状態にリセットするプロセスを開始する前に、必ず現在のイメージのバックアップを作成してください。
- 工場出荷時の状態にリセットするプロセスを開始する前に、In-Service Software Upgrade (ISSU) または In-Service Software Downgrade (ISSD) が進行中でないことを確認します。

工場出荷時の状態へのリセットの実行に関する制限

- スイッチにインストールされているソフトウェアパッチは、工場出荷時設定へのリセット操作後に復元されません。

- vty セッションを通じて Factory Reset 操作後に復元されません。コマンドが発行された場合、Factory Reset プロセスの完了後にセッションは復元されません。

初期設定へのリセットに関する情報

工場出荷時にリセットでは、出荷時以降にデバイスに追加されたすべての顧客固有のデータを削除します。消去されるデータには、構成、ログファイル、ブート変数、コアファイル、FIPS 関連キーなどの資格情報が含まれます。

次の表に、工場出荷時設定へのリセットプロセス中に消去および保持されるデータの詳細を示します。

表 13: 初期設定へのリセット時に消去および保持されるデータ

消去されるデータ	保持されるデータ
現在のブートイメージを含むすべての Cisco IOS イメージ	リモート Field-Replaceable Unit (FRU) からのデータ
クラッシュ情報とログ	コンフィギュレーション レジスタの値
ユーザーデータ、スタートアップおよび実行コンフィギュレーション	USB の内容
FIPS 関連キーなどのクレデンシャル	Secure Unique Device Identifier (SUDI) 証明書、公開キーインフラストラクチャ (PKI) キーなどの認証情報
オンボード障害ロギング (OBFL) ログ	ライセンス
ユーザーが追加した ROMMON 変数	

工場出荷時設定へのリセットプロセスは、次の 2 つのシナリオで使用できます。

- デバイスの返品許可 (RMA) : RMA のためにデバイスをシスコに返送する必要がある場合は、そのデバイスの RMA 証明書を取得する前に、お客様固有のデータをすべて削除してください。
- 侵害を受けたデバイスのリカバリ : デバイ스에保存されているキーマテリアルまたはクレデンシャルが侵害を受けた場合は、デバイスを初期設定にリセットし、デバイスを再設定してください。

デバイスがリロードされ、工場出荷時設定へのリセットタスクが実行されます。このリロードにより ROMMON モードになることに注意してください。

工場出荷時設定へのリセット操作が完了したら、USB または TFTP 経由で Cisco IOS イメージをロードできます。

工場出荷時の状態へのリセット方法

手順

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 パスワードを入力します（要求された場合）。
ステップ 2	factory-reset {all config boot-vars} 例： デバイス# factory-reset all	すべてのオプションを有効にしてコマンドを使用してください。 factory reset コマンドを使用するために必要なシステム設定はありません。 NVRAM のすべての内容、現在のブートイメージ、ブート変数、起動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データ、およびユーザーデータを含むすべての Cisco IOS イメージを消去します。 オプション config を使用して、起動構成をリセットします。 オプション boot-vars を使用して、ユーザーが追加したブート変数をリセットします。 工場出荷時設定へのリセットプロセスが正常に完了すると、デバイスは再起動し、ROMMON モードで停止します。

工場出荷時の状態へのリセットに関する機能履歴と情報

リリース	機能情報
Cisco IOS XE Fuji 16.8.1a	この機能が導入されました。



第 14 章

セキュアストレージの設定

- [セキュアストレージについて \(309 ページ\)](#)
- [セキュアストレージの有効化 \(309 ページ\)](#)
- [セキュアストレージの無効化 \(310 ページ\)](#)
- [暗号化のステータスの確認 \(311 ページ\)](#)
- [セキュアストレージの機能情報 \(311 ページ\)](#)

セキュアストレージについて

セキュアストレージ機能では、重要な設定情報を暗号化して保護できます。非対称キーペア、事前共有秘密、タイプ 6 のパスワード暗号化キーおよび特定のクレデンシャルを暗号化します。インスタンス固有の暗号キーは、危険にさらされることを防ぐためにハードウェアのトラストアンカーに保管されます。

セキュアストレージの有効化

始める前に

この機能はデフォルトで無効になっています。

手順の概要

1. **configure terminal**
2. **service private-config-encryption**
3. **end**
4. **write memory**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# <code>configure terminal</code>	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	service private-config-encryption 例： デバイス (config)# <code>service private-config-encryption</code>	デバイスでセキュアストレージ機能を有効にします。
ステップ 3	end 例： デバイス (config)# <code>end</code>	特権 EXEC モードに戻ります。
ステップ 4	write memory 例： デバイス# <code>write memory</code>	private-config ファイルを暗号化し、暗号化フォーマットで保存します。

セキュアストレージの無効化

始める前に

デバイスでセキュアストレージ機能を無効にするには、次のタスクを実行します。

手順の概要

1. `configure terminal`
2. `no service private-config-encryption`
3. `end`
4. `write memory`

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	configure terminal 例： デバイス# <code>configure terminal</code>	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	no service private-config-encryption 例：	デバイスでセキュリティストレージ機能を無効にします。セキュアストレージを無効にすると、すべて

	コマンドまたはアクション	目的
	デバイス(config)# no service private-config-encryption	のユーザーデータがプレーンテキストでNVRAMに保存されます。
ステップ 3	end 例： デバイス(config)# end	特権 EXEC モードに戻ります。
ステップ 4	write memory 例： デバイス# write memory	private-config ファイルを復号し、プレーンフォーマットで保存します。

暗号化のステータスの確認

暗号化のステータスを確認するには、**show parser encrypt file status** コマンドを使用します。次のコマンド出力は、機能は利用できるが、ファイルが暗号化されていないことを示します。ファイルは「プレーンテキスト」形式です。

```
デバイス#show parser encrypt file status
Feature: Enabled
File Format: Plain Text
Encryption Version: Ver1
```

セキュアストレージの機能情報

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	セキュアなストレージ	セキュアストレージ機能では、重要な設定情報を暗号化して保護できます。非対称キーペア、事前共有秘密、タイプ 6 のパスワード暗号化キーおよび特定のクレデンシャルを暗号化します。インスタンス固有の暗号キーは、危険にさらされることを防ぐためにハードウェアのトラストアンカーに保管されます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 15 章

条件付きデバッグとラジオアクティブトレース

- 機能情報の確認 (313 ページ)
- 条件付きデバッグの概要 (313 ページ)
- ラジオアクティブトレースの概要 (314 ページ)
- 条件付きデバッグとラジオアクティブトレースの設定方法 (315 ページ)
- 条件付きデバッグのモニターリング (319 ページ)
- 条件付きデバッグの設定例 (319 ページ)
- 条件付きデバッグとラジオアクティブトレースに関するその他の関連資料 (320 ページ)
- 条件付きデバッグとラジオアクティブトレースの機能履歴 (321 ページ)

機能情報の確認

ご使用のソフトウェアリリースでは、このモジュールで説明されるすべての機能がサポートされているとは限りません。最新の機能情報および警告については、使用するプラットフォームおよびソフトウェアリリースの **Bug Search Tool** およびリリース ノートを参照してください。このモジュールに記載されている機能の詳細を検索し、各機能がサポートされているリリースのリストを確認する場合は、このモジュールの最後にある機能情報の表を参照してください。

プラットフォームのサポートおよびシスコソフトウェアイメージのサポートに関する情報を検索するには、**Cisco Feature Navigator** を使用します。**Cisco Feature Navigator** には、<http://www.cisco.com/go/cfn> からアクセスします。**Cisco.com** のアカウントは必要ありません。

条件付きデバッグの概要

条件付きデバッグ機能によって、定義した条件に基づき、特定の機能のデバッグおよびロギングを選択して有効にすることができます。この機能は、多くの機能がサポートされているシステムで有用です。



(注) コントロールプレーントレースのみがサポートされています。

条件付きデバッグでは、多数の機能が導入されていて大規模に稼働しているネットワークにおけるきめ細かなデバッグが可能です。これにより、システム内の細かなインスタンスに対しても、詳細なデバッグを実行できます。これは、何千ものセッションのうち特定のセッションのみをデバッグするような場合に、非常に有用です。条件は複数指定することもできます。

条件とは、機能またはアイデンティティをいいます。アイデンティティは、インターフェイス、IP アドレス、MAC アドレスなどです。



(注) サポートされる条件は MAC アドレスであることのみです。

これは、処理する機能オブジェクトを区別せずに出力を生成する、一般的なデバッグコマンドとは対照的です。一般的なデバッグコマンドは、多数のシステムリソースを消費し、システムパフォーマンスに影響します。

ラジオアクティブトレースの概要

ラジオアクティブトレースにより、冗長性のレベルを高めた状態で、システムの全体にわたって目的とする動作を連鎖的に実行できます。また、複数のスレッド、プロセス、および関数呼び出しにわたって、デバッグ情報を条件に基づいて (DEBUG レベルまで、または指定のレベルまで) 出力する方法を提供します。



(注) デフォルトのレベルは **DEBUG** です。ユーザーは別のレベルに変更することはできません。

ラジオアクティブトレースでは、次の機能が有効になっています。

- IGMP スヌーピング
- レイヤ 2 マルチキャスト

条件付きデバッグとラジオアクティブトレースの設定方法

条件付きデバッグおよび放射線トレース

条件付きデバッグと組み合わせた放射線トレースによって、条件に関連するすべての実行コンテキストをデバッグする単一のデバッグ CLI を取得できます。これは、ボックス内の機能のさまざまな制御フロープロセスを認識していなくても行うことができ、これらのプロセスでデバッグを個別に発行する必要もありません。

トレースファイルの場所

デフォルトでは、トレースファイル ログは各プロセスで生成され、`/tmp/rp/trace` または `/tmp/fp/trace` ディレクトリに保存されます。この一時ディレクトリで、トレースログがファイルに書き込まれます。各ファイルは 1 MB サイズです。このディレクトリでは、特定のプロセスのこうしたファイルを、最大 25 件保持できます。`/tmp` ディレクトリのトレースファイルがその 1 MB 制限またはブート時に設定されたサイズに達した場合、ローテーションから外れ、`tracelogs` ディレクトリの `/crashinfo` パーティションの下にあるアーカイブの場所に移動します。

`/tmp` ディレクトリが 1 つのプロセスで保持するトレースファイルは 1 つのみです。ファイルがそのファイルサイズの制限に達すると、ローテーションから外れ、`/crashinfo/tracelogs` に移動します。アーカイブ ディレクトリに蓄積されるファイルは最大 25 ファイルであり、その後は最も古いものから順に、`/tmp` から新たにローテーションされたファイルに置換されます。

`crashinfo` ディレクトリ内のトレースファイルは次の形式で配置されます。

1. Process-name_Process-ID_running-counter.timestamp.gz
例 : IOSRP_R0-0.bin_0.14239.20151101234827.gz
2. Process-name_pmanlog_Process-ID_running-counter.timestamp.bin.gz
例 : wcm_pmanlog_R0-0.30360_0.20151028233007.bin.gz

条件付きデバッグの設定

条件付デバッグを設定するには、以下の手順に従います。

手順の概要

1. **enable**
2. **debug platform condition mac** {mac-address}
3. **debug platform condition start**

4. **show platform condition** または **show debug**
5. **debug platform condition stop**
6. **request platform software trace archive** [last {number} days] [target {crashinfo: |flashinfo:}]
7. **show platform software trace** [filter-binary | level | message]
8. **clear platform condition all**

手順の詳細

	コマンドまたはアクション	目的
ステップ 1	enable 例： デバイス> enable	特権 EXEC モードを有効にします。 • パスワードを入力します（要求された場合）。
ステップ 2	debug platform condition mac {mac-address} 例： デバイス# debug platform condition mac bc16.6509.3314	指定された MAC アドレスの条件付きデバッグを設定します。
ステップ 3	debug platform condition start 例： デバイス# debug platform condition start	条件付きデバッグを開始します（上記のいずれかの条件に一致すると放射線トレースを開始します）。
ステップ 4	show platform condition または show debug 例： デバイス# show platform condition デバイス# show debug	現在設定されている条件を表示します。
ステップ 5	debug platform condition stop 例： デバイス# debug platform condition stop	条件付きデバッグを停止します（放射線トレースを停止します）。
ステップ 6	request platform software trace archive [last {number} days] [target {crashinfo: flashinfo:}] 例： デバイス# request platform software trace archive last 2 days	（任意）システムのマージされたトレースファイルの履歴ログを表示します。日数またはロケーションの組み合わせのフィルタ。
ステップ 7	show platform software trace [filter-binary level message] 例： デバイス# show platform software trace message	（任意）最新のトレースファイルからマージされたログを表示します。アプリケーションの状態、トレース モジュール名およびトレース レベルをさまざまな組み合わせでフィルタリングします。 • filter-binary : 照合するモジュールをフィルタリングします。

	コマンドまたはアクション	目的
		<ul style="list-style-type: none"> • level : トレース レベルを表示します。 • message : トレース メッセージのリングの内容を表示します。 <p>(注) デバイス上では次が可能です。</p> <ul style="list-style-type: none"> • Linux シェルだけでなく、IOS のコンソールからも使用できます。 • マージされたログでファイルを生成します。 • ステージング エリアからのみマージされたログを表示します。
ステップ 8	clear platform condition all 例 : デバイス# clear platform condition all	すべての条件をクリアします。

次のタスク



(注) **request platform software trace filter-binary** コマンドと **show platform software trace filter-binary** コマンドは同様の動作をします。唯一の違いは次のとおりです。

- **request platform software trace filter-binary** : データ ソースとして履歴ログを使用します。
- **show platform software trace filter-binary** : データ ソースとしてフラッシュの一時ディレクトリを使用します。

その中でも、`mac_log <..date.>` は、デバッグする MAC 用のメッセージを伝えるため、最も重要なファイルです。コマンド **show platform software trace filter-binary** も同じフラッシュ ファイルを生成し、また、画面に `mac_log` を出力します。

L2 マルチキャストの放射線トレース

特定のマルチキャスト受信者を特定するには、参加者または受信側クライアントの MAC アドレス、グループのマルチキャスト IP アドレスおよびスヌーピング VLAN を指定します。また、デバッグのトレース レベルを有効にします。デバッグ レベルでは、詳細なトレースとシステムへの高い可視性が提供されます。

debug platform condition feature multicast controlplane mac client MAC address ip Group IP address vlan id level debug level

トレース ファイルの推奨ワークフロー

トレース ファイルの推奨ワークフローの概要は次のとおりです。

1. 特定の時間帯のトレースログを要求する場合。
たとえば 1 日。
使用するコマンドは、次のとおりです。
デバイス# **request platform software trace archive last 1 day**
2. システムは、/flash: ロケーション内のトレースログの tar ball (.gz ファイル) を生成します。
3. スイッチ外にファイルをコピーします。ファイルをコピーすることによって、オフラインでトレースログが使用できます。ファイルのコピーについての詳細は、次のセクションを参照してください。
4. /flash: location からトレースログファイル (.gz) ファイルを削除します。これにより、他の操作に十分な領域がスイッチに確保されます。

ボックス外へのトレース ファイルのコピー

トレース ファイルの例を以下に示します。

```

デバイス# dir crashinfo:/tracelogs
Directory of crashinfo:/tracelogs/

50664 -rwx 760 Sep 22 2015 11:12:21 +00:00 plogd_F0-0.bin_0.gz
50603 -rwx 991 Sep 22 2015 11:12:08 +00:00 fed_pmanlog_F0-0.bin_0.9558.20150922111208.gz
50610 -rw- 11 Nov 2 2015 00:15:59 +00:00 timestamp
50611 -rwx 1443 Sep 22 2015 11:11:31 +00:00
auto_upgrade_client_sh_pmanlog_R0-.bin_0.3817.20150922111130.gz
50669 -rwx 589 Sep 30 2015 03:59:04 +00:00 cfgwr-8021_R0-0.bin_0.gz
50612 -rwx 1136 Sep 22 2015 11:11:46 +00:00 reflector_803_R0-0.bin_0.1312.20150922111116.gz
50794 -rwx 4239 Nov 2 2015 00:04:32 +00:00 IOSRP_R0-0.bin_0.14239.20151101234827.gz
50615 -rwx 131072 Nov 2 2015 00:19:59 +00:00 linux_iosd_image_pmanlog_R0-0.bin_0
--More--

```

トレース ファイルは、次に示すさまざまなオプションのいずれかを使用して、コピーできます。

```

デバイス# copy crashinfo:/tracelogs ?
crashinfo: Copy to crashinfo: file system
flash: Copy to flash: file system
ftp: Copy to ftp: file system
http: Copy to http: file system
https: Copy to https: file system
null: Copy to null: file system
nvram: Copy to nvram: file system
rcp: Copy to rcp: file system
running-config Update (merge with) current system configuration
scp: Copy to scp: file system
startup-config Copy to startup configuration

```

```
syslog: Copy to syslog: file system
system: Copy to system: file system
tftp: Copy to tftp: file system
tmpsys: Copy to tmpsys: file system
```

TFTP サーバーにコピーするための一般的な構文は次のとおりです。

```
デバイス# copy source: tftp:
デバイス# copy crashinfo:/tracelogs/IOSRP_R0-0.bin_0.14239.20151101234827.gz tftp:
Address or name of remote host []? 2.2.2.2
Destination filename [IOSRP_R0-0.bin_0.14239.20151101234827.gz]?
```



(注) tracelog および他の目的に使用可能な空き容量があることを確認するために、生成されたレポート/アーカイブ ファイルをスイッチからクリアすることが重要です。

条件付きデバッグのモニターリング

以下の表に、条件付きデバッグのモニターに使用できる各種コマンドを示します。

コマンド	目的
show platform condition	現在設定されている条件を表示します。
show debug	現在設定されているデバッグ条件を表示します。
show platform software trace filter-binary	最新のトレース ファイルからマージされたログを表示します。
request platform software trace filter-binary	システムにマージされたトレース ファイルの履歴ログを表示します。

条件付きデバッグの設定例

次に、`show platform condition` コマンドの出力例を示します。

```
デバイス# show platform condition
Conditional Debug Global State: Stop
Conditions Direction
```

```
-----|-----
MAC Address 0024.D7C7.0054 N/A
Feature Condition Type Value
```

```
-----|-----
デバイス#
```

次に、`show debug` コマンドの出力例を示します。

```
デバイス# show debug
IOSXE Conditional Debug Configs:
```

```
Conditional Debug Global State: Start
Conditions Direction
```

```
-----|-----
MAC Address 0024.D7C7.0054 N/A
Feature Condition Type Value
```

```
-----|-----
Packet Infra debugs:
Ip Address Port
```

```
-----|-----
デバイス#
```

次に、`debug platform condition stop` コマンドの例を示します。

```
デバイス# debug platform condition stop
Conditional Debug Global State: Stop
```

条件付きデバッグとラジオアクティブトレースに関するその他の関連資料

関連資料

関連項目	マニュアルタイトル
この章で使用するコマンドの完全な構文および使用方法の詳細。	<i>Command Reference (Catalyst 9300 Series Switches)</i>

MIB

MIB	MIB のリンク
本リリースでサポートするすべての MIB	<p>選択したプラットフォーム、Cisco IOS リリース、およびフィチャセットに関する MIB を探してダウンロードするには、次の URL にある Cisco MIB Locator を使用します。</p> <p>http://www.cisco.com/go/mibs</p>

シスコのテクニカル サポート

説明	リンク
<p>シスコのサポート Web サイトでは、シスコの製品やテクノロジーに関するトラブルシューティングにお役立ていただけるように、マニュアルやツールをはじめとする豊富なオンラインリソースを提供しています。</p> <p>お使いの製品のセキュリティ情報や技術情報を入手するために、Cisco Notification Service (Field Notice からアクセス)、Cisco Technical Services Newsletter、Really Simple Syndication (RSS) フィードなどの各種サービスに加入できます。</p> <p>シスコのサポート Web サイトのツールにアクセスする際は、Cisco.com のユーザ ID およびパスワードが必要です。</p>	http://www.cisco.com/support

条件付きデバッグとラジオアクティブトレースの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	条件付きデバッグとラジオアクティブトレース	条件付きデバッグ機能によって、定義した条件に基づき、特定の機能のデバッグおよびロギングを選択して有効にすることができます。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。



第 16 章

ソフトウェア設定のトラブルシューティング

この章では、スイッチが稼働する Cisco IOS ソフトウェアに関連する問題を特定し、解決する方法について説明します。問題の性質に応じて、コマンドラインインターフェイス (CLI)、デバイス マネージャ、または Network Assistant を使用して、問題を特定し解決できます。

LEDの説明など、トラブルシューティングの詳細については、ハードウェアインストールガイドを参照してください。

- [ソフトウェア設定のトラブルシューティングに関する情報 \(323 ページ\)](#)
- [ソフトウェア設定のトラブルシューティング方法 \(332 ページ\)](#)
- [ソフトウェア設定のトラブルシューティングの確認 \(347 ページ\)](#)
- [ソフトウェア設定のトラブルシューティングのシナリオ \(349 ページ\)](#)
- [ソフトウェアのトラブルシューティングの設定例 \(354 ページ\)](#)
- [ソフトウェア設定のトラブルシューティングの機能履歴 \(356 ページ\)](#)

ソフトウェア設定のトラブルシューティングに関する情報

スイッチのソフトウェア障害

スイッチソフトウェアがアップグレード中に破損する原因として、誤ったファイルがスイッチにダウンロードされた場合やイメージファイルが削除された場合があります。これらのどの場合も、スイッチは、電源投入時自己診断テスト (POST) に合格せず、接続はありません。ソフトウェア障害から回復するには、[ソフトウェア障害からの回復 \(332 ページ\)](#) の項で説明されている手順に従います。

のパスワードを紛失したか忘れた場合 デバイス

deviceのデフォルト設定では、deviceに物理的にアクセスしているエンドエンド ユーザーは、スイッチの電源投入中に起動プロセスを中断して新しいパスワードを入力することにより、パスワードを紛失した状態から回復できます。ここで紹介する回復手順を実行するには、deviceに物理的にアクセスする必要があります。



(注) これらのdevicesでは、システム管理者は、デフォルト設定に戻すことに同意した場合に限り、エンドユーザーによるパスワードのリセットを許可することによって、この機能の一部を無効化できます。パスワード回復がディセーブルになっている場合に、エンドユーザーがパスワードをリセットしようとする、ステータスメッセージで回復プロセスの間はデフォルトの設定に戻すように指示されます。



(注) Cisco WLC の設定を複数の Cisco WLC 間でコピーすると、暗号化パスワード キーを回復できなくなります (RMA の場合)。

パスワードを紛失または忘れた場合にそのパスワードを回復するには、[パスワードを忘れた場合の回復 \(336 ページ\)](#) の項で説明する手順に従います。

Power over Ethernet (PoE) ポート

Power over Ethernet (PoE) スイッチポートでは、回路に電力が供給されていないことをスイッチが検知した場合、接続している次のデバイスに電力が自動的に供給されます。

- シスコ先行標準受電デバイス (Cisco IP Phone や Cisco Aironet アクセス ポイントなど)
- IEEE 802.3af 準拠の受電装置
- IEEE 802.3at 準拠の受電装置

受電デバイスが PoE スイッチポートおよび AC 電源に接続されている場合、冗長電力として利用できます。受電デバイスが PoE ポートにだけ接続されている場合、受電デバイスには冗長電力は供給されません。

受電デバイスを検出すると、スイッチは受電デバイスの電力要件を判断し、受電デバイスへの電力供給を許可または拒否します。また、スイッチは消費電力をモニタリングおよびポリシングすることで、装置の電力の消費をリアルタイムに検知できます。

詳細については、『*Interface and Hardware Component Configuration Guide (Catalyst 9300 Switches)*』の「Configuring PoE」の章を参照してください。

PoE のさまざまなトラブルシューティング シナリオについては、[Power over Ethernet \(PoE\) に関するトラブルシューティングのシナリオ \(349 ページ\)](#) の項を参照してください。

電力消失によるポートの障害

PoE デバイスポートに接続され、AC 電源から電力が供給されている受電デバイス（Cisco IP Phone 7910 など）に AC 電源から電力が供給されない場合、そのデバイスは **errdisable** ステートになることがあります。**errdisable** ステートから回復するには、**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から、**no shutdown** インターフェイス コマンドを入力します。デバイスで自動回復を設定し、**errdisable** ステートから回復することもできます。

デバイスの場合、**errdisable recovery cause loopback** および **errdisable recovery interval seconds** グローバル コンフィギュレーション コマンドは、指定した期間が経過したあと自動的にインターフェイスを **errdisable** ステートから復帰させます。

不正リンク アップによるポート障害

シスコ受電デバイスをポートに接続し、**power inline never** インターフェイス コンフィギュレーション コマンドを使用してポートを設定した場合は、不正リンクアップが発生し、ポートが **errdisable** ステートになることがあります。ポートを **errdisable** ステートから回復するには、**shutdown** および **no shutdown** インターフェイス コンフィギュレーション コマンドを入力します。

power inline never コマンドで設定したポートにシスコ受電デバイスを接続しないでください。

ping

デバイスは IP の ping をサポートしており、これを使用してリモートホストへの接続をテストできます。ping はアドレスにエコー要求パケットを送信し、応答を待ちます。ping は次のいずれかの応答を返します。

- 正常な応答：正常な応答（*hostname* が存在する）は、ネットワーク トラフィックにもよりますが、1 ~ 10 秒以内で発生します。
- 宛先の応答なし：ホストが応答しない場合、*no-answer* メッセージが返されます。
- 不明なホスト：ホストが存在しない場合、*unknown host* メッセージが返されます。
- 宛先到達不能：デフォルト ゲートウェイが指定されたネットワークに到達できない場合、*destination-unreachable* メッセージが返されます。
- ネットワークまたはホストへの到達不能：ルートテーブルにホストまたはネットワークのエントリがない場合、*network or host unreachable* メッセージが返されます。

ping の動作を理解するには、[ping の実行（343 ページ）](#) の項を参照してください。

レイヤ 2 トレースルート

レイヤ 2 トレースルート機能により、パケットが通過する送信元デバイスから宛先デバイスまでの物理パスを識別できます。レイヤ 2 トレースルートは、ユニキャストの送信元および宛先 MAC アドレスだけをサポートします。**transroute** は、パス内にあるデバイスの MAC アドレス テーブルを使用してパスを識別します。デバイスがレイヤ 2 パス内でレイヤ 2 トレースルート

をサポートしていないデバイスを検知した場合、デバイスはレイヤ2トレースクエリを送信し続け、タイムアウトにします。

デバイスは、送信元デバイスから宛先デバイスへのパスのみを識別できます。パケットが通過する、送信元ホストから送信元デバイスまで、または宛先デバイスから宛先ホストまでのパスは識別できません。

レイヤ2の traceroute のガイドライン

- ネットワーク内のすべてのデバイスで、Cisco Discovery Protocol (CDP) をイネーブルにする必要があります。レイヤ2 traceroute が適切に動作するために、CDP を無効にしないでください。

物理パス内のデバイスが CDP に対して透過的な場合、スイッチはこれらのデバイスを通過するパスを識別できません。

- ping 特権 EXEC コマンドを使用して接続をテストできれば、このデバイスは別のデバイスから到達可能であると定義できます。物理パス内のすべてのデバイスは、他のデバイスから相互に到達可能でなければなりません。
- パス内で識別可能な最大ホップ数は 10 です。
- 送信元デバイスと宛先デバイス間の物理パス内にないデバイスで、**traceroute mac** または **traceroute mac ip** の特権 EXEC コマンドを実行できます。パス内のすべてのデバイスは、このスイッチから到達可能でなければなりません。
- 指定された送信元および宛先アドレスが同じ VLAN にある場合、**traceroute mac** コマンド出力はレイヤ2パスを表示します。指定した送信元および宛先 MAC アドレスが、それぞれ異なる VLAN に属している場合は、レイヤ2パスは識別されず、エラーメッセージが表示されます。
- マルチキャストの送信元または宛先 MAC アドレスを指定すると、パスは識別されず、エラーメッセージが表示されます。
- 送信元または宛先 MAC アドレスが複数の VLAN に属する場合は、送信元および宛先 MAC アドレスの両方が属している VLAN を指定する必要があります。VLAN を指定しないと、パスは識別されず、エラーメッセージが表示されます。
- 指定された送信元および宛先の IP アドレスが同一のサブネット内にある場合、**traceroute mac ip** コマンド出力はレイヤ2パスを表示します。IP アドレスを指定した場合、デバイスは Address Resolution Protocol (ARP) を使用し、IP アドレスとそれに対応する MAC アドレスおよび VLAN ID を対応させます。
 - 指定の IP アドレスの ARP のエントリが存在している場合、デバイスは関連付けられた MAC アドレスを使用し、物理パスを識別します。
 - ARP のエントリが存在しない場合、デバイスは ARP クエリを送信し、IP アドレスを解決しようと試みます。IP アドレスが解決されない場合は、パスは識別されず、エラーメッセージが表示されます。

- 複数のデバイスがハブを介して1つのポートに接続されている場合（たとえば複数の CDP ネイバーがポートで検出された場合）、レイヤ 2 **traceroute** 機能はサポートされません。複数の CDP ネイバーが1つのポートで検出された場合、レイヤ 2 パスは特定されず、エラーメッセージが表示されます。
- この機能は、トークンリング VLAN ではサポートされません。
- レイヤ 2 トレースルートは、ユーザデータグラム プロトコル (UDP) ポート 2228 でリスニングソケットを開きます。このポートは、任意の IPv4 アドレスを使用してリモートからアクセスでき、認証は必要ありません。この UDP ソケットにより、VLAN 情報、リンク、特定の MAC アドレスの存在、および CDP ネイバー情報をデバイスから読み取ることができます。この情報を使用することにより、最終的にレイヤ 2 ネットワークトポロジの全体像を構築できます。
- レイヤ 2 トレースルートはデフォルトで有効になっており、グローバルコンフィギュレーションモードで **no l2 traceroute** コマンドを実行することによって無効にできます。レイヤ 2 トレースルートを再度有効にするには、グローバル コンフィギュレーションモードで **l2 traceroute** コマンドを使用します。

IP トレースルート

IP **traceroute** を使用すると、ネットワーク上でパケットが通過するパスをホップバイホップで識別できます。このコマンドを実行すると、トラフィックが宛先に到達するまでに通過するルータなどのすべてのネットワーク層 (レイヤ 3) デバイスが表示されます。

デバイスは、**traceroute** 特権 EXEC コマンドの送信元または宛先として指定できます。また、**traceroute** コマンドの出力でホップとして表示される場合があります。デバイスを **traceroute** の宛先とすると、スイッチは、**traceroute** の出力で最終の宛先として表示されます。中間デバイスが同じ VLAN 内でポート間のパケットのブリッジングだけを行う場合、**traceroute** の出力に中間スイッチは表示されません。ただし、中間デバイスが特定の packets をルーティングするマルチレイヤ デバイスの場合、このデバイスは **traceroute** の出力にホップとして表示されます。

traceroute 特権 EXEC コマンドは、IP ヘッダーの存続可能時間 (TTL) フィールドを使用して、ルータおよびサーバで特定のリターンメッセージが生成されるようにします。**traceroute** の実行は、ユーザデータグラム プロトコル (UDP) データグラムを、TTL フィールドが 1 に設定されている宛先ホストへ送信することから始まります。ルータで TTL 値が 1 または 0 であることを検出すると、データグラムをドロップし、インターネット制御メッセージプロトコル (ICMP) **time-to-live-exceeded** メッセージを送信元に送信します。**traceroute** は、ICMP **time-to-live-exceeded** メッセージの送信元アドレスフィールドを調べて、最初のホップのアドレスを判別します。

ネクストホップを識別するために、**traceroute** は TTL 値が 2 の UDP パケットを送信します。1 番目のルータは、TTL フィールドの値から 1 を差し引いて次のルータにデータグラムを送信します。2 番目のルータは、TTL 値が 1 であることを確認すると、このデータグラムを廃棄し、**time-to-live-exceeded** メッセージを送信元へ返します。このように、データグラムが宛先ホスト

に到達するまで（または TTL の最大値に達するまで）TTL の値は増分され、処理が続けられます。

データグラムが宛先に到達したことを学習するために、`tracert` は、データグラムの UDP 宛先ポート番号を、宛先ホストが使用する可能性のない大きな値に設定します。ホストが、ローカルで使用されない宛先ポート番号を持つ自分自身宛てのデータグラムを受信すると、送信元に ICMP ポート到達不能エラーを送信します。ポート到達不能エラーを除くすべてのエラーは中間ホップから送信されるため、ポート到達不能エラーを受信するということは、このメッセージが宛先ポートから送信されたことを意味します。

例：IP ホストに対する `tracert` の実行（355 ページ）に進み、IP `tracert` プロセスの例を参照してください。

debug コマンド



注意 デバッグ出力は CPU プロセスで高プライオリティが割り当てられているため、デバッグ出力を行うとシステムが使用できなくなることがあります。したがって、`debug` コマンドを使用するのは、特定の問題のトラブルシューティング時、またはシスコのテクニカルサポート担当者とともにトラブルシューティングを行う場合に限定してください。ネットワークトラフィック量やユーザ数が少ない期間に `debug` コマンドを使用することをお勧めします。デバッグをこのような時間帯に行うと、`debug` コマンド処理のオーバーヘッドの増加によりシステムの使用に影響が及ぶ可能性が低くなります。

`debug` コマンドはすべて特権 EXEC モードで実行します。ほとんどの `debug` コマンドは引数を取りません。

システム レポート

システム レポートまたは `crashinfo` ファイルには、シスコのテクニカルサポート担当者が Cisco IOS イメージの障害（クラッシュ）が原因で起きた問題をデバッグするときに使用する情報が保存されています。明瞭度と整合性の高い重要なクラッシュ情報を迅速かつ確実に収集することが必要です。さらに、この情報の収集とバンドルが、特定のクラッシュの発生に対し関連付けが特定ができるような方法で行われることが必要です。

システム レポートは次の状況で生成されます。

- スイッチ障害の場合：システム レポートは障害が発生したメンバーで生成されます。スタック内の他のメンバーではレポートは生成されません。
- スイッチオーバーの場合：システム レポートはハイアベイラビリティ（HA）のメンバースイッチでのみ生成されます。非 HA メンバーについてはレポートは生成されません。

リロード時はレポートは生成されません。

クラッシュ プロセス時は、次の情報がスイッチからローカルに収集されます。

1. 完全なプロセス core

2. トレースログ
3. IOS の syslog (非アクティブなクラッシュの場合には保証されません)
4. システムプロセス情報
5. ブートアップログ
6. リロードログ
7. 特定のタイプの /proc 情報

この情報は個別のファイルに格納されてから、アーカイブされて1つのバンドルに圧縮されます。これにより、クラッシュのスナップショットを1つの場所で取得して、分析のためにボックス外に移動できるようになります。このレポートは、スイッチが ROMmon/ブートローダにダウンロードする前に生成されます。

完全な core およびトレースログ以外はテキスト ファイルです。

コアダンプを生成するには、**request platform software process core fed switch active** コマンドを使用します。

```
Device# request platform software process core fed switch active
SUCCESS: Core file generated.
```

```
Device# dir bootflash:/core
Directory of bootflash:/core/
16430  -rw-          10941657   Apr 6 2022 00:15:20 +00:00
Switch_1_RP_0_fed_18469_20220406-001511-UTC.core.gz
16812  -rw-           1   Apr 6 2022 00:01:48 +00:00 .callhome
16810  drwx           4096   Jan 18 2022 21:10:35 +00:00 modules
```

crashinfo ファイル

デフォルトでは、生成されたシステム レポート ファイルは /crashinfo ディレクトリに格納されます。Ifit は、領域不足のため crashinfo パーティションに保存できません。そのため、/flash ディレクトリに保存されます。

ファイルを表示するには、**dir crashinfo:** コマンドを入力します。次に crashinfo ディレクトリの出力例を示します。

```
Device# dir crashinfo:
Directory of crashinfo:/

23665 drwx 86016 Jun 9 2017 07:47:51 -07:00 tracelogs
11 -rw- 0 May 26 2017 15:32:44 -07:00 koops.dat
12 -rw- 4782675 May 29 2017 15:47:16 -07:00 system-report_1_20170529-154715-PDT.tar.gz
1651507200 bytes total (1519386624 bytes free)
```

システムレポートは、次の形式で crashinfo ディレクトリにあります。

```
system-report_[switch number]_[date]-[timestamp]-UTC.gz
```

スイッチがクラッシュしたら、システムレポートファイルを確認します。最後に生成されたシステムレポートファイルは crashinfo ディレクトリの下に last_systemreport というファイル名で保存されます。問題のトラブルシューティングを行う際、システム レポートおよび crashinfo ファイルが TAC の役に立ちます。

生成されたシステム レポートは、TFTP や HTTP などいくつかのオプションを使用して、さらにコピーできます。

```
Device# copy crashinfo: ?
crashinfo: Copy to crashinfo: file system
flash: Copy to flash: file system
ftp: Copy to ftp: file system
http: Copy to http: file system
https: Copy to https: file system
null: Copy to null: file system
nvram: Copy to nvram: file system
rcp: Copy to rcp: file system
running-config Update (merge with) current system configuration
scp: Copy to scp: file system
startup-config Copy to startup configuration
syslog: Copy to syslog: file system
system: Copy to system: file system
tftp: Copy to tftp: file system
tmpsys: Copy to tmpsys: file system
```

TFTP サーバーにコピーするための一般的な構文は次のとおりです。

```
Device# copy crashinfo: tftp:
Source filename [system-report_1_20150909-092728-UTC.gz]?
Address or name of remote host []? 1.1.1.1
Destination filename [system-report_1_20150909-092728-UTC.gz]?
```

スタックの全メンバーからのトレースログは、**trace archive** コマンドを発行することで収集できます。このコマンドには、時間帯オプションがあります。コマンド構文は次のとおりです。

```
Device# request platform software trace archive ?
last Archive trace files of last x days
target Location and name for the archive file
```

crashinfo: または **flash:** ディレクトリに格納されている過去 3650 日以内のトレースログが取得できます。

```
Device# request platform software trace archive last ?
<1-3650> Number of days (1-3650)
Switch#request platform software trace archive last 3650 days target ?
crashinfo: Archive file name and location
flash: Archive file name and location
```



(注) 一度コピーされたら、システム レポートやトレースのアーカイブを **flash** ディレクトリまたは **crashinfo** ディレクトリからクリアし、トレースログやその他の目的に使用できる領域を確保することが重要です。

スイッチのオンボード障害ロギング

オンボード障害ロギング (OBFL) 機能を使用すれば、デバイスに関する情報を収集できます。この情報には稼働時間、温度、電圧などの情報が含まれており、シスコのテクニカルサポート担当者がデバイスの問題をトラブルシューティングする際に役立ちます。OBFL はイネーブルにしておき、フラッシュメモリに保存されたデータは消さないようにすることを推奨します。

OBFLは、デフォルトでイネーブルになっています。デバイスおよびSmall Form-Factor Pluggable (SFP) モジュールに関する情報が収集されます。デバイスは、次の情報をフラッシュメモリに保存します。

- CLI コマンド：スタンドアロンデバイスまたはスイッチスタックメンバに入力されたOBFL CLI コマンドの記録。
- 環境データ：スタンドアロン デバイス またはスイッチスタックメンバおよび接続されているすべてのFRUデバイスの一意のデバイス ID (UDI) 情報、製品 ID (PID)、バージョン ID (VID)、およびシリアル番号。
- メッセージ：スタンドアロンデバイス またはスイッチスタックメンバにより生成されたハードウェア関連のシステムメッセージの記録。
- Power over Ethernet (PoE)：スタンドアロン デバイス またはスイッチスタックメンバの PoE ポートの消費電力の記録。
- 温度：スタンドアロン デバイス またはスイッチスタックメンバの温度。
- 稼働時間：スタンドアロン デバイス またはスイッチスタックメンバが起動された際の時刻、デバイス が再起動された理由、および デバイス が最後に再起動されて以来の稼働時間。
- 電圧：スタンドアロン デバイス またはスイッチスタックメンバのシステム電圧。

システム時計は、手動で時刻を設定するか、またはネットワーク タイム プロトコル (NTP) を使用するように設定します。

デバイスの稼働中には、**show logging onboard** 特権 EXEC コマンドを使用することにより、OBFL データを取得できます。デバイス に障害が発生した場合のデータの取得方法については、お客様担当のシスコテクニカルサポート担当者にお問い合わせください。

OBFL がイネーブルになっている デバイス が再起動された場合、新しいデータの記録が開始するまでに 10 分間の遅延があります。

ファン障害

デフォルトでは、この機能はディセーブルです。現場交換可能ユニット (FRU) または電源装置の複数のファンが故障した場合、デバイスはシャットダウンせず、次のようなエラーメッセージが表示されます。

```
Multiple fan(FRU/PS) failure detected. System may get overheated. Change fan quickly.
```

デバイスが過熱状態となり、シャットダウンすることもあります。

ファン障害機能をイネーブルにするには、**system env fan-fail-action shut** 特権 EXEC コマンドを入力します。デバイス内の複数のファンに障害が発生した場合、デバイスは自動的にシャットダウンし、次のようなエラーメッセージが表示されます。

```
Faulty (FRU/PS) fans detected, shutting down system!
```

最初のファンの停止後、デバイスが2つめのファンの障害を検知すると、デバイスは20秒待機してからシャットダウンします。

デバイスを再起動するには、電源をオフにしてから再度オンにする必要があります。

CPU 使用率が高い場合に起こりうる症状

CPU 使用率が高すぎることで次の現象が発生する可能性があります。他の原因で発生する場合もあります。次にその一部を示します。

- スパニングツリー トポロジの変更
- 通信が切断されたために EtherChannel リンクがダウンした
- 管理要求 (ICMP ping、SNMP のタイムアウト、低速な Telnet または SSH セッション) に応答できない
- UDLD フラッピング
- SLA の応答が許容可能なしきい値を超えたことによる IP SLA の失敗
- スイッチが要求を転送しない、または要求に応答しない場合の DHCP または IEEE 802.1x の処理の失敗

ソフトウェア設定のトラブルシューティング方法

ソフトウェア障害からの回復

始める前に

ここで紹介する回復手順を実行するには、スイッチを直接操作する必要があります。

ここで紹介する手順では、破損したイメージファイルまたは不適切なイメージファイルの回復に `boot loader` コマンドおよび TFTP を使用します。

スイッチのコンソールポートのデフォルトレートである 9600 ビット/秒 (bps) と一致するように、端末のボーレートを設定します。ボーレートが 9600 bps 以外の値に設定されている場合、速度がデフォルトに戻るまでコンソールへのアクセスは失われます。

ステップ 1 PC 上で、Cisco.com からソフトウェア イメージファイル (`image.bin`) をダウンロードします。

ステップ 2 TFTP サーバーにソフトウェア イメージをロードします。

ステップ 3 PC をスイッチのイーサネット管理ポートに接続します。

ステップ 4 スイッチの電源コードを取り外します。

ステップ5 [Mode] ボタンを押しながら、電源コードをスイッチに再接続します。

ステップ6 ブートローダープロンプトで、TFTP サーバーに ping を実行できることを確認します。

- a) スwitchの IP アドレスを設定します：**set IP_ADDRESS ip_address**

例：

```
switch: set IP_ADDRESS 192.0.2.123
```

- b) スwitchのサブネットマスクを設定します：**set IP_SUBNET_MASK subnet_mask**

例：

```
switch: set IP_SUBNET_MASK 255.255.255.0
```

- c) デフォルトゲートウェイを設定します：**set DEFAULT_GATEWAY ip_address**

例：

```
switch: set DEFAULT_ROUTER 192.0.2.1
```

- d) 次のコマンドを実行して、TFTP サーバーに ping を実行できることを確認します。**switch: ping ip_address_of_TFTP_server**

例：

```
switch: ping 192.0.2.15
ping 192.0.2.1 with 32 bytes of data...
Host 192.0.2.1 is alive.
switch:
```

ステップ7 次のいずれかを選択します。

- ブートローダープロンプトで、**boot tftp** コマンドを開始します。これにより、スイッチでソフトウェアイメージを容易に回復できます。

```
switch: boot tftp://10.168.0.1/cat9k/cat9k_iosxe.2017-08-25_09.41.bin
attempting to boot from [tftp://10.168.0.1/cat9k/cat9k_iosxe.2017-08-25_09.41.SSA.bin]

interface : eth0
macaddr   : E4:AA:5D:59:7B:44
ip        : 10.168.247.10
netmask   : 10.255.0.0
gateway   : 10.168.0.1
server    : 10.168.0.1
file      : cat9k/cat9k_iosxe.2017-08-25_09.41.bin
```

Restricted Rights Legend

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c) of the Commercial Computer Software - Restricted Rights clause at FAR sec. 52.227-19 and subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS sec. 252.227-7013.

cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software [Everest], Catalyst L3 Switch Software (CAT9K_IOSXE), Version 16.6.1 RELEASE SOFTWARE (fc2)
Copyright (c) 1986-2017 by Cisco Systems, Inc.
Compiled Thu 24-Aug-17 13:23 by mcpre

Cisco IOS-XE software, Copyright (c) 2005-2017 by cisco Systems, Inc.
All rights reserved. Certain components of Cisco IOS-XE software are licensed under the GNU General Public License ("GPL") Version 2.0. The software code licensed under GPL Version 2.0 is free software that comes with ABSOLUTELY NO WARRANTY. You can redistribute and/or modify such GPL code under the terms of GPL Version 2.0. For more details, see the documentation or "License Notice" file accompanying the IOS-XE software, or the applicable URL provided on the flyer accompanying the IOS-XE software.

FIPS: Flash Key Check : Begin
FIPS: Flash Key Check : End, Not Found, FIPS Mode Not Enabled

This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
<http://www.cisco.com/wwl/export/crypto/tool/stqrg.html>

If you require further assistance please contact us by sending email to export@cisco.com.

cisco C9XXX (X86) processor (revision V00) with 869398K/6147K bytes of memory.
Processor board ID FXS1939Q3LZ
144 Gigabit Ethernet interfaces
16 Ten Gigabit Ethernet interfaces
4 Forty Gigabit Ethernet interfaces
32768K bytes of non-volatile configuration memory.
15958516K bytes of physical memory.
11161600K bytes of Bootflash at bootflash:.
1638400K bytes of Crash Files at crashinfo:.
0K bytes of WebUI ODM Files at webui:.

%INIT: waited 0 seconds for NVRAM to be available

Press RETURN to get started!

- リカバリパーティションからソフトウェアをインストールします。この回復イメージは、`emergency-install` 機能を使用して回復を実施する場合に必要となります。

- a) 回復パーティション (`sda9`) に回復イメージが存在することを確認します。

例 :

```
switch: dir sda9:
```

```
Size           Attributes      Name
-----
21680202      -rw-           cat9k-recovery.SSA.bin
-----
```

- b) ブートローダープロンプトで、`emergency-install` 機能を開始します。この機能を使用すると、スイッチでソフトウェアイメージを容易に回復できます。警告 : `emergency-install` コマンドを実行すると、ブートブッシュ全体が消去されます。

例 :

```
switch: emergency-install tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin
WARNING: The system partition (bootflash:) will be erased during the system recovery install
process.
Are you sure you want to proceed? [y] y/n [n]: y
Starting system recovery (tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin)
...
Attempting to boot from [sda9:cat9k-recovery.SSA.bin]
Located cat9k-recovery.SSA.bin
#####

Warning: ignoring ROMMON var "BOOT_PARAM"

PLATFORM_TYPE C9X00 speed 9600

Booting Recovery Image 16.5.1a

Initiating Emergency Installation of bundle
tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin

Downloading bundle tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
curl_vrf=2
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
 100 485M  100 485M    0     0  5143k      0  0:01:36  0:01:36  ---:--:-- 5256k
 100 485M  100 485M    0     0  5143k      0  0:01:36  0:01:36  ---:--:-- 5143k

Validating bundle tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Installing bundle tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Verifying bundle tftp://10.255.254.254/auto/tftpboot/X86/cat9k_iosxe.16.05.01a.SPA.bin...
Package cat9k-cc_srdriver.16.05.01a.SPA.pkg /temp//stage/cat9k-cc_srdriver.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-espbase.16.05.01a.SPA.pkg /temp//stage/cat9k-espbase.16.05.01a.SPA.pkg is Digitally
Signed
Package cat9k-guestshell.16.05.01a.SPA.pkg /temp//stage/cat9k-guestshell.16.05.01a.SPA.pkg is
Digitally Signed
Package cat9k-rpbase.16.05.01a.SPA.pkg /temp//stage/cat9k-rpbase.16.05.01a.SPA.pkg is Digitally
Signed
Package cat9k-sipbase.16.05.01a.SPA.pkg /temp//stage/cat9k-sipbase.16.05.01a.SPA.pkg is Digitally
Signed
```

```

Package cat9k-sipspace.16.05.01a.SPA.pkg /temp//stage/cat9k-sipspace.16.05.01a.SPA.pkg is Digitally Signed
Package cat9k-srdriver.16.05.01a.SPA.pkg /temp//stage/cat9k-srdriver.16.05.01a.SPA.pkg is Digitally Signed
Package cat9k-webui.16.05.01a.SPA.pkg /temp//stage/cat9k-webui.16.05.01a.SPA.pkg is Digitally Signed
Package cat9k-wlc.16.05.01a.SPA.pkg /temp//stage/cat9k-wlc.16.05.01a.SPA.pkg is Digitally Signed
Package /cat9k-rpboot.16.05.01a.SPA.pkg /temp//rpboot/cat9k-rpboot.16.05.01a.SPA.pkg is Digitally Signed
Preparing flash....
Flash filesystem unmounted successfully /dev/sdb3
Syncing device....
Emergency Install successful... Rebooting
Will reboot now

Initializing Hardware...

System Bootstrap, Version 16.5.2r, RELEASE SOFTWARE (P)
Compiled Wed 05/31/2017 15:58:35.22 by rel

Current image running:
Primary Rommon Image

Last reset cause: SoftwareReload
C9X00 platform with 8388608 Kbytes of main memory

```

あるいは、Telnet または管理ポートを通じて TFTP からローカルフラッシュにイメージをコピーした後、ローカルフラッシュからデバイスをブートします。

パスワードを忘れた場合の回復

スイッチのデフォルト設定では、スイッチを直接操作するエンドユーザが、スイッチの電源投入時に起動プロセスを中断して新しいパスワードを入力することにより、パスワードを紛失した状態から回復できます。ここで紹介する回復手順を実行するには、スイッチを直接操作してください。



- (注) これらのスイッチでは、システム管理者はデフォルト設定に戻す場合に限りエンドユーザによるパスワードのリセットを許可することによって、この機能の一部をディセーブルにできません。パスワード回復がディセーブルになっている場合に、エンドユーザがパスワードをリセットしようとする、回復プロセスの間、ステータスメッセージにその旨が表示されます。

手順の概要

1. 端末または PC をスイッチに接続します。
2. エミュレーションソフトウェアの回線速度を 9600 ボーに設定します。
3. スタンドアロン スイッチまたはスイッチ スタック全体の電源を切断します。

4. スイッチまたはアクティブスイッチに電源コードを再接続します。システムLEDが点滅したら、すぐに [Mode] ボタンを2〜3回押して放します。スイッチはROMMONモードを開始します。
5. パスワードの回復後、スイッチまたはアクティブスイッチをリロードします。

手順の詳細

ステップ1 端末またはPCをスイッチに接続します。

- 端末または端末エミュレーションソフトウェアが稼働しているPCをスイッチのコンソールポートに接続します。
- PCをイーサネット管理ポートに接続します。

ステップ2 エミュレーションソフトウェアの回線速度を9600ボーに設定します。

ステップ3 スタンドアロンスイッチまたはスイッチスタック全体の電源を切断します。

ステップ4 スイッチまたはアクティブスイッチに電源コードを再接続します。システムLEDが点滅したら、すぐに [Mode] ボタンを2〜3回押して放します。スイッチはROMMONモードを開始します。

リロード中に次のコンソールメッセージが表示されます。

```
Initializing Hardware...

System Bootstrap, Version 16.6.1r [FC1], RELEASE SOFTWARE (P)
Compiled Sat 07/15/2017 8:31:57.39 by rel

Current image running:
Primary Rommon Image

Last reset cause: SoftwareReload <---- Start pressing and releasing the mode button
C9300-24U platform with 8388608 Kbytes of main memory

attempting to boot from [flash:packages.conf]

Located file packages.conf
#
#####

Unable to load cat9k-rpboot.16.06.02b.SPA.pkg
Failed to boot file flash:user/packages.conf
ERROR: failed to boot from flash:packages.conf (Aborted) <--- will abort
switch:
switch: <---- ROMMON
```

「パスワード回復がイネーブルになっている場合の手順」セクションに記載されている手順を実行します。

ステップ5 パスワードの回復後、スイッチまたはアクティブスイッチをリロードします。

スイッチの場合

```
Switch> reload
Proceed with reload? [confirm] y
```

パスワード回復がイネーブルになっている場合の手順

ステップ 1 次のコマンドを使用して、スタートアップ コンフィギュレーションを無視します。

```
Device: SWITCH_IGNORE_STARTUP_CFG=1
```

ステップ 2 `packages.conf` ファイルでスイッチをフラッシュからブートします。

```
Device: boot flash:packages.conf
```

ステップ 3 **No** と応答して初期設定ダイアログを終了します。

```
Would you like to enter the initial configuration dialog? [yes/no]: No
```

ステップ 4 スイッチ プロンプトで、特権 EXEC モードを開始します。

```
Device> enable  
Device#
```

ステップ 5 スタートアップ コンフィギュレーションを実行コンフィギュレーションにコピーします。

```
Device# copy startup-config running-config Destination filename [running-config]?
```

確認を求めるプロンプトに、Return を押して応答します。これで、コンフィギュレーション ファイルがリロードされ、パスワードを変更できます。

ステップ 6 グローバル コンフィギュレーション モードを開始して、イネーブルパスワードを変更します。

```
Device# configure terminal  
Device(config)# enable secret password
```

ステップ 7 特権 EXEC モードに戻ります。

```
Device(config)# exit  
Device#
```

ステップ 8 実行コンフィギュレーションをスタートアップ コンフィギュレーション ファイルに書き込みます。

```
Device# copy running-config startup-config
```

ステップ 9 手動ブート モードがイネーブルになっていることを確認します。

```
Device# show boot  
  
BOOT variable = flash:packages.conf;
```

```
Manual Boot = yes
Enable Break = yes
```

ステップ 10 デバイスのリロード。

```
Device# reload
```

ステップ 11 SWITCH_IGNORE_STARTUP_CFG パラメータを 0 に設定します。

```
Device(config)# no system ignore startupconfig switch all
Device(config)# end
Device# write memory
```

ステップ 12 フラッシュの *packages.conf* ファイルを使用して、デバイスを起動します。

```
Device: boot flash:packages.conf
```

ステップ 13 デバイスが起動したら、デバイスで手動ブートを無効にします。

```
Device(config)# no boot manual
```

パスワード回復がディセーブルになっている場合の手順

パスワード回復メカニズムがディセーブルの場合、次のメッセージが表示されます。

```
The password-recovery mechanism has been triggered, but
is currently disabled. Access to the boot loader prompt
through the password-recovery mechanism is disallowed at
this point. However, if you agree to let the system be
reset back to the default system configuration, access
to the boot loader prompt can still be allowed.
```

```
Would you like to reset the system back to the default configuration (y/n)?
```



注意 デバイスをデフォルト設定に戻すと、既存の設定がすべて失われます。システム管理者に問い合わせ、バックアップデバイスと VLAN (仮想 LAN) コンフィギュレーションファイルがあるかどうかを確認してください。

- **n** (no) を入力すると、Mode ボタンを押さなかった場合と同様に、通常のブートプロセスが継続されます。ブートローダプロンプトにはアクセスできません。したがって、新しいパスワードを入力できません。次のメッセージが表示されます。

```
Press Enter to continue.....
```

- **y** (yes) を入力すると、フラッシュメモリ内のコンフィギュレーションファイルおよび VLAN データベースファイルが削除されます。デフォルト設定がロードされるときに、パスワードをリセットできます。

ステップ1 パスワード回復手順の継続を選択すると、既存の設定が失われます。

```
Would you like to reset the system back to the default configuration (y/n)? Y
```

ステップ2 フラッシュメモリの内容を表示します。

```
Device: dir flash:
```

デバイスのファイルシステムが表示されます。

```
Directory of flash:/
.
.
.i'
15494 drwx      4096   Jan 1 2000 00:20:20 +00:00 kirch
15508 -rw-    258065648   Sep 4 2013 14:19:03 +00:00
cat9k_caa-universalk9.SSA.03.12.02.EZP.150-12.02.EZP.150-12.02.EZP.bin
162196684
```

ステップ3 システムを起動します。

```
Device: boot
```

セットアッププログラムを起動するように求められます。パスワード回復手順を継続するには、プロンプトに **N** を入力します。

```
Continue with the configuration dialog? [yes/no]: N
```

ステップ4 デバイスプロンプトで、特権 EXEC モードを開始します。

```
Device> enable
```

ステップ5 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。

```
Device# configure terminal
```

ステップ6 パスワードを変更します。

```
Device(config)# enable secret password
```

シークレットパスワードは 1 ~ 25 文字の英数字です。数字で始めることができます。大文字と小文字が区別され、スペースを使用できますが、先行スペースは無視されます。

ステップ7 特権 EXEC モードに戻ります。

```
Device (config)# exit  
Device#
```

(注) ステップ 9 に進む前に、接続されているすべてのスタック メンバの電源を入れ、それらが完全に初期化されるまで待ちます。

ステップ 8 実行コンフィギュレーションをスタートアップ コンフィギュレーション ファイルに書き込みます。

```
Device# copy running-config startup-config
```

新しいパスワードがスタートアップ コンフィギュレーションに組み込まれました。

ステップ 9 ここで、デバイスを再設定する必要があります。システム管理者によって、バックアップデバイスと VLAN コンフィギュレーション ファイルが使用可能に設定されている場合は、これらを使用します。

スイッチ スタック問題の回避

スイッチ スタックの問題を防止するには、次の作業を実行する必要があります。

- スイッチスタックに デバイスを追加したり、そこから取り外したりする場合には、必ずその電源を切ってください。スイッチスタックでの電源関連のあらゆ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 (スイッチのインストール)」の章を参照してください。
- 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。デバイスの最後の 2 つのポート LED がグリーンになります。デバイス モデルに応じて、最後の 2 つのポートは 10/100/1000 ポートまたは Small Form-Factor Pluggable モジュールになります。最後の 2 つのポート LED の片方または両方がグリーンになっていない場合は、スタックが全帯域幅で稼働していません。
- スイッチスタックを管理する場合は、1 つの CLI セッションだけを使用することを推奨します。アクティブスイッチに複数の 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。
- スタック内での位置に従ってスタックメンバ番号を手動で割り当てると、リモートから行うデバイス スタックのトラブルシューティングが容易になります。ただし、後からデバイスを追加したり、取り外したり、場所を入れ替えたりする際に、デバイスに手動で番号を割り当てたことを覚えておく必要があります。スタックメンバ番号を手動で割り当てるには、**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 コマンドを使用します。

スタックメンバをまったく同じモデルで置き換えると、新しいデバイスは、置き換えられたデバイスとまったく同じ設定で稼働します。この場合、新しいデバイスは置き換えられたデバイスと同じメンバ番号を使用するものと想定されます。

電源が入った状態のスタック メンバを取り外すと、スイッチスタックが、それぞれ同じ設定を持つ 2 つ以上のスイッチスタックに分割 (パーティション化) されます。スイッチスタック

クを分離されたままにしておきたい場合は、新しく作成されたスイッチ スタックの IP アドレス（複数の場合あり）を変更してください。パーティション化されたスイッチスタックを元に戻すには、次の手順を実行します。

1. 新しく作成されたスイッチ スタックの電源を切ります。
2. 新しいスイッチ スタックを、StackWise Plus ポートを介して元のスイッチ スタックに再度接続します。
3. デバイスの電源を入れます。

スイッチ スタックおよびそのメンバのモニターリングに使用できるコマンドについては、「*Displaying Switch Stack Information*」の項を参照してください。

自動ネゴシエーションの不一致の防止

IEEE 802.3ab 自動ネゴシエーション プロトコルは速度（10 Mbps、100 Mbps、および SFP モジュールポート以外の 1000 Mbps）およびデュプレックス（半二重または全二重）に関するデバイスの設定を管理します。このプロトコルは設定を適切に調整しないことがあり、その場合はパフォーマンスが低下します。不一致は次の条件で発生します。

- 手動で設定した速度またはデュプレックスのパラメータが、接続ポート上で手動で設定された速度またはデュプレックスのパラメータと異なっている場合。
- ポートを自動ネゴシエーションに設定したが、接続先ポートは自動ネゴシエーションを使用しない全二重に設定されている場合。

デバイスのパフォーマンスを最大限に引き出してリンクを確保するには、次のいずれかの注意事項に従って、デュプレックスおよび速度の設定を変更してください。

- 速度とデュプレックスの両方について、両方のポートで自動ネゴシエーションを実行させます。
- 接続の両側でポートの速度とデュプレックスのパラメータを手動で設定します。



(注) 接続先装置が自動ネゴシエーションを実行しない場合は、2つのポートのデュプレックス設定を一致させます。速度パラメータは、接続先のポートが自動ネゴシエーションを実行しない場合でも自動調整が可能です。

SFP モジュールのセキュリティと識別に関するトラブルシューティング

シスコの Small Form-Factor Pluggable (SFP) モジュールは、モジュールのシリアル番号、ベンダー名とベンダー ID、一意のセキュリティ コード、および巡回冗長検査 (CRC) が格納されたシリアル EEPROM（電氣的に消去可能でプログラミング可能な ROM）を備えています。デ

デバイスに SFP モジュールを装着すると、デバイス ソフトウェアは、EEPROM を読み取ってシリアル番号、ベンダー名、およびベンダー ID を確認し、セキュリティコードと CRC を再計算します。シリアル番号、ベンダー名、ベンダー ID、セキュリティコード、または CRC が無効な場合、ソフトウェアは、セキュリティ エラー メッセージを生成し、インターフェイスを **errdisable** ステートにします。



- (注) セキュリティ エラー メッセージは、**GBIC_SECURITY** 機能を参照します。デバイスは、SFP モジュールをサポートしていますが、GBIC (ギガビット インターフェイス コンバータ) モジュールはサポートしていません。エラーメッセージテキストは、GBIC インターフェイスおよびモジュールを参照しますが、セキュリティ メッセージは、実際は SFP モジュールおよびモジュール インターフェイスを参照します。

他社の SFP モジュールを使用している場合、デバイス から SFP モジュールを取り外し、シスコのモジュールに交換します。シスコの SFP モジュールを装着したら、**errdisable recovery cause gbic-invalid** 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ポートのステータスを確認し、**error-disabled** 状態から回復する時間間隔を入力します。この時間間隔が経過すると、デバイスは **error-disabled** 状態からインターフェイスを回復させ、操作を再試行します。**errdisable recovery** コマンドの詳細については、このリリースに対応するコマンドリファレンスを参照してください。

モジュールがシスコ製 SFP モジュールとして識別されたにもかかわらず、システムがベンダーデータ情報を読み取ってその情報が正確かどうかを確認できないと、SFP モジュール エラーメッセージが生成されます。この場合、SFP モジュールを取り外して再び装着してください。それでも障害が発生する場合は、SFP モジュールが不良品である可能性があります。

SFP モジュール ステータスのモニタリング

show interfaces transceiver 特権 EXEC コマンドを使用すると、SFP モジュールの物理または動作ステータスを確認できます。このコマンドは、温度や特定のインターフェイス上の SFP モジュールの現状などの動作ステータスと、アラームステータスを表示します。また、このコマンドを使用して SFP モジュールの速度およびデュプレックス設定も確認できます。詳細については、このリリースに対応するコマンドリファレンスにある **show interfaces transceiver** コマンドを参照してください。

ping の実行

別の IP サブネットワーク内のホストに ping を実行する場合は、ネットワークへのスタティックルートを定義するか、またはこれらのサブネット間でルーティングされるように IP ルーティングを設定する必要があります。

IP ルーティングは、デフォルトではすべてのデバイスでディセーブルになります。



- (注) **ping** コマンドでは、他のプロトコルキーワードも使用可能ですが、このリリースではサポートされていません。

このコマンドは、デバイスからネットワーク上の他のデバイスに ping を実行する目的で使用します。

コマンド	目的
ping ip <i>host address</i> デバイス# ping 172.20.52.3	IP またはホスト名やネットワークアドレスを指定してリモートホストに ping を実行します。

温度のモニタリング

デバイスは温度条件をモニターし、温度情報を使用してファンを制御します。

温度の値、状態、しきい値を表示するには、**show env temperature status** 特権 EXEC コマンドを使用します。温度の値は、デバイス内の温度であり、外部の温度ではありません。**system env temperature threshold yellow value** 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イエローのしきい値レベル（摂氏）だけを設定し、イエローのしきい値およびレッドのしきい値の差を設定できます。グリーンまたはレッドのしきい値は設定できません。詳細については、このリリースのコマンドリファレンスを参照してください。

物理パスのモニタリング

次のいずれかの特権 EXEC コマンドを使用して、パケットが通過する、送信元デバイスから宛先デバイスへの物理パスをモニタできます。

表 14: 物理パスのモニタリング

コマンド	目的
tracetroute mac [interface <i>interface-id</i>] { <i>source-mac-address</i> } [interface <i>interface-id</i>] { <i>destination-mac-address</i> } [vlan <i>vlan-id</i>] [detail]	指定の送信元 MAC アドレスから、指定の宛先 MAC アドレスまでをパケットが通過するレイヤ 2 パスを表示します。
tracetroute mac ip { <i>source-ip-address</i> <i>source-hostname</i> } { <i>destination-ip-address</i> <i>destination-hostname</i> } [detail]	指定の送信元 IP アドレスまたはホスト名から、指定の宛先 IP アドレスまたはホスト名を通過するパケットのレイヤ 2 パスを表示します。

IP traceroute の実行



(注) **tracetroute** 特権 EXEC コマンドでは、他のプロトコルキーワードも使用可能ですが、このリリースではサポートされていません。

コマンド	目的
traceroute ip host デバイス# traceroute ip 192.51.100.1	ネットワーク上でパケットが通過するパスを追跡します。

デバッグおよびエラーメッセージ出力のリダイレクト

デフォルトでは、ネットワークサーバが **debug** コマンドからの出力とシステムエラーメッセージをコンソールに送信します。このデフォルトの設定を使用する場合は、コンソールポートまたはイーサネット管理ポートに接続する代わりに、仮想端末接続によってデバッグ出力をモニターできます。

指定できる宛先として、コンソール、仮想端末、内部バッファ、およびsyslogサーバを実行している UNIX ホストがあります。Syslog フォーマットは、4.3 BSD UNIX およびそのバリエーションと互換性があります。



- (注) デバッグの出力先がシステムのオーバーヘッドに影響を与えることがないように注意してください。メッセージをコンソールに記録すると、非常に高いオーバーヘッドが発生します。仮想端末にメッセージを記録すると、発生するオーバーヘッドは低くなります。Syslog サーバでメッセージロギングを行うと、オーバーヘッドはさらに小さくなり、内部バッファであれば最小限ですみます。

システムメッセージのロギングに関する詳細については、「システムメッセージロギングの設定」を参照してください。

show platform forward コマンドの使用

show platform forward 特権 EXEC コマンドの出力からは、インターフェイスに着信するパケットがシステムを介して送信された場合の転送結果に関する有意義な情報がいくつか得られます。パケットに関して入力されたパラメータに応じて、参照テーブル結果、転送宛先の計算に使用されるポートマップ、ビットマップ、および出力側の情報が表示されます。

このコマンドで出力される情報のほとんどは、主に、デバイスの特定用途向け集積回路 (ASIC) に関する詳細情報を使用するテクニカルサポート担当者に役立つものです。ただし、パケット転送情報はトラブルシューティングにも役立ちます。

show debug コマンドの使用法

show debug コマンドは特権 EXEC モードで入力します。このコマンドは、スイッチで使用可能なすべてのデバッグ オプションを表示します。

すべての条件付きデバッグオプションを表示するには、コマンド **show debug condition** を実行します。コマンドは、条件 ID <1-1000> または *all* 条件を選択することで一覧表示できます。

デバッグを無効にするには、**no debug all** コマンドを使用します。



注意 デバッグ出力は CPU プロセスで高プライオリティが割り当てられているため、デバッグ出力を行うとシステムが使用できなくなることがあります。したがって、**debug** コマンドを使用するのは、特定の問題のトラブルシューティング時、またはシスコのテクニカルサポート担当者とともにトラブルシューティングを行う場合に限定してください。さらに、**debug** コマンドは、ネットワークトラフィックが少なく、ユーザも少ないときに使用することを推奨します。デバッグをこのような時間帯に行うと、**debug** コマンド処理のオーバーヘッドの増加によりシステムの使用に影響が及ぶ可能性が低くなります。

OBFL の設定



注意 OBFL はディセーブルにせず、フラッシュメモリに保存されたデータは削除しないことを推奨します。

- OBFL をイネーブルにするには、**hw-switch switch [switch-number] logging onboard [message level level]** 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。スイッチでは、*switch-number* に指定できる範囲は 1～9 です。スイッチが生成してフラッシュメモリに保存するハードウェア関連のメッセージの重大度を指定するには、**message level level** パラメータを使用します。
- OBFL データをローカルネットワークまたは特定のファイルシステムにコピーするには、**copy logging onboard switch switch-number url url-destination** 特権 EXEC コマンドを使用します。
- OBFL をイネーブルにするには、**no hw-switch switch [switch-number] logging onboard [message level]** 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。
- フラッシュメモリ内の稼働時間と CLI コマンド情報以外のすべての OBFL データをクリアするには、**clear logging onboard switch switch-number** 特権 EXEC コマンドを使用します。
フラッシュメモリ内の稼働時間以外のすべての OBFL データをクリアするには、**clear logging onboard RP active {application}** 特権 EXEC コマンドを使用します。

ソフトウェア設定のトラブルシューティングの確認

OBFL 情報の表示

表 15: OBFL 情報を表示するためのコマンド

コマンド	目的
show onboard switch <i>switch-number</i> cliilog デバイス# show onboard switch 1 cliilog	スタンドアロンスイッチまたは指定されたスタックメンバで入力された OBFL CLI コマンドを表示します。
show onboard switch <i>switch-number</i> environment デバイス# show onboard switch 1 environment	スタンドアロンスイッチまたは指定されたスタックメンバおよび接続されているすべての FRU デバイスの UDI 情報、PID、VID、およびシリアル番号を表示します。
show onboard switch <i>switch-number</i> message デバイス# show onboard switch 1 message	スタンドアロンスイッチまたは指定されたスタックメンバによって生成されたハードウェア関連のメッセージを表示します。
show onboard switch <i>switch-number</i> counter デバイス# show onboard switch 1 counter	スタンドアロンスイッチまたは指定されたスタックメンバのカウンタ情報を表示します。
show onboard switch <i>switch-number</i> temperature デバイス# show onboard switch 1 temperature	スタンドアロンスイッチまたは指定されたスイッチスタックメンバの温度を表示します。
show onboard switch <i>switch-number</i> uptime デバイス# show onboard switch 1 uptime	スタンドアロンスイッチまたは指定されたスタックメンバが起動した時刻、スタンドアロンスイッチまたは指定されたスタックメンバが再起動された理由、およびスタンドアロンスイッチまたは指定されたスタックメンバが最後に再起動されて以来の稼働時間を表示します。
show onboard switch <i>switch-number</i> voltage デバイス# show onboard switch 1 voltage	スタンドアロンスイッチまたは指定されたスタックメンバのシステム電圧を表示します。

コマンド	目的
show onboard switch switch-number status デバイス# show onboard switch 1 status	スタンドアロンスイッチまたは指定されたスタックメンバの状態を表示します。

例：高い CPU 使用率に関する問題と原因の確認

CPU 使用率が高いことが問題となっているかどうか判断するには、**show processes cpu sorted** 特権 EXEC コマンドを入力します。出力例の 1 行目にある下線が付いた部分に注目してください。

```

デバイス# show processes cpu sorted
CPU utilization for five seconds: 8%/0%; one minute: 7%; five minutes: 8%
PID Runtime(ms) Invoked uSecs 5Sec 1Min 5Min TTY Process
309 42289103 752750 56180 1.75% 1.20% 1.22% 0 RIP Timers
140 8820183 4942081 1784 0.63% 0.37% 0.30% 0 HRPC qos request
100 3427318 16150534 212 0.47% 0.14% 0.11% 0 HRPC pm-counters
192 3093252 14081112 219 0.31% 0.14% 0.11% 0 Spanning Tree
143 8 37 216 0.15% 0.01% 0.00% 0 Exec
...
<output truncated>

```

この例は、正常な CPU 使用率を示しています。この出力によると、最後の 5 秒間の使用率が 8%/0% となっていますが、この意味は次のとおりです。

- Cisco IOS の処理時間と割り込みの処理にかかった時間を合わせた CPU の合計の使用率は全体の 8%
- 割り込みの処理にかかった時間は全体の 0%

表 16: CPU 使用率に関する問題のトラブルシューティング

問題のタイプ	原因	修正措置
割り込みのパーセント値が合計の CPU 使用率の値とほぼ同程度に高い	CPU がネットワークから受信するパケット数が多すぎる。	ネットワーク パケットのソースを判別する。データの流れを遮断するか、スイッチの設定を変更します。「Analyzing Network Traffic (ネットワークトラフィックの解析)」の項を参照してください。
割り込みの所要時間は最小限であったにもかかわらず CPU の合計使用率が 50% を超える	CPU 時間を過度に消費する Cisco IOS 処理が 1 つ以上存在する。これは通常、処理をアクティブ化するイベントによって始動されます。	異常なイベントを特定して根本的な原因を解消する。「Debugging Active Processes (アクティブなプロセスのデバッグ)」のセクションを参照してください。

ソフトウェア設定のトラブルシューティングのシナリオ

Power over Ethernet (PoE) に関するトラブルシューティングのシナリオ

表 17: *Power over Ethernet* に関するトラブルシューティングのシナリオ

症状または問題	考えられる原因と解決法
<p>PoE がないポートは1つに限られません。</p> <p>1つのスイッチポートに限り問題が発生する。このポートではPoE装置と PoE 非対応の装置のいずれも動作しないが、他のポートでは動作します。</p>	

症状または問題	考えられる原因と解決法
	<p>この受電デバイスが他の PoE ポートで動作するかを確認する。</p> <p>show run または show interface status ユーザ EXEC コマンドを使用して、ポートがシャットダウンしていないか、または error-disabled になっていないかを確認します。</p> <p>(注) ほとんどのスイッチはポートがシャットダウンしているときはポートの電力供給をオフにします。これは、IEEE 仕様でこれがオプションに指定されている場合も同様です。</p> <p>該当するインターフェイスまたはポートに power inline never が設定されていないことを確認します。</p> <p>受電デバイスからスイッチポートまでのイーサネットケーブルの動作が正常であることを確認します。具体的には、既知の正常な PoE 非対応のイーサネット装置とイーサネットケーブルを接続して、受電デバイスがリンクを確立し他のホストとトラフィックを交換することを確認します。</p> <p>(注) シスコ受電装置は、ストレート ケーブルでのみ機能します。クロスオーバーケーブルでは機能しません。</p> <p>スイッチのフロントパネルから受電デバイスまでのケーブル長の合計が 100 メートル以下であることを確認します。</p> <p>スイッチポートからイーサネットケーブルを外します。短いイーサネットケーブルを使用して、既知の正常なイーサネット装置を、スイッチのフロントパネルの (パッチパネルではない) このポートに直接接続します。これによってイーサネットリンクが確立され他のホストとトラフィックを交換できることを確認します。あるいは、ポートの VLAN SVI で ping を実行してください。次に、受電デバイスをこのポートに接続し、電源がオンになることを確認します。</p> <p>パッチコードをスイッチポートに接続しても受電デバイスの電源がオンにならない場合、接続する受電デバイスの合計数とスイッチの電力バジェット (使用可能な PoE) とを比較してください。 show power inline コマンドを使用して、利用可能な電力量を確認します。</p>

症状または問題	考えられる原因と解決法
<p>すべてのポートまたは1つのポートグループで PoE が機能しない。</p> <p>すべてのスイッチポートで問題が発生する。電力が供給されていないイーサネット装置がどのポートでもイーサネットリンクを確立できず、PoE装置の電源がオンになりません。</p>	

症状または問題	考えられる原因と解決法
	<p>電力に関するアラームが継続的に発生する、断続的に発生する、または再発する場合は、可能であれば電源モジュールを交換します（現場交換可能ユニットです）。そうでない場合はスイッチを交換してください。</p> <p>連続する複数のポートで問題があるものの、すべてのポートで問題が発生するわけではない場合、電源の故障ではないと考えられ、スイッチのPoEレギュレータに関連した異常の可能性があります。</p> <p>PoE の状況やステータスの変更について過去に報告されているアラームまたはシステムメッセージがないか、show log 特権 EXEC コマンドを使用して調べます。</p> <p>アラームがない場合は、show interface status コマンドを使用して、ポートがシャットダウンしていないか、または error-disabled になっていないかを確認します。ポートが error-disabled の場合、shut および no shut インターフェイス コンフィギュレーション コマンドを使用して、ポートを再度有効にします。</p> <p>show env power および show power inline 特権 EXEC コマンドを使用して、PoEのステータスおよび電力バジェット（使用可能な PoE）を調べます。</p> <p>実行コンフィギュレーションを調べて、power inline never がこのポートに設定されていないことを確認します。</p> <p>受電していないイーサネット装置をスイッチポートに直接接続します。接続には短いパッチ コードだけを使用します。既存の配線ケーブルは使用しないでください。shut および no shut 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、イーサネットリンクが確立されていることを確認します。正しく接続している場合、短いパッチコードを使用して受電デバイスをこのポートに接続し、電源がオンになることを確認します。装置の電源がオンになったら、すべての中間パッチパネルが正しく接続されているか確認してください。</p> <p>1本を除くすべてのイーサネットケーブルをスイッチポートから抜きます。短いパッチコードを使用して、1つのPoEポートにだけ受電デバイスを接続します。スイッチポートからの受電に比較して、受電デバイスが多くの電力を必要としないことを確認してください。</p> <p>show power inline 特権 EXEC コマンドを使用して、ポートがシャットダウンされていない場合に、受電デバイスに電力が供給されることを確認します。あるいは、受電デバイ</p>

症状または問題	考えられる原因と解決法
	<p>スを観察して電源がオンになることを確認してください。</p> <p>1 台の受電デバイスだけがスイッチに接続している際に電力が供給される場合、残りのポートで shut および no shut 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から、イーサネットケーブルをスイッチの PoE ポートに 1 本ずつ再接続してください。 show interface status および show power inline 特権 EXEC コマンドを使用して、インラインパワーの統計情報とポートのステータスをモニタします。</p> <p>すべてのポートで、まだ PoE が機能しない場合は、電源装置の PoE セクションでヒューズを開くことができる場合があります。この場合、アラームが生成されるのが一般的です。過去にシステムメッセージでアラームが報告されていないか、ログをもう一度チェックしてください。</p>
<p>シスコ先行標準受電装置は、切断またはリセットされます。</p> <p>正常に動作した後で、シスコ電話機が断続的にリロードしたり、PoE から切断されたりします。</p>	<p>スイッチから受電デバイスまでのすべての電気系統を確認してください。信頼性の低い接続は、電力供給の中断や受電デバイスの機能が不安定になる原因となり、受電デバイスの断続的な切断やリロードが発生します。</p> <p>スイッチポートから受電デバイスまでのケーブル長が 100 メートル以下であることを確認してください。</p> <p>スイッチが配置されている場所で電気環境にどのような変化があるか、切断時に、受電デバイスに何が起きるかについて注意してください。</p> <p>切断と同時にエラーメッセージが表示されたか注意します。 show log 特権 EXEC コマンドを使用して、エラーメッセージを確認します。</p> <p>リロードの発生直前に IP Phone から Call Manager へのアクセスが失われていないか確認してください (PoE の障害ではなくネットワークに問題が発生している場合があります)。</p> <p>受電デバイスを PoE 非対応の装置に交換し、装置が正しく動作することを確認します。PoE 非対応の装置にリンク障害または高いエラー率がある場合、スイッチポートと受電デバイスを接続する信頼性の低いケーブル接続が問題の可能性もあります。</p>

症状または問題	考えられる原因と解決法
IEEE 802.3af 準拠または IEEE 802.3at 準拠の受電装置は、Cisco PoE スイッチでは機能しません。 シスコ PoE スイッチに接続するシスコ以外の受電デバイスに電源が供給されないか、電源投入後すぐに電源が切れます。PoE 非対応装置は正常に動作します。	<p>show power inline コマンドを使用して、受電デバイスの接続前後に、スイッチの電力バジェット（使用可能な PoE）が枯渇していないか確認します。受電デバイスを接続する前に、このタイプの装置に十分な電力が使用可能であることを確認します。</p> <p>show interface status コマンドを使用して、接続されている受電デバイスがスイッチに検出されることを確認します。</p> <p>show log コマンドを使用して、ポートの過電流状態を報告したシステムメッセージがないか確認します。症状を正確に特定してください。最初に電力が受電デバイスに供給され、その後、切断される状態ですか。その場合は、問題は最初のサージ電流（突入電流）が原因で、ポートの電流上限しきい値が超過した可能性があります。</p>

ソフトウェアのトラブルシューティングの設定例

例：IP ホストの ping

次に、IP ホストに ping を実行する例を示します。

```
デバイス# ping 172.20.52.3
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 172.20.52.3, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
デバイス#
```

表 18: ping の出力表示文字

文字	Description
!	感嘆符 1 個につき 1 回の応答を受信したことを示します。
.	ピリオド 1 個につき応答待ちの間にネットワーク サーバのタイムアウトが 1 回発生したことを示します。
U	宛先到達不能エラー PDU を受信したことを示します。
C	輻輳に遭遇したパケットを受信したことを示します。
I	ユーザによりテストが中断されたことを示します。

文字	Description
?	パケットタイプが不明です。
&	パケットの存続時間を超過したことを示します。

pingセッションを終了するには、エスケープシーケンス（デフォルトではCtrl+^X）を入力してください。Ctrlキー、Shiftキー、および6キーを同時に押してから放し、その後Xキーを押します。

例：IP ホストに対する **traceroute** の実行

次に、IP ホストに **traceroute** を実行する例を示します。

```
デバイス# traceroute ip 192.0.2.10
```

```
Type escape sequence to abort.  
Tracing the route to 192.0.2.10
```

```
 1 192.0.2.1 0 msec 0 msec 4 msec  
 2 192.0.2.203 12 msec 8 msec 0 msec  
 3 192.0.2.100 4 msec 0 msec 0 msec  
 4 192.0.2.10 0 msec 4 msec 0 msec
```

ディスプレイには、送信される3つのプローブごとに、ホップカウント、ルータのIPアドレス、およびラウンドトリップタイム（ミリ秒単位）が表示されます。

表 19: **traceroute** の出力表示文字

文字	Description
*	プローブがタイムアウトになりました。
?	パケットタイプが不明です。
A	管理上、到達不能です。通常、この出力は、アクセスリストがトラフィックをブロックしていることを表しています。
H	ホストが到達不能です。
N	ネットワークが到達不能です。
P	プロトコルが到達不能です。
Q	発信元。
U	ポートが到達不能です。

実行中の追跡を終了するには、エスケープシーケンス（デフォルトではCtrl+^X）を入力してください。Ctrl キー、Shift キー、および6 キーを同時に押してから放し、その後 X キーを押します。

ソフトウェア設定のトラブルシューティングの機能履歴

次の表に、このモジュールで説明する機能のリリースおよび関連情報を示します。

これらの機能は、特に明記されていない限り、導入されたリリース以降のすべてのリリースで使用できます。

リリース	機能	機能情報
Cisco IOS XE Everest 16.5.1a	ソフトウェア設定のトラブルシューティング	ソフトウェア設定のトラブルシューティングでは、スイッチが稼働する Cisco IOS ソフトウェアに関連する問題を特定し、解決する方法について説明します。

Cisco Feature Navigator を使用すると、プラットフォームおよびソフトウェアイメージのサポート情報を検索できます。Cisco Feature Navigator には、<http://www.cisco.com/go/cfn> [英語] からアクセスします。

翻訳について

このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。